

# **Triconex Ethernet Driver Help**

© 2011 Kepware Technologies

# Table of Contents

|   |    |
|---|----|
| Table of Contents .....   | 2  |
| Triconex Ethernet Driver Help .....   | 4  |
| Overview .....  | 4  |
| Device Setup .....  | 5  |
| Network Interface Channel Properties .....  | 5  |
| CM Configuration Device Properties .....  | 6  |
| Communications Device Properties .....  | 6  |
| Tag Import Device Properties .....  | 8  |
| Cable Diagrams For Tricon .....   | 8  |
| Cable Diagram For Trident .....   | 10 |
| Network Redundancy .....  | 11 |
| Optimizing Your Triconex Ethernet Communications .....                            | 13 |
| Tag Import .....  | 15 |
| Data Types Description .....  | 16 |
| Address Descriptions .....  | 17 |
| Tricon Addressing .....   | 17 |
| Trident Addressing .....  | 19 |
| Driver System Tags .....  | 20 |
| Error Descriptions .....  | 21 |
| Address Validation .....  | 21 |
| Address '<address>' is out of range for the specified device or register .....    | 21 |
| Data Type '<type>' is not valid for device address '<address>' .....              | 22 |
| Device address '<address>' contains a syntax error .....                          | 22 |
| Device address '<address>' is not supported by model '<model name>' .....         | 22 |
| Device address '<address>' is Read Only .....                                     | 22 |
| Missing address .....   | 22 |
| Device Status Messages .....  | 23 |
| Could not detect available network adapters for <primary/secondary> network ..... | 23 |
| Device '<device name>' is not responding .....                                    | 23 |
| Failed to load Winsock or related library .....                                   | 24 |
| Tag import error - invalid bin in item path: '<item path>' .....                  | 24 |
| Tag import error - invalid item path format: '<item path>' .....                  | 24 |
| Tag import error - invalid name: '<tag name>' .....                               | 24 |
| Tag import error - invalid offset in item path: '<item path>' .....               | 24 |
| Tag import failed. Could not find data for node '<node>' .....                    | 25 |
| Tag import failed. Could not open file '<file>' .....                             | 25 |
| Tag import failed due to unexpected XML format in '<file>' .....                  | 25 |
| Tag import failed. File name given, but no node name .....                        | 25 |

|   |           |
|---|-----------|
| Tag import failed. No import file name or system tags specified.....                      | 25        |
| Unable to bind to local address (<IP, port>).....   | 26        |
| Unable to process data update on device '<device>'. CRC error.....                        | 26        |
| Unable to process data update on device '<device>'. Unexpected message format.....        | 26        |
| Unable to process data update on device '<device>'. Unexpected model.....                 | 26        |
| Unable to process data update on device '<device>'. Unexpected node number.....           | 26        |
| Unable to use network redundancy on channel '<channel>'.....                              | 27        |
| Unable to use specified primary network adapter '<adapter>' on channel '<channel>'.....   | 27        |
| Unable to use specified secondary network adapter '<adapter>' on channel '<channel>'..... | 27        |
| Winsock initialization failed (OS Error = <error code>).....                              | 28        |
| Winsock V1.1 or higher must be installed to use the Triconex Ethernet device driver.....  | 28        |
| Write failed for tag '<address>' on device '<device>'. CRC error.....                     | 28        |
| Write failed for tag '<address>' on device '<device>'. No data for Read/Modify/Write..... | 28        |
| Write failed for tag '<address>' on device '<device>'. Timeout.....                       | 28        |
| Write failed for tag '<address>' on device '<device>'. TSAA Response Code: <code>.....    | 29        |
| Write failed for tag '<address>' on device '<device>'. Unexpected model.....              | 29        |
| Write failed for tag '<address>' on device '<device>'. Unexpected node number.....        | 29        |
| Write failed for tag '<address>' on device '<device>'. Unexpected response format.....    | 30        |
| <b>Index</b> .....  | <b>31</b> |

## Triconex Ethernet Driver Help

---

Help version 1.009

### CONTENTS

#### [Overview](#)

What is the Triconex Ethernet Driver?

#### [Device Setup](#)

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

#### [Network Redundancy](#)

How can I use a redundant network with this driver?

#### [Optimizing Ethernet Communications](#)

How can I configure this driver for optimum performance?

#### [Tag Import](#)

How can I import tags from a Triconex TriStation project?

#### [Data Types Description](#)

What data types does this driver support?

#### [Address Descriptions](#)

How do I address a data location on a Triconex Ethernet device?

#### [Error Descriptions](#)

What error messages does the Triconex Ethernet driver produce?

### Overview

---

The Triconex Ethernet Driver provides an easy and reliable way to connect Triconex Ethernet devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It is intended for use with Triconex Tricon and Trident devices equipped with Ethernet communications modules. Network redundancy is supported.

## Device Setup

---

### Supported Devices

Tricon  
Trident

### Communication Protocol

Triconex System Access Application (TSAA)

### Device ID

The Device ID is the node number of the device. Node numbers range from 1 to 31.

### Request timeout

This parameter specifies the number of milliseconds that the driver will wait for responses before retrying. Users may specify a request timeout between 100 to 30000 milliseconds. For this driver, request timeouts only apply to Write operations. The device may take up to 4 scan cycles to complete a Write request. The controller may be programmed to use a scan period as large as 400ms. Set the request timeout accordingly: setting it too low may cause the driver to assume a write has failed, when it really hasn't been processed by the controller yet.

**Note:** Data is read from cache, which is updated at a regular interval. For more information, refer to [Communications Device Properties](#).

### Fail after <number> successive timeouts

This parameter specifies the number of times the driver will retry a request before failing it. Again, this only pertains to write requests for this driver. Users may select between 1 and 10 retries.

### Cable Diagrams

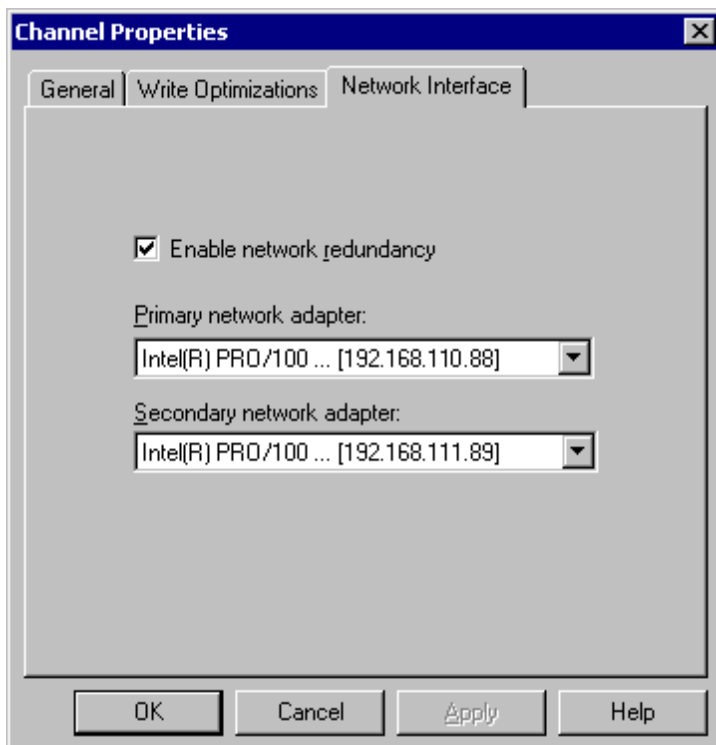
Select a link from the list below to view a specific model's cable diagram.

[Cable Diagrams for Tricon](#)

[Cable Diagram for Trident](#)

## Network Interface Channel Properties

---



### Enable network redundancy

If the PC has more than one network adapter card (NIC) installed, the feature will be enabled. If not, and users desire to enable it, simply check the box. For more information, refer to [Network Redundancy](#).

**Primary network adapter**

This parameter specifies the NIC to which the primary network is connected.

**Secondary network adapter**

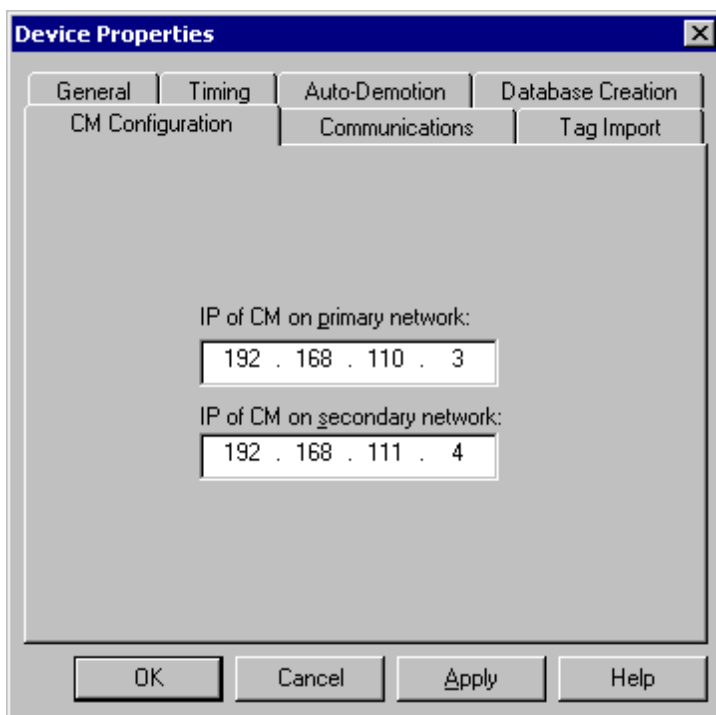
This parameter specifies the NIC to which the optional secondary network is connected. This secondary NIC must not be the same as the primary. It is also recommended that the primary and secondary networks have different subnet addresses. For more information, refer to [Network Redundancy](#).

**Note:** These settings are specific to the host computer. If a Triconex driver project is copied to another machine, these settings will likely need to be changed. The driver will post a message to the server event log if changes are required. The network adapters must be connected to a device or hub in order for Windows to correctly report their configuration to the driver.

**See Also:** [Unable to use specified primary network adaptor](#), [Unable to use specified secondary network adapter](#) and [Unable to use network redundancy](#).

**CM Configuration Device Properties**

---

**IP of CM on primary network**

This parameter specifies the IP address of the communications module (CM) that is connected to the primary network.

**IP of CM on secondary network**

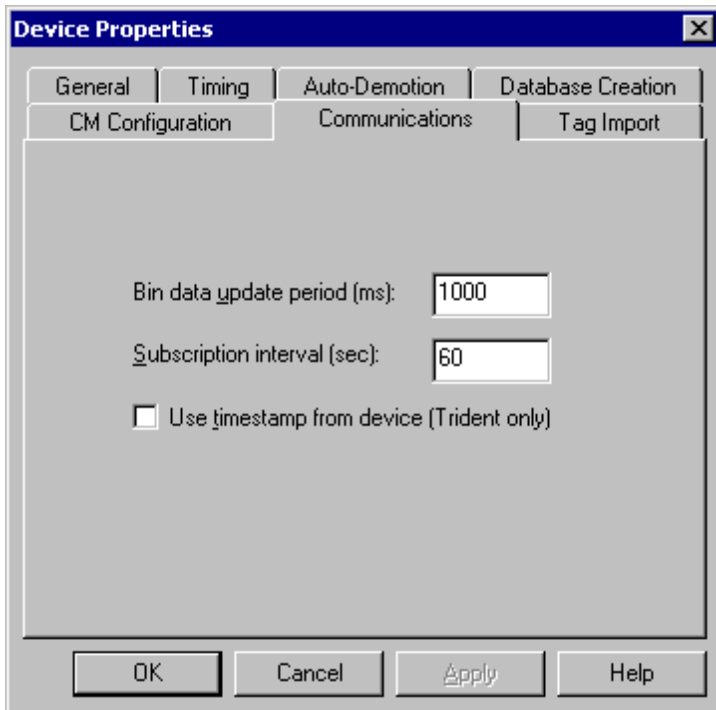
This parameter specifies the IP address of the CM that is connected to the secondary network. It will be enabled if network redundancy is enabled.

**Note:** The primary and secondary IP addresses must be different. We also recommend that the primary and secondary networks have different subnet addresses.

**See Also:** [Network Interface Channel Properties](#) and [Network Redundancy](#).

**Communications Device Properties**

---



### Bin data update period (ms)

This parameter specifies the amount of time between updates. Instead of polling the device for data, this driver requests that the device send data updates at a given interval. Values for entire bins will be sent on each update. This method has proven to be more efficient with the Triconex controllers. Values may range from 50 to 10000 ms.

**Note 1:** When the driver services a Read request from the client application, it returns the last value sent by the device. The data quality will be **Good** if the last update was received within 2 times this update interval. The data quality will be **Uncertain** if the cached data is stale. The data quality will be **Bad** until the first update is received. For more information on how the driver processes data updates when network redundancy is enabled, refer to [Network Redundancy](#).

**Note 2:** The bin data update period is a requested value. If the device is unable to deliver updates at the rate requested here, the OPC data quality may fluctuate between updates and network errors may be reported. We recommend that this value be at least twice the device scan period. If data from many bins is requested, or if a large amount of memory is allocated for some bins, then a larger margin may be required. This is primarily due to message segmentation. Each update will contain all of the values for each bin referenced in the client. The device will break up update messages into 500 byte segments and may take up to 20 ms to send each segment. **See Also:** [Driver System Tags](#).

### Subscription interval (sec)

This parameter specifies the frequency at which the driver should renew its subscription for data updates. The device requires that subscriptions be renewed at least every 120 seconds, or data updates will stop. The valid range of values for **Subscription interval (sec)** is 10 to 120 seconds. If a TR\_PORT\_STATUS function block is being used in the device configuration, users should use the default subscription interval setting of 10 seconds. This will ensure that the driver will send something to the device at least every 10 seconds and maintain the function block's internal receive status flag.

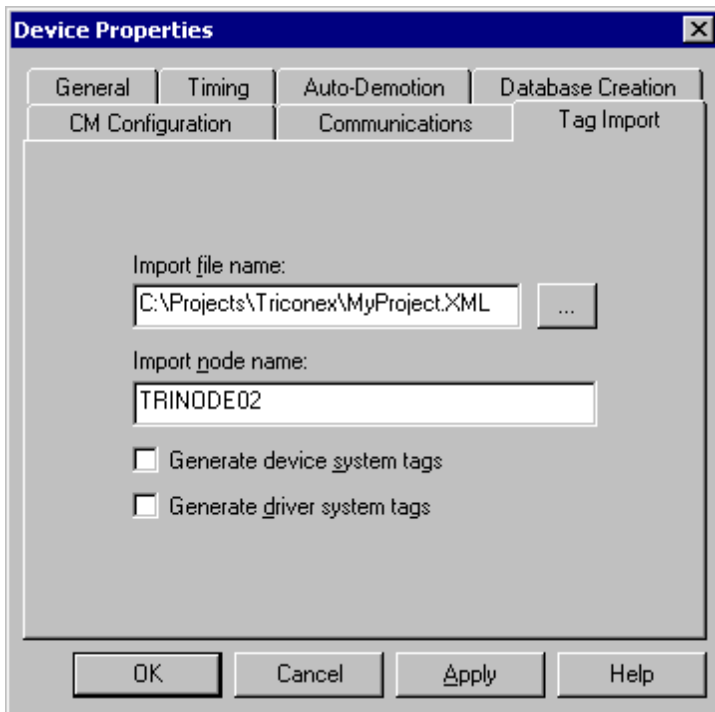
### Use timestamp from device (Trident only)

The timestamp associated with data is normally set by the server as the time of receipt. Select this option to use the timestamp sent up from the device with data updates instead. This option is supported by Trident devices only. Timestamps will always be set by the server when using the Tricon model.

**Note 1:** The Bin data update rate is a requested update rate. If the device cannot send updates at this rate (or if it is configured with a slower scan rate) the data will likely be flagged as stale between updates.

**Note 2:** While reducing the setting, keep in mind that the device continues to send updates for two minutes after the last request was made. The device may not change its update rate for up to two minutes, even when new subscriptions are sent.

## Tag Import Device Properties



### Import file name

Before the driver can automatically create tags for all of the variables defined in the TriStation project, users must export the project from TriStation in XML format. Next, start the OPC server and create a Triconex driver project or load an existing one. Specify the name and location of the TriStation export file from which data will be imported. Users may browse to the import file by clicking the "..." button.

### Import node name

The XML export file may contain data for multiple devices, each with a unique node name. In this parameter, enter the node name associated with the tags that will be imported.

### Generate device system tags

The TriStation export file will not contain device system tags. To have the driver generate a standard set of the most useful system tags, check this box. Users do not need to specify an import file to generate system tags. Data for these device system tags is contained in bins 8 and 9. Since the Trident controllers do not support these bins, this option will not be enabled if the Trident model is selected.

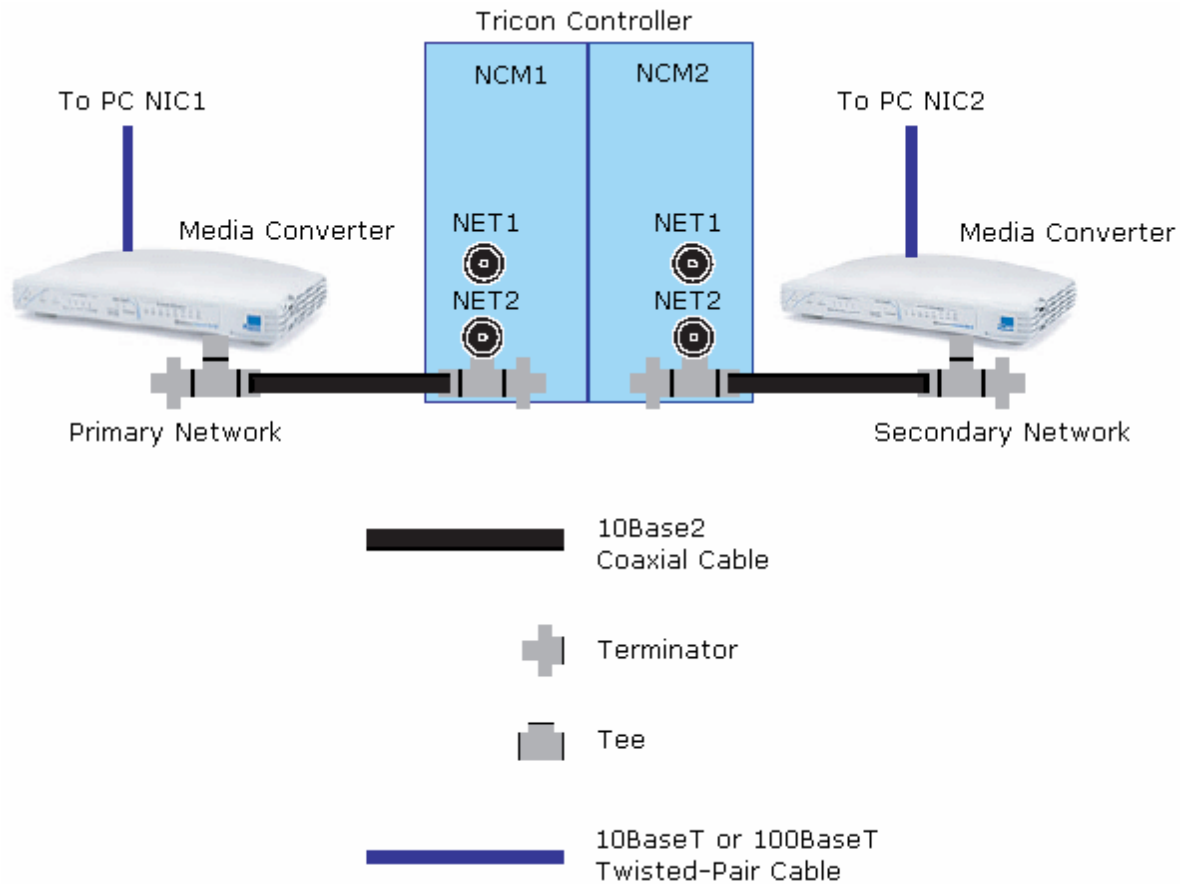
### Generate driver system tags

The driver provides a number of system tags that can be used to monitor driver operation and network health. To automatically generate these tags, check this box. Users do not need to specify an import file to generate system tags. For a list of tags, refer to [Driver System Tags](#).

**Note:** Tags will not be generated at this point. To generate tags immediately, click **Apply** and then select the **Database Creation** tab. Next, click **Auto Create** to generate the tags. Users may configure the driver to perform tag generation in other ways, such as each time the driver starts up. For more information, click on the Help button in Database Creation.

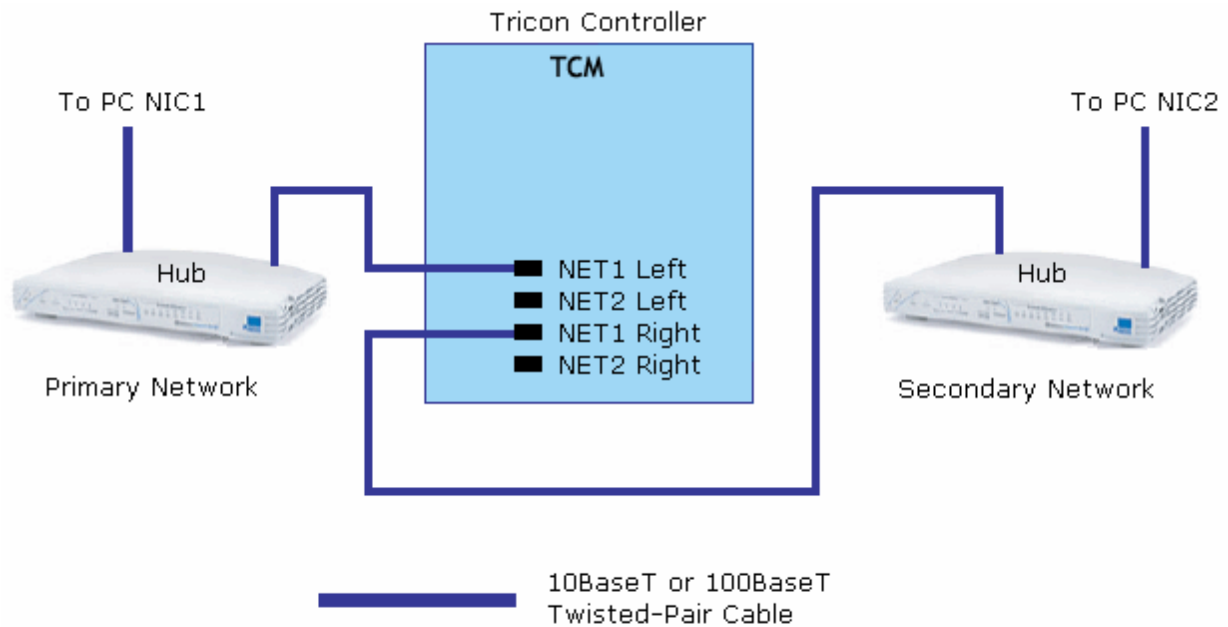
## Cable Diagrams For Tricon

### Tricon Controller with NCM



**Note:** Users must connect to NET2.

**Tricon Controller with TCM (Tricon Communications Module)**

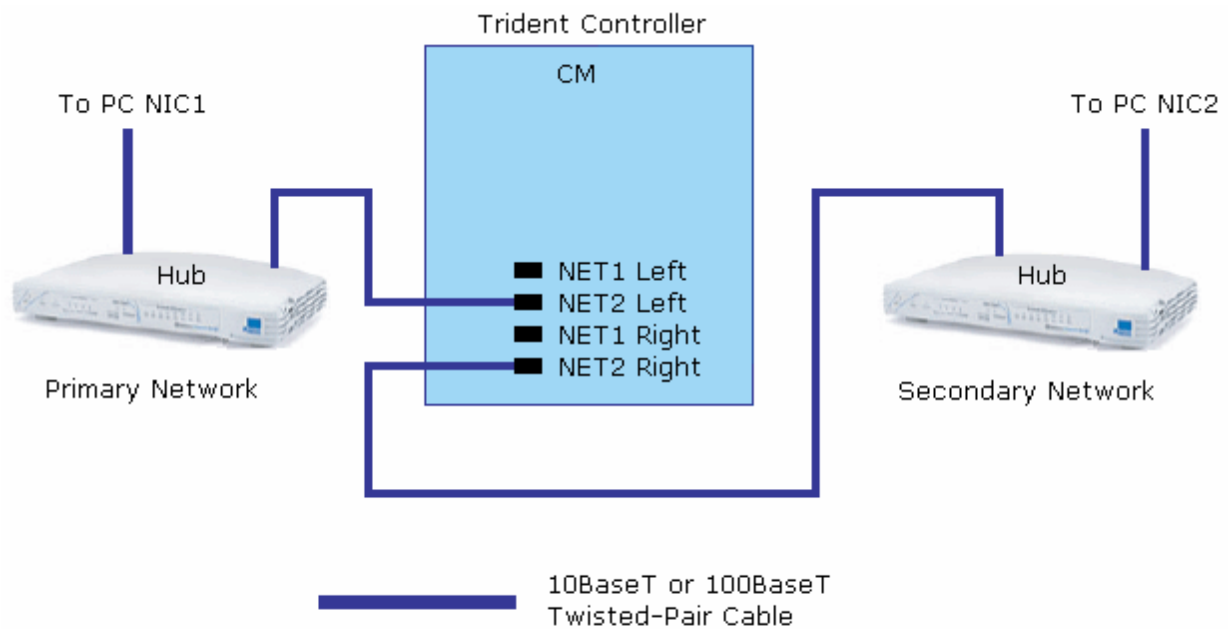


**Note:** For TCM models 4351A/4351B and 4352A/4352B, connect to NET1 or NET2. For TCM models 4353 and 4354, connect to NET1.

See Also: [Network Redundancy](#)

### Cable Diagram For Trident

---



**Note:** Users can connect to NET1 or NET2. NET1 supports 10BaseT only. NET2 supports 10BaseT and 100BaseT.

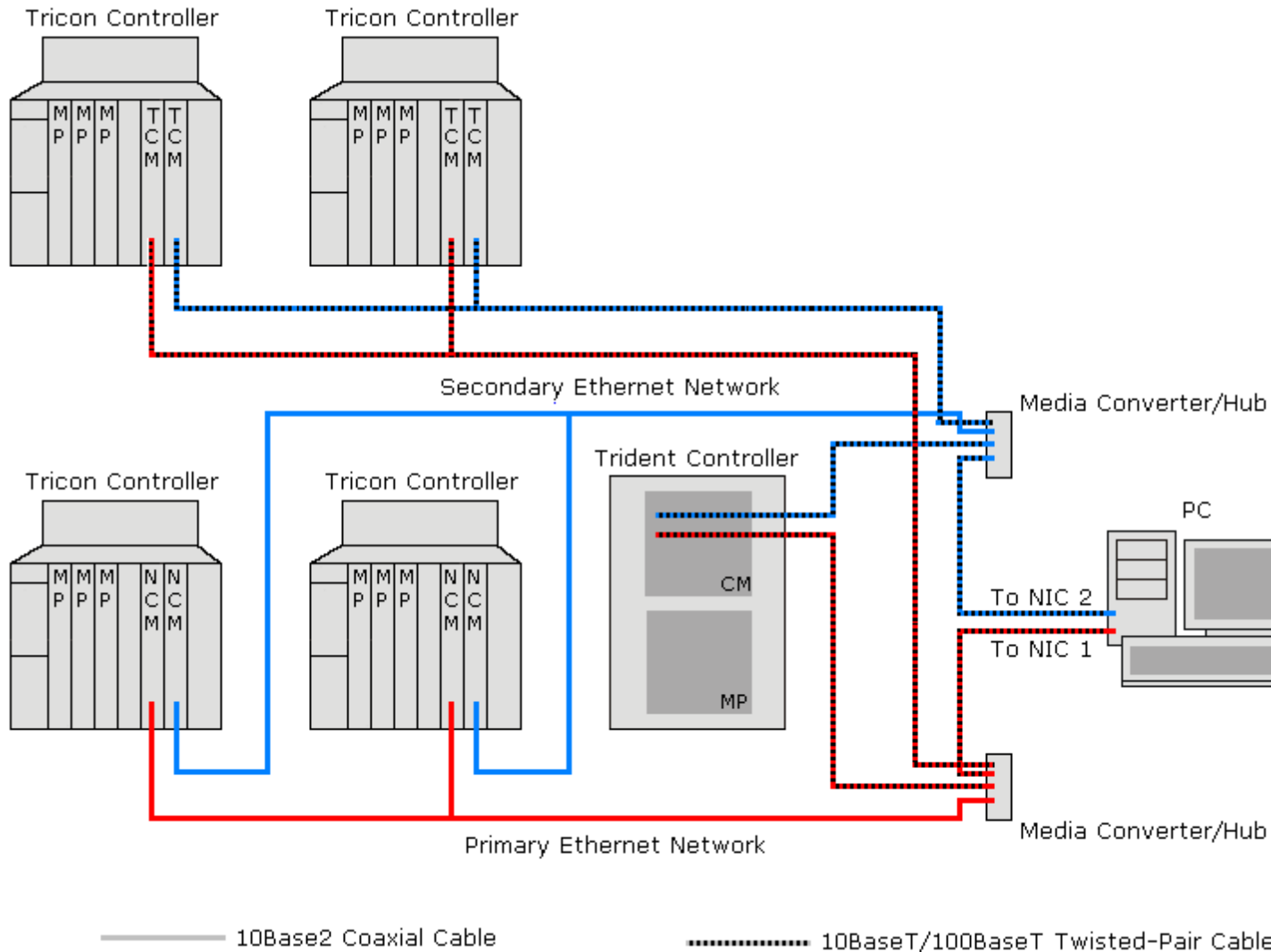
See Also: [Network Redundancy](#)

## Network Redundancy

### Network Configuration

Network redundancy requires that each controller be equipped with two communications modules and that the computer running this driver be equipped with two network interface cards. We recommend that the primary and secondary networks have different subnet addresses. **See Also:** [Network Interface Channel Properties](#) and [CM Configuration Device Properties](#).

Media converters are usually required to connect Tricon controllers (which require 2BaseT coaxial cable) to the NICs. NICs normally require 10BaseT or 100BaseT twisted pair cable. Trident controllers take 10BaseT and 100BaseT twisted pair cables, which can be connected to the NIC via a hub. Triconex recommends combination media converter/hubs from 3Com. A redundant network mixing Tricon and Trident controllers would appear similar to the one shown below.



**See Also:** [Cable Diagrams for Tricon](#) and [Cable Diagram for Trident](#).

### Reads with Network Redundancy Enabled

This driver does not poll the device for data. Instead, it instructs the controller to send data updates at a given interval. Values for entire bins will be sent on each update. This method has proven to be more efficient with the Triconex controllers. **See Also:** [Communications Device Properties](#).

When network redundancy is enabled, the driver will request that the device send data updates on both networks. The driver will process updates from a single network however. If an update fails to arrive on that network within 2 update intervals, the driver will begin processing updates from the other network. Therefore, fresh data may not be available for up to 3 update periods following a network problem. The **UpdatesFromPrimary** and

driver system tags can be used to monitor which network the driver is currently processing updates from.

Though the driver will process updates from a single network, it does monitor incoming traffic from both networks. If nothing has been received on a network for over 2 update periods, the driver will flag that network as being in error. The **PrimaryNetworkError** and **SecondaryNetworkError** driver system tags can be used to monitor the network error state.

**See Also:** [Driver System Tags](#)

### **Writes with Network Redundancy Enabled**

The driver will send a Write request over a single network. That network will be the primary, unless the primary is known to be in error and the secondary is not. In that case, the request will go out the secondary. If a Write acknowledgement is not received within a request timeout period, the request will be retried on the other network. If a Write acknowledgement is still not received after another request timeout period, the driver will send the request again, but this time out both networks. The driver will continue to retry the request on both networks until an acknowledgement is received or the maximum number of retry attempts have been performed.

**Note:** The device may take up to 4 scan cycles to complete a write request. The controller may be programmed to use a scan period as large as 400ms. Set the request timeout accordingly. Setting it too low may cause the driver to assume a Write has failed even though it just hasn't been processed by the controller yet.

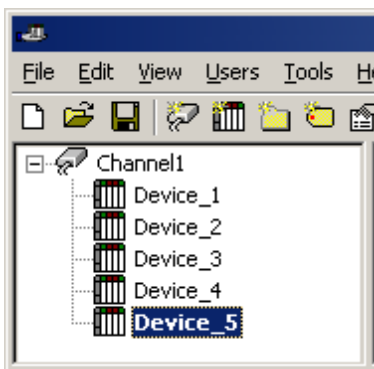
## Optimizing Your Triconex Ethernet Communications

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

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

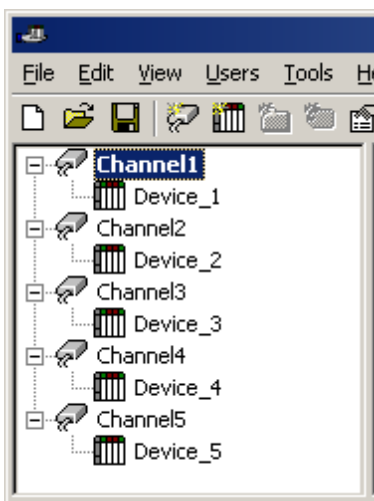
It's important to understand how the Triconex Ethernet driver Reads and Writes data. This driver is unlike most others in the two following ways:

1. Instead of sending Read requests to the devices, the driver tells each device to send periodic data updates whose data is then stored in cache. When the server asks the driver to read a piece of data, the driver will return the last value stored in cache. This method is more efficient than polling for Triconex controllers.
2. The Triconex Ethernet driver can service Write requests to multiple devices on a channel simultaneously. The driver must, however, wait for the last Write request pending on that channel to be completed before allowing the next set of writes to be issued (or any Reads to be performed). Since a Triconex device can take up to 4 scan cycles to complete a Write, and the scan period can be as large as 400 ms, Writes can have a tremendous effect on overall performance.



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

If the Triconex Ethernet driver could only define one single channel, then the example shown above would be the only option available; however, the Triconex Ethernet driver can define up to 32 channels. Using multiple channels distributes the data collection workload by simultaneously issuing multiple requests to the network. An example of how the same application may appear when configured using multiple channels to improve performance is shown below.



Each device has now been defined under its own channel. In this new configuration, a single path of execution is dedicated to the task of gathering data from each device. If the application has 32 or fewer devices, it can be optimized exactly how it is shown here.

The performance will improve even if the application has more than X32 devices. While 32 or fewer devices may be ideal, the application will still benefit from additional channels. Although spreading the device load across all channels will cause the server to move from device to device again, it can now do so with far less devices to process on a single channel.

Because Writes control the overall performance of the Triconex Ethernet driver, it is important to consider how the client application writes to the server. If many writes must be performed, it may be better to send them at nearly the same time so that the driver can block them together. As discussed above, writes can be sent to multiple devices even if they are on the same channel. Writes to multiple points in a given device can also be blocked in a single request, if the client issues them at nearly the same time. Users should be familiar with the server's Write optimization features, which can be accessed from the Channel Properties.

**Note:** For an application where writes never occur, there is little advantage to distributing the devices over multiple channels because the devices are not polled.

## Tag Import

---

The Triconex Ethernet driver has the ability to automatically create tags for all of the variables in the TriStation project. Follow the instructions below for information on how to automatically generate tags.

1. Export the project from **TriStation** using **XML format**. Then, save the file on the computer that will be running the driver.
2. Next, either create a new Triconix OPC server project or load an existing one.
3. Invoke the **Device Properties** of the device for which tags will be generated.
4. Select the **Tag Import** tab and then enter the file's name and path. Specify the node name associated with the tags that will be imported.
5. Click **OK**.
6. Next, open **Device Properties | Database Creation**. Click **Auto Create**. The driver will read the XML file and then generate the required tags.
7. Save the OPC server project.

**See Also:** [Tag Import Device Properties](#)

## Data Types Description

---

| Data Type | Description   |
|-----------|---|
| Boolean   | Single bit  |
| Long      | Signed 32 bit value<br>bit 0 is the low bit<br>bit 30 is the high bit<br>bit 31 is the sign bit |
| DWord     | Unsigned 32 bit value<br>bit 0 is the low bit<br>bit 31 is the high bit                         |
| Float     | 32 bitreal  |

## 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.

[Tricon](#)  
[Trident](#)

**Note:** The driver has several system tags that can be used to monitor the driver's operation. For more information, refer to [Driver System Tags](#).

## Tricon Addressing

The Tricon model supports the following addresses. Memory may be addressed using either Bin:Offset syntax or Modbus aliases. The default data types for dynamically defined tags are shown in **bold** where appropriate.

### Bin:Offset Addressing

| Variable Type        | Address Range                            | Data Type                             | Access     |
|----------------------|--|---------------------------------------|------------|
| Output (BOOL)        | 00:0000-00:1999                          | Boolean                               | Read/Write |
| Memory (BOOL)        | 01:0000-01:1999                          | Boolean                               | Read/Write |
| Input (BOOL)         | 02:0000-02:1999                          | Boolean                               | Read Only  |
| Memory (BOOL)        | 03:0000-03:1999                          | Boolean                               | Read Only  |
| Input (REAL)         | 04:0000-04:0999<br>04:xxxx.00-04:xxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read Only  |
| Memory (DINT)        | 05:0000-05:0999<br>05:xxxx.00-05:xxxx.31 | <b>DWord</b> , Long, Float<br>Boolean | Read Only  |
| Input (REAL)         | 06:0000-06:0119<br>06:xxxx.00-06:xxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read Only  |
| Memory (REAL)        | 07:0000-07:0999<br>07:xxxx.00-07:xxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read Only  |
| System Status (BOOL) | 08:0000-08:5998                          | Boolean                               | Read Only  |
| System Status (DINT) | 09:0000-09:0368<br>09:xxxx.00-09:xxxx.31 | <b>DWord</b> , Long, Float<br>Boolean | Read Only  |
| Output (REAL)        | 10:0000-10:0249<br>10:xxxx.00-10:xxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read/Write |
| Memory (DINT)        | 11:0000-11:0749<br>11:xxxx.00-11:xxxx.31 | <b>DWord</b> , Long, Float<br>Boolean | Read/Write |
| Memory (REAL)        | 12:0000-12:0999<br>12:xxxx.00-12:xxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read/Write |

### Modbus Alias Addressing

| Variable Type        | Address Range                    | Data Type                             | Access     |
|----------------------|----------------------------------|---------------------------------------|------------|
| Output (BOOL)        | 00001-02000                      | Boolean                               | Read/Write |
| Memory (BOOL)        | 02001-04000                      | Boolean                               | Read/Write |
| Input (BOOL)         | 10001-12000                      | Boolean                               | Read Only  |
| Memory (BOOL)        | 12001-14000                      | Boolean                               | Read Only  |
| Input (REAL)         | 30001-32000<br>xxxxx.00-xxxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read Only  |
| Memory (DINT)        | 31001-32000<br>xxxxx.00-xxxxx.31 | <b>DWord</b> , Long, Float<br>Boolean | Read Only  |
| Input (REAL)         | 32001-32120<br>xxxxx.00-xxxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read Only  |
| Memory (REAL)        | 33001-34000<br>xxxxx.00-xxxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read Only  |
| System Status (BOOL) | 14001-19999                      | Boolean                               | Read Only  |
| System Status (DINT) | 39631-39999<br>xxxxx.00-xxxxx.15 | <b>Word</b> , Short<br>Boolean        | Read Only  |
| Output (REAL)        | 40001-40250<br>xxxxx.00-xxxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read/Write |

|               |                                  |                                       |            |
|---------------|----------------------------------|---------------------------------------|------------|
| Memory (DINT) | 40251-41000<br>xxxxx.00-xxxxx.31 | <b>DWord</b> , Long, Float<br>Boolean | Read/Write |
| Memory (REAL) | 41001-42000<br>xxxxx.00-xxxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read/Write |

**Note:** DINT system status variables, bin 9 data, are addressed as 16-bit registers when using Modbus Aliases. All other DINT and REAL variable types are addressed as 32 bitregisters when using Modbus Aliases.

### Examples

The following examples show the relationship between some Modbus Aliases and Bin:Offset addresses.

1. 39633 and 9:1 both address the system variable Year.
2. 39635 and 9:2 both address the system variable Month.
3. 39663 addresses the low word of the DINT value addressed by 9:16. This is the current control program version number system variable. The low word is the minor version number.
4. 39664 addresses the high word of the DINT value addressed by 9:16. This is the current control program version number system variable. The high work is the major version number.
5. 00001 and 0:0 both address the same BOOL output.
6. 00002 and 0:1 both address the same BOOL output.
7. 30001 and 4:0 both address the same REAL input.
8. 30002 and 4:1 both address the same REAL input.

### Write Requests

For the device to process Write requests, the following statements must be true.

1. The variables must be defined as Read/Write.
2. For Tricon, the keyswitch must be in the Program or Remote position. For Trident, the Remote mode must be enabled by setting a system attribute.
3. The TriStation configuration setting (Disable Remote Changes to Outputs) which was downloaded to the controller must allow remote changes.

**Note:** This setting only affects changes to discrete outputs and analog outputs.

4. The TriStation configuration for the ACM or NCM (Tricon) and CM (Trident) must be configured as Read/Write.

If the controller cannot write the data, it sends as reject response coded. For more information, refer to [Write failed for tag '<address>' on device '<device>'. TSAA Response Code: <code>](#).

### Basic Address Example

Users may address values using the Bin:Offset syntax or with a Modbus Alias.

10:0012 addresses the bin 10 float value with offset 12.  
The same value can be addressed using the Modbus Alias 40013.

### Bit Address Example

Individual bits in DINT types may be addressed by appending the "dot bit number" to a basic address.

05:0010.2 addresses bit 2 of the bin 5 DINT value with offset 10.  
The same bit can be addressed using the Modbus Alias 31011.2.

### Array Address Examples

Multiple data points may be accessed as array data.

1. 05:0010 [4] addresses bin 5 DINT values with offsets 10, 11, 12, and 13.  
The same range of values can be addressed using the Modbus Alias 31011 [4].
2. 05:0010 [4][5] addresses bin 5 DINT values with offsets from 10, to 29 as a 2-D array.  
The same range of values can be addressed using the Modbus Alias 31011 [4][5].

## Trident Addressing

The Trident model supports the following addresses. Memory can be addressed using either Bin:Offset syntax or Modbus aliases. The default data types for dynamically defined tags are shown in **bold** where appropriate.

### Bin:Offset Addressing

| Variable Type | Address Range                            | Data Type                             | Access     |
|---------------|--|---------------------------------------|------------|
| Output (BOOL) | 00:0000-00:1999                          | Boolean                               | Read/Write |
| Memory (BOOL) | 01:0000-01:1999                          | Boolean                               | Read/Write |
| Input (BOOL)  | 02:0000-02:1999                          | Boolean                               | Read Only  |
| Memory (BOOL) | 03:0000-03:1999                          | Boolean                               | Read Only  |
| Input (REAL)  | 04:0000-04:0999<br>04:xxxx.00-04:xxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read Only  |
| Memory (DINT) | 05:0000-05:0999<br>05:xxxx.00-05:xxxx.31 | <b>DWord</b> , Long, Float<br>Boolean | Read Only  |
| Input (REAL)  | 06:0000-06:0119<br>06:xxxx.00-06:xxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read Only  |
| Memory (REAL) | 07:0000-07:0999<br>07:xxxx.00-07:xxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read Only  |
| Output (REAL) | 10:0000-10:0249<br>10:xxxx.00-10:xxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read/Write |
| Memory (DINT) | 11:0000-11:0749<br>11:xxxx.00-11:xxxx.31 | <b>DWord</b> , Long, Float<br>Boolean | Read/Write |
| Memory (REAL) | 12:0000-12:0999<br>12:xxxx.00-12:xxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read/Write |

### Modbus Alias Addressing

| Variable Type | Address Range                    | Data Type                             | Access     |
|---------------|----------------------------------|---------------------------------------|------------|
| Output (BOOL) | 00001-02000                      | Boolean                               | Read/Write |
| Memory (BOOL) | 02001-04000                      | Boolean                               | Read/Write |
| Input (BOOL)  | 10001-12000                      | Boolean                               | Read Only  |
| Memory (BOOL) | 12001-14000                      | Boolean                               | Read Only  |
| Input (REAL)  | 30001-32000<br>xxxxx.00-xxxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read Only  |
| Memory (DINT) | 31001-32000<br>xxxxx.00-xxxxx.31 | <b>DWord</b> , Long, Float<br>Boolean | Read Only  |
| Input (REAL)  | 32001-32120<br>xxxxx.00-xxxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read Only  |
| Memory (REAL) | 33001-34000<br>xxxxx.00-xxxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read Only  |
| Output (REAL) | 40001-40250<br>xxxxx.00-xxxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read/Write |
| Memory (DINT) | 40251-41000<br>xxxxx.00-xxxxx.31 | <b>DWord</b> , Long, Float<br>Boolean | Read/Write |
| Memory (REAL) | 41001-42000<br>xxxxx.00-xxxxx.31 | DWord, Long, <b>Float</b><br>Boolean  | Read/Write |

**Note:** All DINT and REAL variable types are addressed as 32 bitregisters when using Modbus Aliases.

### Examples

The following examples show the relationship between some Modbus Aliases and Bin:Offset addresses.

- 00001 and 0:0 both address the same BOOL output.
- 00002 and 0:1 both address the same BOOL output.
- 30001 and 4:0 both address the same REAL input.
- 30002 and 4:1 both address the same REAL input.

### Write Requests

For the device to process Write requests, the following statements must be true.

1. The variables must be defined as Read/Write.
2. For Tricon, the keyswitch must be in the Program or Remote position. For Trident, the Remote mode must be enabled by setting a system attribute.
3. The TriStation configuration setting (Disable Remote Changes to Outputs) which was downloaded to the controller must allow remote changes.

**Note:** This setting only affects changes to discrete outputs and analog outputs.

4. The TriStation configuration for the ACM or NCM (Tricon) and CM (Trident) must be configured as Read/Write.

If the controller cannot write the data, it sends as reject response coded. For more information, refer to [Write failed for tag '<address>' on device '<device>'. TSAA Response Code: <code>](#).

### Basic Address Example

Users may address values using the Bin:Offset syntax or with a Modbus Alias.

10:0012 addresses the bin 10 float value with offset 12.  
The same value can be addressed using the Modbus Alias 40013.

### Bit Address Example

Individual bits in DINT types may be addressed by appending the "dot bit number" to a basic address.

05:0010.2 addresses bit 2 of the bin 5 DINT value with offset 10.  
The same bit can be addressed using the Modbus Alias 31011.2.

### Array Address Examples

Multiple data points may be accessed as array data.

1. 05:0010 [4] addresses bin 5 DINT values with offsets 10, 11, 12, and 13.  
The same range of values can be addressed using the Modbus Alias 31011 [4].
2. 05:0010 [4][5] addresses bin 5 DINT values with offsets from 10, to 29 as a 2-D array.  
The same range of values can be addressed using the Modbus Alias 31011 [4][5].

### Driver System Tags

Tags may be created with the following addresses to monitor driver operation.

| Address               | Data Type | Access    | Description   |
|-----------------------|-----------|-----------|---|
| PrimaryNetworkError   | Boolean   | Read Only | Set to 0 if the state of the primary network appears to be good or is uncertain.<br><br>Set to 1 if the state of the primary network appears to be bad.<br>Set when an expected message is not received.                                      |
| SecondaryNetworkError | Boolean   | Read Only | Set to 0 if the state of the primary network appears to be good or is uncertain.<br><br>Set to 1 if the state of the primary network appears to be bad.<br>Set when an expected message is not received or network redundancy is not enabled. |
| UpdatesFromPrimary    | Boolean   | Read Only | Set to 1 when the driver is processing data updates from the primary network. (Updates may also be coming in on secondary network, but are ignored by driver.)  |
| UpdatesFromSecondary  | Boolean   | Read Only | Set to 1 when the driver is processing data updates from the secondary network. (Updates may also be coming in on primary network, but are ignored by driver.)  |

## 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](#)

[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

[Could not detect available network adapters for <primary/secondary> network](#)

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

[Failed to load Winsock or related library](#)

[Tag import error - invalid bin in item path: '<item path>'](#)

[Tag import error - invalid item path format: '<item path>'](#)

[Tag import error - invalid name: '<tag name>'](#)

[Tag import error - invalid offset in item path: '<item path>'](#)

[Tag import failed. Could not find data for node '<node>'](#)

[Tag import failed. Could not open file '<file>'](#)

[Tag import failed due to unexpected XML format in '<file>'](#)

[Tag import failed. File name given, but no node name](#)

[Tag import failed. No import file name or system tags specified](#)

[Unable to bind to local address \(<IP, port>\)](#)

[Unable to process data update on device '<device>'. CRC error](#)

[Unable to process data update on device '<device>'. Unexpected message format](#)

[Unable to process data update on device '<device>'. Unexpected model](#)

[Unable to process data update on device '<device>'. Unexpected node number](#)

[Unable to use network redundancy on channel '<channel>'](#)

[Unable to use specified primary network adapter '<adapter>' on channel '<channel>'. Using '<adapter>'](#)

[Unable to use specified secondary network adapter '<adapter>' on channel '<channel>'. Using '<adapter>'](#)

[Winsock initialization failed \(OS Error = <error code>\)](#)

[Winsock V1.1 or higher must be installed to use the Triconex Ethernet device driver](#)

[Write failed for tag '<address>' on device '<device>'. CRC error](#)

[Write failed for tag '<address>' on device '<device>'. No data for read/modify/write](#)

[Write failed for tag '<address>' on device '<device>'. Timeout](#)

[Write failed for tag '<address>' on device '<device>'. TSAA Response Code: <code>](#)

[Write failed for tag '<address>' on device '<device>'. Unexpected model](#)

[Write failed for tag '<address>' on device '<device>'. Unexpected node number](#)

[Write failed for tag '<address>' on device '<device>'. Unexpected response format](#)

### 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](#)

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

**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:**

1. Verify that the address is correct; if it is not, re-enter it in the client application.
2. 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.

## Device Status Messages

---

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

### Device Status Messages

[Could not detect available network adapters for <primary/secondary> network](#)

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

[Failed to load Winsock or related library](#)

[Tag import error - invalid bin in item path: '<item path>'](#)

[Tag import error - invalid item path format: '<item path>'](#)

[Tag import error - invalid name: '<tag name>'](#)

[Tag import error - invalid offset in item path: '<item path>'](#)

[Tag import failed. Could not find data for node '<node>'](#)

[Tag import failed. Could not open file '<file>'](#)

[Tag import failed due to unexpected XML format in '<file>'](#)

[Tag import failed. File name given, but no node name](#)

[Tag import failed. No import file name or system tags specified](#)

[Unable to bind to local address \(<IP, port>\)](#)

[Unable to process data update on device '<device>'. CRC error](#)

[Unable to process data update on device '<device>'. Unexpected message format](#)

[Unable to process data update on device '<device>'. Unexpected model](#)

[Unable to process data update on device '<device>'. Unexpected node number](#)

[Unable to use network redundancy on channel '<channel>'](#)

[Unable to use specified primary network adapter '<adapter>' on channel '<channel>'. Using '<adapter>'](#)

[Unable to use specified secondary network adapter '<adapter>' on channel '<channel>'. Using '<adapter>'](#)

[Winsock initialization failed \(OS Error = <error code>\)](#)

[Winsock V1.1 or higher must be installed to use the Triconex Ethernet device driver](#)

[Write failed for tag '<address>' on device '<device>'. CRC error](#)

[Write failed for tag '<address>' on device '<device>'. No data for read/modify/write](#)

[Write failed for tag '<address>' on device '<device>'. Timeout](#)

[Write failed for tag '<address>' on device '<device>'. TSAA Response Code: <code>](#)

[Write failed for tag '<address>' on device '<device>'. Unexpected model](#)

[Write failed for tag '<address>' on device '<device>'. Unexpected node number](#)

[Write failed for tag '<address>' on device '<device>'. Unexpected response format](#)

### Could not detect available network adapters for <primary/secondary> network

---

**Error Type:**

Serious

**Possible Cause:**

The network adapters are not connected to a hub or other device.

**Solution:**

The network adapters must be connected to a device or hub in order for Windows to correctly report their configuration to the driver. Verify network connection.

### Device '<device name>' is not responding

---

**Error Type:**

Serious

**Possible Cause:**

1. The network connection between the device and the host PC is broken.
2. The named device may have been assigned an incorrect IP, Device ID or model.
3. An expected data update was late.

**Solution:**

1. Verify the cabling between the PC and the PLC device.
2. Verify the Device Properties.

3. Consider increasing the Bin Data Update Period if the device is not capable of sending updates at the requested rate.

**See Also:**

[Communications Device Properties](#)

**Failed to load Winsock or related library**

---

**Error Type:**

Fatal

**Possible Cause:**

The driver could not load Winsock (winsock.dll) or the IP Helper API dll (iphlpapi.dll).

**Solution:**

These libraries are normally installed with Windows. Verify that these files exist in the system folder.

**Tag import error - invalid bin in item path: '<item path>'**

---

**Error Type:**

Warning

**Possible Cause:**

An entry in the import file had an item path with an invalid bin number.

**Solution:**

Verify that the item path has the format <node name>:<bin>:<offset> and that the bin number is between 0 and 12.

**Tag import error - invalid item path format\: '<item path>'**

---

**Error Type:**

Warning

**Possible Cause:**

An entry in the import file had an unexpected item path format.

**Solution:**

Verify that the item paths in the export file have the format <node name>:<bin>:<offset>.

**Tag import error - invalid name: '<tag name>'**

---

**Error Type:**

Warning

**Possible Cause:**

The **PSTAlias** name value does not conform to the server's tag naming rules.

**Solution:**

1. Verify that the name contains only alpha numeric characters and underscores.
2. Verify that the first character is not an underscore. Edit the import file if necessary.

**Note:**

There should be two entries for each tag. The first entry gives the tag name and the second gives the Modbus alias. The driver skips the entries with Modbus alias values.

**Tag import error - invalid offset in item path: '<item path>'**

---

**Error Type:**

Warning

**Possible Cause:**

An entry in the import file had an item path with an invalid offset value.

**Solution:**

Verify that the item path has the format <node name>:<bin>:<offset> and that the offset is within the valid range for the bin.

**See Also:**

[Address Descriptions](#)

---

**Tag import failed. Could not find data for node '<node>'**

---

**Error Type:**

Warning

**Possible Cause:**

The driver did not find any entires in the import file with the specified node name.

**Solution:**

Verify that the correct node name has been specified.

**Note:**

The node name is given by the PSTAliasGroup name.

---

**Tag import failed. Could not open file '<file>'**

---

**Error Type:**

Warning

**Possible Cause:**

The driver could not find or open the import file specified on the Tag Import page in Device Properties.

**Solution:**

1. Verify that the file exists and that the specified path is correct.
2. Make sure that no other application has opened the file.

**See Also:**

[Tag Import Device Properties](#)

---

**Tag import failed due to unexpected XML format in '<file>'**

---

**Error Type:**

Warning

**Possible Cause:**

The driver could not process the import file because it did not have the expected format.

**Solution:**

Verify that the data is exported from TriStation in XML format.

---

**Tag import failed. File name given, but no node name**

---

**Error Type:**

Warning

**Possible Cause:**

No node name was specified on the Tag Import page in Device Properties.

**Solution:**

Specify the node name used in the import file.

**See Also:**

[Tag Import Device Properties](#)

---

**Tag import failed. No import file name or system tags specified**

---

**Error Type:**

Warning

**Possible Cause:**

No tag import options were specified in the Tag Import page in Device Properties when the driver attempted to generate a tag database.

**Solution:**

Verify that an import file name is specified or that one of the system tag options is selected when tag generation is started.

**See Also:**

[Tag Import Device Properties](#)

**Unable to bind to local address (<IP, port>)**

---

**Error Type:**

Fatal

**Possible Cause:**

The driver was not able to bind a socket to receive data sent to the specified IP and port.

**Solution:**

Make sure no other application is using that IP and port.

**Unable to process data update on device '<device>'. CRC error**

---

**Error Type:**

Warning

**Possible Cause:**

The driver rejected a data update message because its CRC check sum was not valid.

**Solution:**

This error implies noise in the network. If this error is persistent, efforts should be made to improve network reliability.

**Unable to process data update on device '<device>'. Unexpected message format**

---

**Error Type:**

Warning

**Possible Cause:**

The driver rejected a data update message because it did not have the expected format.

**Solution:**

If the problem persists and does not appear to be network related, perform a Channel Diagnostics capture of the incoming update messages and contact Technical Support.

**Unable to process data update on device '<device>'. Unexpected model**

---

**Error Type:**

Warning

**Possible Cause:**

The driver rejected a data update message because it did not have the expected model number.

**Solution:**

Verify the model setting.

**See Also:**

[Device Setup](#)

**Unable to process data update on device '<device>'. Unexpected node number**

---

**Error Type:**

Warning

**Possible Cause:**

The driver rejected a data update message because it did not have the expected node number.

**Solution:**

Verify the Device ID setting.

**See Also:**

[Device Setup](#)

---

**Unable to use network redundancy on channel '<channel>'**

---

**Error Type:**

Warning

**Possible Cause:**

1. The driver loaded a project that was created on a computer which had multiple
2. The network adapter cards are not connected to a device or hub.

**Solution:**

1. Reset the Network Interface Channel Properties to be compatible with the host computer. In order to use network redundancy, the host computer must have 2 network interface cards (NICs) installed. Save the adjusted setting.
2. The network adapters must be connected to a device or hub in order for Windows to correctly report their configuration to the driver. Ensure that the adapters are connected.

**See Also:**

[Network Interface Channel Properties](#)

[Network Redundancy](#)

---

**Unable to use specified primary network adapter '<adapter>' on channel '<channel>'**

---

**Error Type:**

Warning

**Possible Cause:**

1. The driver loaded a project that was created on another computer with different network interface hardware or IP address.
2. The network adapter cards are not connected to a device or hub.

**Solution:**

1. Reset the Network Interface Channel Properties to be compatible with the host computer. In order to use network redundancy, the host computer must have 2 network interface cards (NICs) installed. Save the adjusted setting.
2. The network adapters must be connected to a device or hub in order for Windows to correctly report their configuration to the driver. Ensure that the adapters are connected.

**See Also:**

[Network Interface Channel Properties](#)

[Network Redundancy](#)

---

**Unable to use specified secondary network adapter '<adapter>' on channel '<channel>'**

---

**Error Type:**

Warning

**Possible Cause:**

1. The driver loaded a project that was created on another computer with different network interface hardware or IP address.
2. The network adapter cards are not connected to a device or hub.

**Solution:**

1. Reset the Network Interface Channel Properties to be compatible with the host computer. In order to use network redundancy, the host computer must have 2 network interface cards (NICs) installed. Save the adjusted setting.
2. The network adapters must be connected to a device or hub in order for Windows to correctly report their configuration to the driver. Ensure that the adapters are connected.

**See Also:**

[Network Interface Channel Properties](#)

[Network Redundancy](#)

### **Winsock initialization failed (OS Error = <error code>)**

---

**Error Type:**

Fatal

| OS Error | Indication   | Possible Solution  |
|----------|--|--|
| 10091    | Indicates that the underlying network subsystem is not ready for network communication.        | Wait a few seconds and restart the driver.                                       |
| 10067    | Limit on the number of tasks supported by the Windows Sockets implementation has been reached. | Close one or more applications that may be using Winsock and restart the driver. |

### **Winsock V1.1 or higher must be installed to use the Triconex Ethernet device driver**

---

**Error Type:**

Fatal

**Possible Cause:**

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

**Solution:**

Upgrade Winsock to version 1.1 or higher.

### **Write failed for tag '<address>' on device '<device>'. CRC error**

---

**Error Type:**

Warning

**Possible Cause:**

The driver rejected a write complete message because its CRC check sum was not valid.

**Solution:**

This error implies noise in the network. If this error is persistent, efforts should be made to improve network reliability.

### **Write failed for tag '<address>' on device '<device>'. No data for Read/Modify/Write**

---

**Error Type:**

Warning

**Possible Cause:**

Bit writes require a read/modify/write procedure. The last known value has bad quality.

**Solution:**

Ensure that the driver has had time to receive a data update for the value in question before bit writes are performed. Failure can occur before an initial data update has been received or following a network failure.

### **Write failed for tag '<address>' on device '<device>'. Timeout**

---

**Error Type:**

Warning

**Possible Cause:**

The driver did not receive a write complete message after the request timeout period and number of retries.

**Solution:**

Triconex devices can take a long time to process Writes. Thus, a timeout error does not necessarily mean the device did not receive the write request or has failed to process it. Ensure that an ample request timeout period has been specified that is consistent with the device's scan rate.

**Note:**

Other reasons for timeout errors could be related to hardware or network failure.

**See Also:**

[Device Setup](#)

---

### Write failed for tag '<address>' on device '<device>'. TSAA Response Code: <code>

---

**Error Type:**

Warning

**Possible Cause:**

The device could not complete the write request for the reason given by the response code and reason.

**Solution:**

Proper solution depend on failure reason. For more information, refer to the table below.

| Code | Description   |
|------|---|
| 1    | No buffer available to process the request.   |
| 2    | Bin number specified in the request was out of range.   |
| 3    | The Tricon NCM or Trident CM module is busy processing previous request and cannot accept another request. This can happen if more than four write request are outstanding at one time. |
| 4    | No MP is running.   |
| 5    | TSX has rejected the request. the Reason contains the specific reason.  |
| 6    | Request to TSX timed out.   |
| 7    | Invalid response from TX.   |
| 8    | Message was too big.  |
| 9    | Offset or number of values in the request was invalid.  |
| 10   | No control program (TriStation applications).   |
| 11   | Read Only port.   |
| 236  | Bad SOE number.   |
| 237  | Invalid SOE type.   |
| 238  | Invalid SOE state.  |

---

### Write failed for tag '<address>' on device '<device>'. Unexpected model

---

**Error Type:**

Warning

**Possible Cause:**

The driver rejected a Write complete message because it did not have the expected model number.

**Solution:**

Verify the model setting.

**See Also:**

[Device Setup](#)

---

### Write failed for tag '<address>' on device '<device>'. Unexpected node number

---

**Error Type:**

Warning

**Possible Cause:**

The driver rejected a write complete message because it did not have the expected node number.

**Solution:**

Verify the Device ID setting.

**See Also:**

[Device Setup](#)

**Write failed for tag '<address>' on device '<device>'. Unexpected response format**

---

**Error Type:**

Warning

**Possible Cause:**

The driver rejected a write complete message because it did not have the expected format.

**Solution:**

If the problem persists and does not appear to be network related, perform a Channel Diagnostics capture of the incoming update messages and contact Technical Support.

# Index

## A

|   |        |
|---|--------|
| Address '<address>' is out of range for the specified device or register..... | 21     |
| Address Descriptions.....   | 17     |
| Address Validation.....   | 21     |
| Array Address.....  | 18, 20 |

## B

|                  |        |
|------------------|--------|
| Bit Address..... | 18, 20 |
| Boolean.....     | 16     |

## C

|  |    |
|--|----|
| Cable Diagram for Trident.....   | 10 |
| Cable Diagrams for Tricon.....   | 8  |
| CM Configuration Device Properties.....  | 6  |
| Communication Protocol.....  | 5  |
| Communications Device Properties.....  | 6  |
| Could not detect available network adapters for <primary/secondary> network..... | 23 |

## D

|  |        |
|--|--------|
| Data Type '<type>' is not valid for device address '<address>'.....      | 22     |
| Data Types Description.....  | 16     |
| Device '<device name>' is not responding.....                            | 23     |
| Device address '<address>' contains a syntax error.....                  | 22     |
| Device address '<address>' is not supported by model '<model name>'..... | 22     |
| Device address '<address>' is Read Only.....                             | 22     |
| Device ID.....   | 5      |
| Device Setup.....  | 5      |
| Device Status Messages.....  | 23     |
| Device System Tags.....  | 8      |
| Driver System Tags.....  | 12, 20 |

**DWord**..... 16

## **E**

**Enable Network Redundancy**..... 5

**Error Descriptions**..... 21

## **F**

**Failed to load Winsock or related library**..... 24

**Float**..... 16

## **I**

**Import File**..... 8

## **L**

**Long**..... 16

## **M**

**Missing address**..... 22

## **N**

**Network Interface Channel Properties**..... 5

**Network Redundancy**..... 11

**Node Name**..... 8

**Node Number**..... 5

**O**

|   |    |
|---|----|
| OPC Data Quality.....                                 | 7  |
| Optimizing Your Triconex Ethernet Communications..... | 13 |
| Overview.....   | 4  |

**P**

|                              |   |
|------------------------------|---|
| Primary Network Adapter..... | 6 |
|------------------------------|---|

**R**

|                      |    |
|----------------------|----|
| Reads.....           | 11 |
| Request Timeout..... | 5  |

**S**

|                                |   |
|--------------------------------|---|
| Secondary Network Adapter..... | 6 |
|--------------------------------|---|

**T**

|  |    |
|--|----|
| Tag Generation.....  | 8  |
| Tag Import.....  | 15 |
| Tag Import Device Properties.....                                    | 8  |
| Tag import error - invalid bin in item path/<br>'<item path>'.....   | 24 |
| Tag import error - invalid item path format: '<item path>'.....      | 24 |
| Tag import error - invalid name: '<tag name>'.....                   | 24 |
| Tag import error - invalid offset in item path: '<item path>'.....   | 24 |
| Tag import failed due to unexpected XML format in '<file>'.....      | 25 |
| Tag import failed. Could not find data for node '<node>'.....        | 25 |
| Tag import failed. Could not open file '<file>'.....                 | 25 |
| Tag import failed. File name given, but no node name.....            | 25 |
| Tag import failed. No import file name or system tags specified..... | 25 |
| Timestamp.....   | 7  |

|  |    |
|--|----|
| Tricon Addressing.....                         | 17 |
| Triconex System Access Application (TSAA)..... | 5  |
| Trident Addressing.....                        | 19 |
| TriStation.....                                | 15 |

## U

|   |    |
|---|----|
| Unable to bind to local address (<IP, port>).....   | 26 |
| Unable to process data update on device '<device>'. CRC error.....                        | 26 |
| Unable to process data update on device '<device>'. Unexpected message format.....        | 26 |
| Unable to process data update on device '<device>'. Unexpected model.....                 | 26 |
| Unable to process data update on device '<device>'. Unexpected node number.....           | 26 |
| Unable to use network redundancy on channel '<channel>'.....                              | 27 |
| Unable to use specified primary network adapter '<adapter>' on channel '<channel>'.....   | 27 |
| Unable to use specified secondary network adapter '<adapter>' on channel '<channel>'..... | 27 |
| Update Rate.....  | 7  |

## W

|   |    |
|---|----|
| Winsock initialization failed (OS Error = <error code>).....                              | 28 |
| Winsock V1.1 or higher must be installed to use the Triconex Ethernet device driver.....  | 28 |
| Write failed for tag '<address>' on device '<device>'. CRC error.....                     | 28 |
| Write failed for tag '<address>' on device '<device>'. No data for Read/Modify/Write..... | 28 |
| Write failed for tag '<address>' on device '<device>'. Timeout.....                       | 28 |
| Write failed for tag '<address>' on device '<device>'. TSAA Response Code: <code>.....    | 29 |
| Write failed for tag '<address>' on device '<device>'. Unexpected model.....              | 29 |
| Write failed for tag '<address>' on device '<device>'. Unexpected node number.....        | 29 |
| Write failed for tag '<address>' on device '<device>'. Unexpected response format.....    | 30 |
| Writes.....   | 12 |

## X

|          |       |
|----------|-------|
| XML..... | 8, 15 |
|----------|-------|