Toshiba Ethernet Driver

Table of Contents

Toshiba Ethernet Driver

Help version 1.021

CONTENTS

Overview

The Toshiba Ethernet Driver provides a reliable way to connect Toshiba Ethernet devices to OPC client applications; including HMI, SCADA, Historian, MES, ERP, and countless custom applications.

It is important to understand the data types and the addressing scheme before using the Toshiba Ethernet Driver in an OPC server software project. *For more information, refer to **Data Types Description** and **Address Descriptions**.*

Setup

Supported Devices

T2, T3, S2, and S3 PLCs with Ethernet option.

Communications Protocol

Toshiba ASCII Computer Link

Note: This driver requires Winsock V1.1 or higher.

Connection Timeout

This property specifies the time that the driver will wait for a connection to be made with a device. Depending on network load, the connect time may vary with each connection attempt. The default setting is 3 seconds. The valid range is 1 to 30 seconds.

Request Timeout

This property specifies the time that the driver will wait on a response from the device before giving up and going on to the next request. Longer timeouts only affect performance if a device is not responding. The default setting is 1000 milliseconds. The valid range is 50 to 9999999 milliseconds.

Retry Attempts

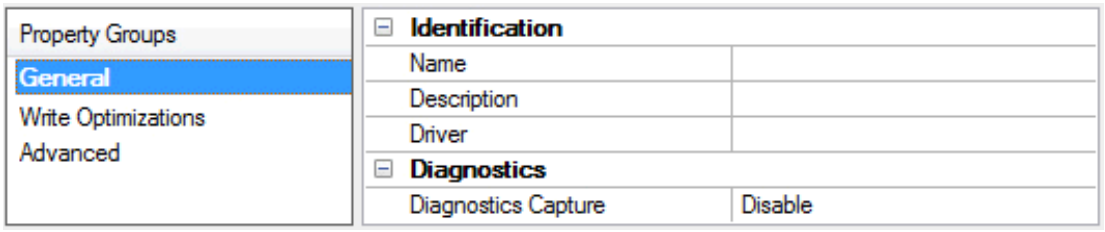
This property specifies the number of times the driver will retry a message before giving up and going on to the next message. The default setting is 3 retries. The valid range is 1 to 10.

Device ID

This property specifies the device's IP address.

Channel Properties - General

This server supports the use of simultaneous multiple communications drivers. Each protocol or driver used in a server project is called a channel. A server project may consist of many channels with the same communications driver or with unique communications drivers. A channel acts as the basic building block of an OPC link. This group is used to specify general channel properties, such as the identification attributes and operating mode.



Identification

Name: User-defined identity of this channel. In each server project, each channel name must be unique. Although names can be up to 256 characters, some client applications have a limited display window when browsing the OPC server's tag space. The channel name is part of the OPC browser information.
For information on reserved characters, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in the server help.

Description: User-defined information about this channel.
Many of these properties, including Description, have an associated system tag.

Driver: Selected protocol / driver for this channel. This property specifies the device driver that was selected during channel creation. It is a disabled setting in the channel properties.

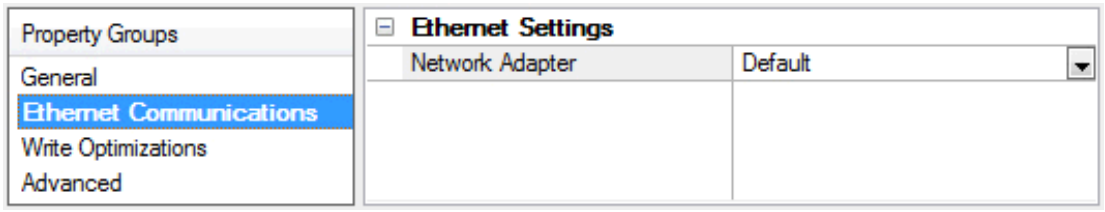
Note: With the server's online full-time operation, these properties can be changed at any time. This includes changing the channel name to prevent clients from registering data with the server. If a client has already acquired an item from the server before the channel name is changed, the items are unaffected. If, after the channel name has been changed, the client application releases the item and attempts to re-acquire using the old channel name, the item is not accepted. With this in mind, changes to the properties should not be made once a large client application has been developed. Utilize the User Manager to prevent operators from changing properties and restrict access rights to server features.

Diagnostics

Diagnostics Capture: When enabled, this option makes the channel's diagnostic information available to OPC applications. Because the server's diagnostic features require a minimal amount of overhead processing, it is recommended that they be utilized when needed and disabled when not. The default is disabled.
Note: This property is disabled if the driver does not support diagnostics.
For more information, refer to "Communication Diagnostics" in the server help.

Channel Properties - Ethernet Communications

Ethernet Communication can be used to communicate with devices.

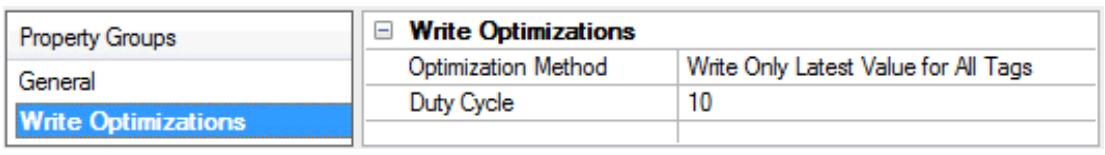


Ethernet Settings

Network Adapter: Specify the network adapter to bind. When Default is selected, the operating system selects the default adapter.

Channel Properties - Write Optimizations

As with any OPC server, writing data to the device may be the application's most important aspect. The server intends to ensure that the data written from the client application gets to the device on time. Given this goal, the server provides optimization properties that can be used to meet specific needs or improve application responsiveness.



Write Optimizations

Optimization Method: controls how write data is passed to the underlying communications driver. The options are:

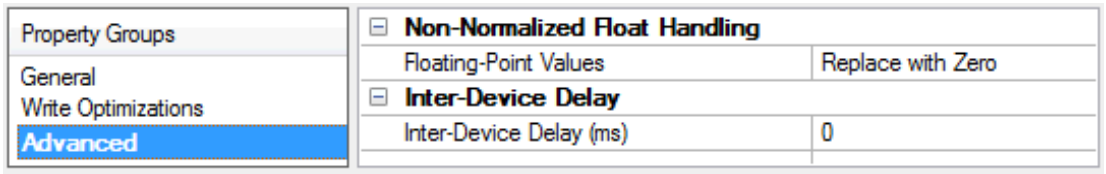
- **Write All Values for All Tags**: This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.
- **Write Only Latest Value for Non-Boolean Tags**: Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.
  Note: This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.
- **Write Only Latest Value for All Tags**: This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write queue before they are sent. This is the default mode.

Duty Cycle: is used to control the ratio of write to read operations. The ratio is always based on one read for every one to ten writes. The duty cycle is set to ten by default, meaning that ten writes occur for each read operation. Although the application is performing a large number of continuous writes, it must be ensured that read data is still given time to process. A setting of one results in one read operation for every write operation. If there are no write operations to perform, reads are processed continuously. This allows optimization for applications with continuous writes versus a more balanced back and forth data flow.

Note: It is recommended that the application be characterized for compatibility with the write optimization enhancements before being used in a production environment.

Channel Properties - Advanced

This group is used to specify advanced channel properties. Not all drivers support all properties; so the Advanced group does not appear for those devices.



Non-Normalized Float Handling: Non-normalized float handling allows users to specify how a driver handles non-normalized IEEE-754 floating point data. A non-normalized value is defined as Infinity, Not-a-Number (NaN), or as a Denormalized Number. The default is Replace with Zero. Drivers that have native float handling may default to Unmodified. Descriptions of the options are as follows:

- **Replace with Zero**: This option allows a driver to replace non-normalized IEEE-754 floating point values with zero before being transferred to clients.
- **Unmodified**: This option allows a driver to transfer IEEE-754 denormalized, normalized, non-number, and infinity values to clients without any conversion or changes.

Note: This property is disabled if the driver does not support floating point values or if it only supports the option that is displayed. According to the channel's float normalization setting, only real-time driver tags (such as values and arrays) are subject to float normalization. For example, EFM data is not affected by this setting.

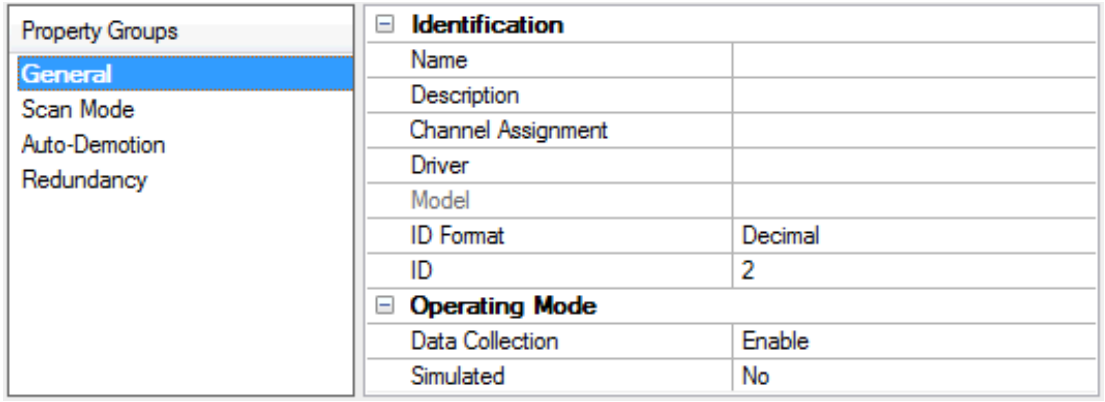
For more information on the floating point values, refer to "How To ... Work with Non-Normalized Floating Point Values" in the server help.

Inter-Device Delay: Specify the amount of time the communications channel waits to send new requests to the next device after data is received from the current device on the same channel. Zero (0) disables the delay.

Note: This property is not available for all drivers, models, and dependent settings.

Device Properties - General

A device represents a single target on a communications channel. If the driver supports multiple controllers, users must enter a device ID for each controller.



Identification

Name: This property specifies the name of the device. It is a logical user-defined name that can be up to 256 characters long, and may be used on multiple channels.

Note: Although descriptive names are generally a good idea, some OPC client applications may have a limited display window when browsing the OPC server's tag space. The device name and channel name become part of the browse tree information as well. Within an OPC client, the combination of channel name and device name would appear as "ChannelName.DeviceName".
For more information, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in server help.

Description: User-defined information about this device.
Many of these properties, including Description, have an associated system tag.

Channel Assignment: User-defined name of the channel to which this device currently belongs.

Driver: Selected protocol driver for this device. This property specifies the driver selected during channel creation. It is disabled in the channel properties.

Model: This property specifies the specific type of device that is associated with this ID. The contents of the drop-down menu depends on the type of communications driver being used. Models that are not supported by a driver are disabled. If the communications driver supports multiple device models, the model selection can only be changed when there are no client applications connected to the device.

Note: If the communication driver supports multiple models, users should try to match the model selection to the physical device. If the device is not represented in the drop-down menu, select a model that conforms closest to the target device. Some drivers support a model selection called "Open," which allows users to communicate without knowing the specific details of the target device. For more information, refer to the driver help documentation.

ID: This property specifies the device's driver-specific station or node. The type of ID entered depends on the communications driver being used. For many communication drivers, the ID is a numeric value. Drivers that support a Numeric ID provide users with the option to enter a numeric value whose format can be changed to suit the needs of the application or the characteristics of the selected communications driver. The ID format can be Decimal, Octal, and Hexadecimal.

Note: If the driver is Ethernet-based or supports an unconventional station or node name, the device's TCP/IP address may be used as the device ID. TCP/IP addresses consist of four values that are separated by periods, with each value in the range of 0 to 255. Some device IDs are string based. There may be additional properties to configure within the ID field, depending on the driver. For more information, refer to the driver's help documentation.

Operating Mode

Data Collection: This property controls the device's active state. Although device communications are enabled by default, this property can be used to disable a physical device. Communications are not attempted when a device is disabled. From a client standpoint, the data is marked as invalid and write operations are not accepted. This property can be changed at any time through this property or the device system tags.

Simulated: This option places the device into Simulation Mode. In this mode, the driver does not attempt to communicate with the physical device, but the server continues to return valid OPC data. Simulated stops physical communications with the device, but allows OPC data to be returned to the OPC client as valid data. While in Simulation Mode, the server treats all device data as reflective: whatever is written to the simulated device is read back and each OPC item is treated individually. The item's memory map is based on the group Update Rate. The data is not saved if the server removes the item (such as when the server is reinitialized). The default is No.

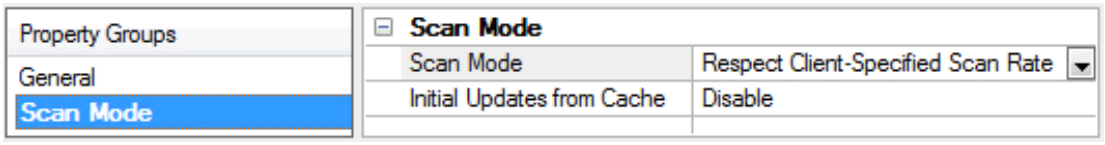
Notes:

1. This System tag (_Simulated) is read only and cannot be written to for runtime protection. The System tag allows this property to be monitored from the client.
2. In Simulation mode, the item's memory map is based on client update rate(s) (Group Update Rate for OPC clients or Scan Rate for native and DDE interfaces). This means that two clients that reference the same item with different update rates return different data.

Simulation Mode is for test and simulation purposes only. It should never be used in a production environment.

Device Properties - Scan Mode

The Scan Mode specifies the subscribed-client requested scan rate for tags that require device communications. Synchronous and asynchronous device reads and writes are processed as soon as possible; unaffected by the Scan Mode properties.



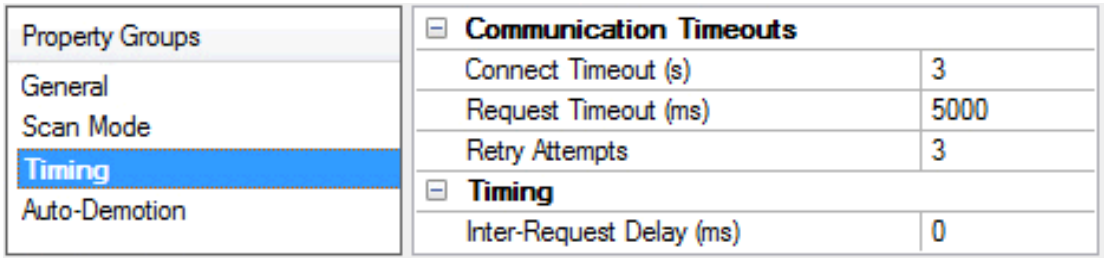
Scan Mode: specifies how tags in the device are scanned for updates sent to subscribed clients. Descriptions of the options are:

- **Respect Client-Specified Scan Rate**: This mode uses the scan rate requested by the client.
- **Request Data No Faster than Scan Rate**: This mode specifies the maximum scan rate to be used. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
  Note: When the server has an active client and items for the device and the scan rate value is increased, the changes take effect immediately. When the scan rate value is decreased, the changes do not take effect until all client applications have been disconnected.
- **Request All Data at Scan Rate**: This mode forces tags to be scanned at the specified rate for subscribed clients. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
- **Do Not Scan, Demand Poll Only**: This mode does not periodically poll tags that belong to the device nor perform a read to get an item's initial value once it becomes active. It is the client's responsibility to poll for updates, either by writing to the _DemandPoll tag or by issuing explicit device reads for individual items. *For more information, refer to "Device Demand Poll" in server help*.
- **Respect Tag-Specified Scan Rate**: This mode forces static tags to be scanned at the rate specified in their static configuration tag properties. Dynamic tags are scanned at the client-specified scan rate.

Initial Updates from Cache: When enabled, this option allows the server to provide the first updates for newly activated tag references from stored (cached) data. Cache updates can only be provided when the new item reference shares the same address, scan rate, data type, client access, and scaling properties. A device read is used for the initial update for the first client reference only. The default is disabled; any time a client activates a tag reference the server attempts to read the initial value from the device.

Device Properties - Timing

The device Timing properties allow the driver's response to error conditions to be tailored to fit the application's needs. In many cases, the environment requires changes to these properties for optimum performance. Factors such as electrically generated noise, modem delays, and poor physical connections can influence how many errors or timeouts a communications driver encounters. Timing properties are specific to each configured device.



Communications Timeouts

Connect Timeout: This property (which is used primarily by Ethernet based drivers) controls the amount of time required to establish a socket connection to a remote device. The device's connection time often takes longer than normal communications requests to that same device. The valid range is 1 to 30 seconds. The default is typically 3 seconds, but can vary depending on the driver's specific nature. If this setting is not supported by the driver, it is disabled.

Note: Due to the nature of UDP connections, the connection timeout setting is not applicable when communicating via UDP.

Request Timeout: This property specifies an interval used by all drivers to determine how long the driver waits for a response from the target device to complete. The valid range is 50 to 9,999,999 milliseconds (167.6667 minutes). The default is usually 1000 milliseconds, but can vary depending on the driver. The default timeout for most serial drivers is based on a baud rate of 9600 baud or better. When using a driver at lower baud rates, increase the timeout to compensate for the increased time required to acquire data.

Retry Attempts: This property specifies how many times the driver retries a communications request before considering the request to have failed and the device to be in error. The valid range is 1 to 10. The default is typically 3, but can vary depending on the driver's specific nature. The number of retries configured for an application depends largely on the communications environment.

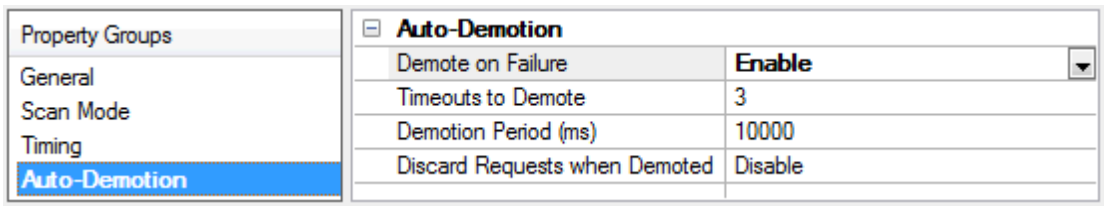
Timing

Inter-Request Delay: This property specifies how long the driver waits before sending the next request to the target device. It overrides the normal polling frequency of tags associated with the device, as well as one-time reads and writes. This delay can be useful when dealing with devices with slow turnaround times and in cases where network load is a concern. Configuring a delay for a device affects communications with all other devices on the channel. It is recommended that users separate any device that requires an inter-request delay to a separate channel if possible. Other communications properties (such as communication serialization) can extend this delay. The valid range is 0 to 300,000 milliseconds; however, some drivers may limit the maximum value due to a function of their particular design. The default is 0, which indicates no delay between requests with the target device.

Note: Not all drivers support Inter-Request Delay. This setting does not appear if it is not available.

Device Properties - Auto-Demotion

The Auto-Demotion properties can temporarily place a device off-scan in the event that a device is not responding. By placing a non-responsive device offline for a specific time period, the driver can continue to optimize its communications with other devices on the same channel. After the time period has been reached, the driver re-attempts to communicate with the non-responsive device. If the device is responsive, the device is placed on-scan; otherwise, it restarts its off-scan time period.



Demote on Failure: When enabled, the device is automatically taken off-scan until it is responding again.

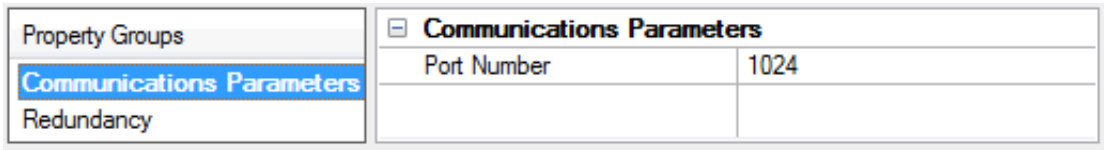
Tip: Determine when a device is off-scan by monitoring its demoted state using the _AutoDemoted system tag.

Timeouts to Demote: Specify how many successive cycles of request timeouts and retries occur before the device is placed off-scan. The valid range is 1 to 30 successive failures. The default is 3.

Demotion Period: Indicate how long the device should be placed off-scan when the timeouts value is reached. During this period, no read requests are sent to the device and all data associated with the read requests are set to bad quality. When this period expires, the driver places the device on-scan and allows for another attempt at communications. The valid range is 100 to 3600000 milliseconds. The default is 10000 milliseconds.

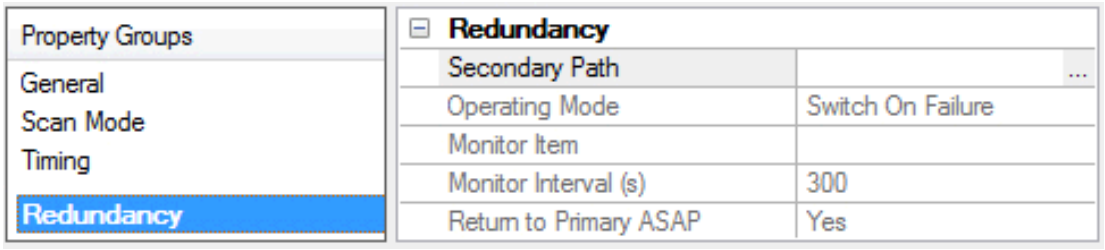
Discard Requests when Demoted: Select whether or not write requests should be attempted during the off-scan period. Disable to always send write requests regardless of the demotion period. Enable to discard writes; the server automatically fails any write request received from a client and does not post a message to the Event Log.

Device Properties - Communications Parameters



Port: Configure the Ethernet port to be used when connecting to the device.

Device Properties - Redundancy



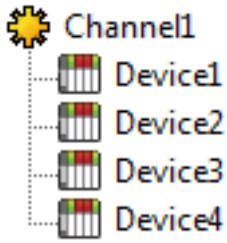
Redundancy is available with the Media-Level Redundancy Plug-In.

Consult the website, a sales representative, or the user manual for more information.

Optimizing Communications

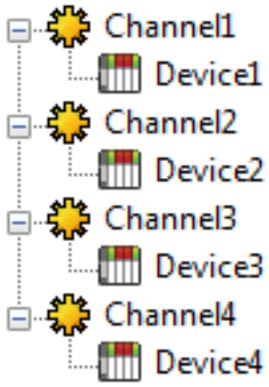
The Toshiba Ethernet Driver has been designed to provide the best performance with the least amount of impact on the system's overall performance. While the Toshiba Ethernet Driver is fast, there are a couple of guidelines that can be used in order to control and optimize the application and gain maximum performance. This server refers to communications protocols like Toshiba 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 Toshiba Ethernet 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 Toshiba Ethernet Driver or the network. An example of how the application may appear when configured using a single channel is shown below.



Each device appears under a single Toshiba 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 Toshiba Ethernet Driver could only define one single channel, then the example shown above would be the only option available; however, the Toshiba Ethernet Driver can define up to 100 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 100 or fewer channels, it can be optimized exactly how it is shown here.

The performance will improve even if the application has more than 100 channels. While 100 or fewer channels 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.

Data Types Description

Data Type	Description
Boolean	Single bit
Word	Unsigned 16-bit value bit 0 is the least significant bit bit 15 the most significant bit
Short	Signed 16-bit value bit 0 is the least significant bit bit 14 the most significant bit bit 15 is the sign bit
DWord	Unsigned 32-bit value bit 0 is the least significant bit bit 31 the most significant bit
Long	Signed 32-bit value bit 0 is the least significant bit bit 30 the most significant bit bit 31 is the sign bit
Float	32-bit floating-point value The driver interprets two consecutive registers as a floating point value by making the second register the high word and the first register the low word.
String	Null-terminated ASCII string Supported on all models, includes HiLo LoHi byte order selection.

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.

T2 Addressing
T3 Addressing
S2 Addressing
S3 Addressing

T2 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate.

Address Type	Range	Data Type	Access
Input Devices	X00...X0F-X127...X127F	Boolean	Read/Write
Direct Input Devices	I00...I0F-I127...I127F	Boolean	Read Only
Output Devices	Y00...Y0F-Y127...Y127F	Boolean	Read/Write
Direct Output Devices	O00...O0F-O127...O127F	Boolean	Read/Write
Input Registers	XW0-XW127 XW0-WW126	**Word**, Short DWord, Long, Float	Read/Write
Direct Input Registers	IW0-IW127 IW0-IW126	**Word**, Short DWord, Long, Float	Read/write
Output Registers	YW0-YW127 YW0-YW126	**Word**, Short DWord, Long, Float	Read/Write
Direct Output Registers	OW0-OW127 OW0-OW126	**Word**, Short DWord, Long, Float	Read/Write
Link Relays	L00...L0F-L2550...L255F	Boolean	Read/Write
Link Registers	LW0-LW255 LW0-LW254	**Word**, Short DWord, Long, Float	Read/Write
Link Registers	W0-W1023 W0-W1022	**Word**, Short DWord, Long, Float	Read/Write
Link Register relays	Z00...Z0F-Z5110...Z511F	Boolean	Read/Write
File Registers	F0-F1023 F0-F1022	**Word**, Short DWord, Long, Float	Read/Write
Auxiliary Devices	R00...R0F-R2550...R255F	Boolean	Read/Write
Auxiliary Registers	RW0-RW255 RW0-RW254	**Word**, Short DWord, Long, Float	Read/Write

Address Type	Range	Data Type	Access
Special Devices	S00...S0F-S2550...S255F	Boolean	Read/Write
Special Registers	SW0-SW255 SW0-SW254	**Word**, Short DWord, Long, Float	Read Only
Counter Registers	C0-C255	**Word**, Short	Read/Write
Counter Devices	C.0-C.255	Boolean	Read Only
Timer Registers	T0-T255	**Word**, Short	Read/Write
Timer Devices	T.0-T.255	Boolean	Read Only
Data Memory	D0-D8191 D0-D8190	**Word**, Short DWord, Long, Float	Read/Write
Data Memory As String with HiLo Byte Order	D0.2H-D4095.64H .Bit is string length, range 2 to 64 bytes.	**String**	Read/Write
Data Memory As String with LoHi Byte Order	D0.2L-D4095.64L .Bit is string length, range 2 to 64 bytes.	**String**	Read/Write

String Support

The Toshiba Ethernet Driver supports reading and writing Data registers as an ASCII string. When using Data registers for string data, each register will contain two bytes of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 64 bytes and is entered in place of a bit number. The length must be entered as an even number. The byte order is specified by appending either an "H" or "L" to the address.

Examples

1. To address a string starting at D200 with a length of 50 bytes and HiLo byte order, enter: D200.50H
2. To address a string starting at D500 with a length of 38 bytes and LoHi byte order, enter: D500.38L

T3 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate.

Address Type	Range	Data Type	Access
Input Devices	X00...X0F-X5110...X511F	Boolean	Read/Write
Direct Input Devices	I00...I0F-I5110...I511F	Boolean	Read Only
Output Devices	Y00...Y0F-Y5110...Y511F	Boolean	Read/Write
Direct Output Devices	O00...O0F- O5110...O511F	Boolean	Read/Write
Input Registers	XW0-XW511 XW0-XW510	**Word**, Short DWord, Long, Float	Read/Write
Direct Input Registers	IW0-IW511 IW0-IW510	**Word**, Short DWord, Long,	Read/Write

Address Type	Range	Data Type	Access
		Float	
Output Registers	YW0-YW511 YW0-YW510	**Word**, Short DWord, Long, Float	Read/Write
Direct Output Registers	OW0-OW511 OW0-OW510	**Word**, Short DWord, Long, Float	Read/Write
Link Relays	L00...L0F-L2550...L255F	Boolean	Read/Write
Link Registers	LW0-LW255 LW0-LW254	**Word**, Short DWord, Long, Float	Read/Write
Link Registers	W0-W2047 W0-W2026	**Word**, Short DWord, Long, Float	Read/Write
Link Register relays	Z00...Z0F-Z9990...Z999F	Boolean	Read/Write
File Registers	F0-F8191 F0-F8190	**Word**, Short DWord, Long, Float	Read/Write
Auxiliary Devices	R00...R0F-R9990...R999F	Boolean	Read/Write
Auxiliary Registers	RW0-RW999 RW0-RW998	**Word**, Short DWord, Long, Float	Read/Write
Special Devices	S00...S0F-S2550...S255F	Boolean	Read/Write
Special Registers	SW0-SW255 SW0-SW254	**Word**, Short DWord, Long, Float	Read Only
Counter Registers	C0-C511	**Word**, Short	Read/Write
Counter Devices	C.0-C.511	Boolean	Read Only
Timer Registers	T0-T999	**Word**, Short	Read/Write
Timer Devices	T.0-T.999	Boolean	Read Only
Data Memory	D0-D8191 D0-D8190	**Word**, Short DWord, Long, Float	Read/Write
Data Memory As String with HiLo Byte Order	D0.2H-D8191.64H .Bit is string length, range 2 to 64 bytes.	**String**	Read/Write
Data Memory As String with LoHi Byte Order	D0.2L-D8191.64L .Bit is string length, range 2 to 64 bytes.	**String**	Read/Write

String Support

The Toshiba Ethernet Driver supports reading and writing Data registers as an ASCII string. When using Data registers for string data, each register will contain two bytes of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to

64 bytes and is entered in place of a bit number. The length must be entered as an even number. The byte order is specified by appending either a "H" or "L" to the address.

Examples

1. To address a string starting at D200 with a length of 50 bytes and HiLo byte order, enter: D200.50H
2. To address a string starting at D500 with a length of 38 bytes and LoHi byte order, enter: D500.38L

S2 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate.

Address Type	Range	Data Type	Access
Input Devices	X0 0...X0 F-X3071 0...X3071 F	Boolean*	Read/Write
Direct Input Devices	I0 0...I0 F-I3071 0...I3071 F	Boolean*	Read Only
Output Devices	Y0 0...Y0 F-Y3071 0...Y3071 F	Boolean*	Read/Write
Direct Output Devices	O0 0...O0 F-O3071 0...O3071 F	Boolean*	Read/Write
Input Registers	XW0-XW3071 XW0-XW3070	**Word**, Short DWord, Long, Float	Read/Write
Direct Input Registers	IW0-IW3071 IW0-IW3070	**Word**, Short DWord, Long, Float	Read/Write
Output Registers	YW0-YW3071 YW0-YW3070	**Word**, Short DWord, Long, Float	Read/Write
Direct Output Registers	OW0-OW3071 OW0-OW3070	**Word**, Short DWord, Long, Float	Read/Write
Auxiliary Devices	R0 0...R0 F-R4095 0...R4095 F	Boolean*	Read/Write
Auxiliary Devices	D[0].B[0]-D[4095].B[F]	Boolean*	Read/Write
Auxiliary Registers	RW0-RW4095 RW0-RW4094	**Word**, Short DWord, Long, Float	Read/Write
Auxiliary Registers	DW[0]-DW[4095] DW[0]-DW[4094]	**Word**, Short DWord, Long, Float	Read/Write
Special Devices	S0 0...S0 F-S511 0...S511 F	Boolean*	Read/Write
Special Registers	SW0-SW511 SW0-SW510	**Word**, Short DWord, Long, Float	Read Only
Data Memory	D0-D4095 D0-D4094	**Word**, Short DWord, Long,	Read/Write

Address Type	Range	Data Type	Access
		Float	
Data Memory As String with HiLo Byte Order	D0.64H-D4095.2H .Bit is string length, range 2 to 64 bytes.	**String**	Read/Write
Data Memory As String with LoHi Byte Order	D0.64L-D4095.2L .Bit is string length, range 2 to 64 bytes.	**String**	Read/Write

*The last character in the address is the bit identifier and is represented in hexadecimal format.

String Support

The Toshiba Ethernet Driver supports reading and writing Data registers as an ASCII string. When using Data registers for string data, each register will contain two bytes of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 64 bytes and is entered in place of a bit number. The length must be entered as an even number. The byte order is specified by appending either an "H" or "L" to the address.

Examples

1. To address a string starting at D200 with a length of 50 bytes and HiLo byte order, enter: D200.50H
2. To address a string starting at D500 with a length of 38 bytes and LoHi byte order, enter: D500.38L

S3 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate.

Address Type	Range	Data Type	Access
Input Devices	X0 0...X0 F-X5119 0...X5119 F	Boolean*	Read/Write
Direct Input Devices	I0 0...I0 F-I5119 0...I5119 F	Boolean*	Read Only
Output Devices	Y0 0...Y0 F-Y5119 0...Y5119 F	Boolean*	Read/Write
Direct Output Devices	O0 0...O0 F-O5119 0...O5119 F	Boolean*	Read/Write
Input Registers	XW0-XW5119 XW0-XW5118	**Word**, Short DWord, Long, Float	Read/Write
Direct Input Registers	IW0-IW5119 IW0-IW5118	**Word**, Short DWord, Long, Float	Read/Write
Output Registers	YW0-YW5119 YW0-YW5118	**Word**, Short DWord, Long, Float	Read/Write
Direct Output Registers	OW0-OW5119 OW0-OW5118	**Word**, Short DWord, Long, Float	Read/Write

Address Type	Range	Data Type	Access
Auxiliary Devices	R0 0...R0 F-R4095 0...R4095 F	Boolean*	Read/Write
Auxiliary Devices	D[0].B[0]-D[4095].B[F]	Boolean*	Read/Write
Auxiliary Registers	RW0-RW4095 RW0-RW4094	**Word**, Short DWord, Long, Float	Read/Write
Auxiliary Registers	DW[0]-DW[4095] DW[0]-DW[4094]	**Word**, Short DWord, Long, Float	Read/Write
Special Devices	S0 0...S0 F-S511 0...S511 F	Boolean*	Read/Write
Special Registers	SW0-SW511 SW0-SW510	**Word**, Short DWord, Long, Float	Read Only
Data Memory	D0-D4095 D0-D4094	**Word**, Short DWord, Long, Float	Read/Write
Data Memory As String with HiLo Byte Order	D0.64H-D4095.2H .Bit is string length, range 2 to 64 bytes.	**String**	Read/Write
Data Memory As String with LoHi Byte Order	D0.64L-D4095.2L .Bit is string length, range 2 to 64 bytes.	**String**	Read/Write

*The last character in the address is the bit identifier and is represented in hexadecimal format.

String Support

The Toshiba Ethernet Driver supports reading and writing Data registers as an ASCII string. When using Data registers for string data, each register will contain two bytes of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 64 bytes and is entered in place of a bit number. The length must be entered as an even number. The byte order is specified by appending either an "H" or "L" to the address.

Examples

1. To address a string starting at D200 with a length of 50 bytes and HiLo byte order, enter: D200.50H
2. To address a string starting at D500 with a length of 38 bytes and LoHi byte order, enter: D500.38L

Error Descriptions

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

Address Validation

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

Device Status Messages

Winsock initialization failed (OS Error = n)
Winsock V1.1 or higher must be installed to use the Toshiba Ethernet device driver
Device <device name> is not responding
Unable to write to <address> on device <device name>
Device <device name> responded with error code 'n' (Tag <tag name>)
Deactivating tag block. Bad address in block [<start address> to <end address>] on device <device name>

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.

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

Error Type:

Warning

Possible Cause:

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

Solution:

Modify the requested data type in the client application.

Device address <address> contains a syntax error

Error Type:

Warning

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

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

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

Error Type:
Warning

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

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

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.

Deactivating tag block. Bad address in block [<start address> to <end address>] on device <device name>

Error Type:
Warning

Possible Cause:

The driver is requesting a block of data with one or more invalid addresses.

Solution:
Verify that all addresses in specified range are valid for the device.

Device <device name> is not responding

Error Type:
Serious

Possible Cause:

1. The named device may not be connected to the PLC network.
2. The named device may have been assigned an incorrect Network ID.
3. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

Solution:

1. Check the PLC network connections.
2. Verify that the Network ID given to the named device matches that of the actual device.
3. Increase the Request Timeout setting so that the entire response can be handled.

Device <device name> responded with error code 'n' (Tag <tag name>)

Error Type:
Warning

Possible Cause:
The device responded with specified error code.

Solution:
Consult the Toshiba documentation for explanation of code and suggested solution.

Unable to write to address <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 communications properties 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 PLC device.
2. Verify that the specified communications properties match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.

Winsock initialization failed (OS Error = n)

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 Toshiba Ethernet Driver

Error Type:
Fatal

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

Solution:
Upgrade Winsock to version 1.1 or higher.

Index

A

B

C

D

E

H

I

L

M

N

O

P

R

S

T

U

W

Appendix: Hardware Configuration

The hardware must be configured before Ethernet communications is possible. For more information about a specific hardware series, refer to the links below.

S Series Setup
T2 and T3 Setup

S Series Setup

To make a connection to the EN731 BNC, users must have two 50 ohm connectors: one on each end of the cable T connectors. Both ends require a T connector with the 50 ohm terminator.

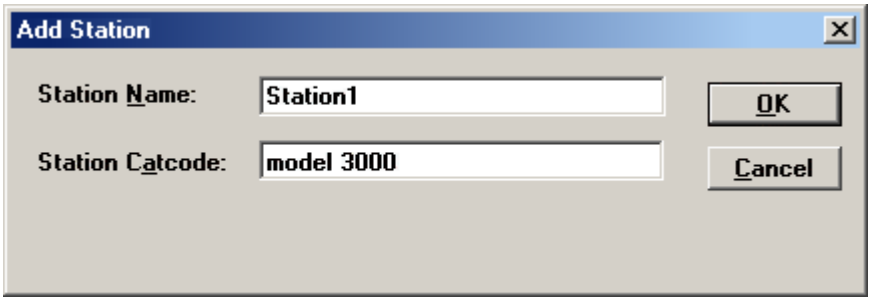
Connecting to a V Series Using the Engineering Tool 2

A ladder is not required for the V series devices to communicate. For more information, refer to the instructions below.

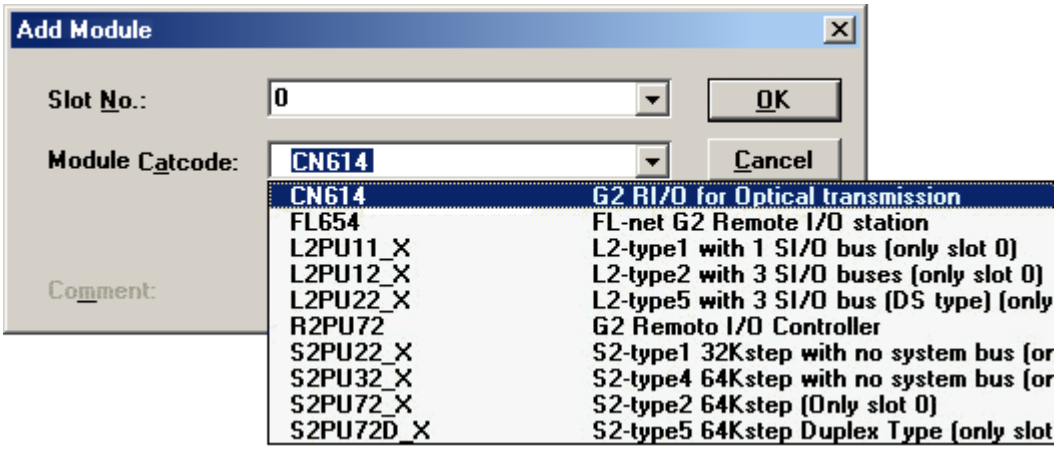
1. To start, right-click on the **System** folder and select **New**. Name the new project and then double-click the newly created project icon. This will create a tree with the topics **Libraries, Networks, and Stations** in the lower left quadrant of the display.

2. Right-click on **Stations** and then select **New**. Select the model as desired.

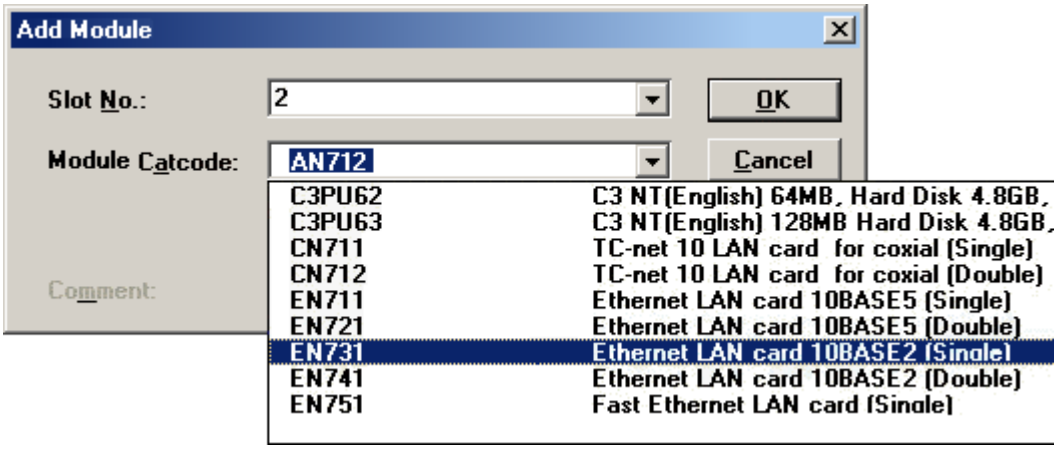
   Note: For the S3 CPUs, choose model 3000.



3. Right-click on **Units** and then select **New**. In this dialog, choose the main board, backplane and model number (such as BU748). This will create a **Modules** subfolder.
4. Next, right-click on **Modules** and select **New**.
5. Select the slot number and CPU model and then click **OK**.

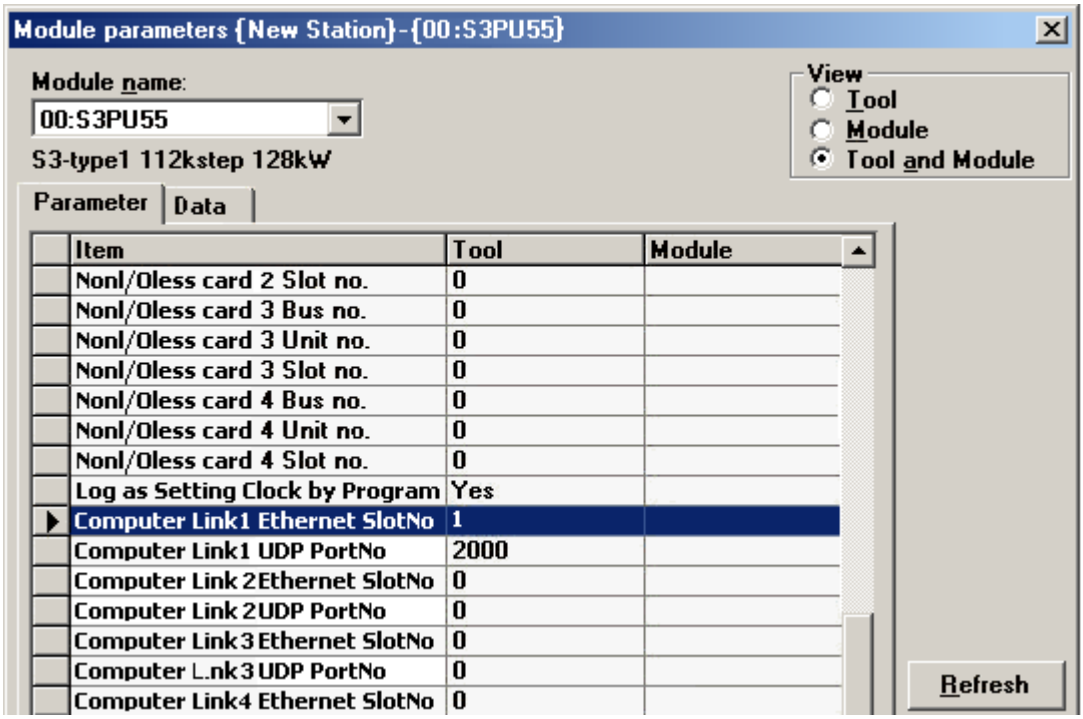


6. Select the Ethernet model and then click **OK**.



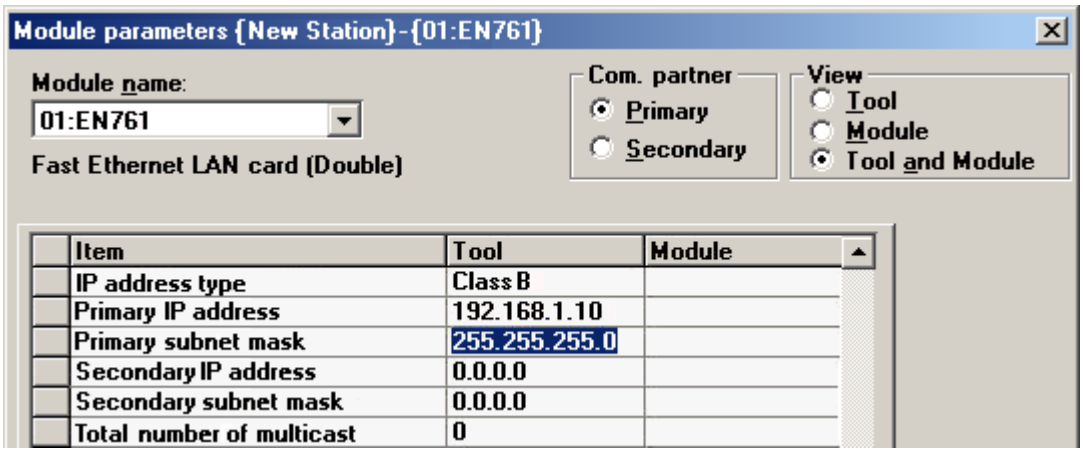
7. Next, right-click on the CPU module and then click **Module Parameters** | **Tool and Module**. Scroll through the parameters until "Computer Link1 Ethernet Slot" and "...UDP PortNo" are located.
8. Enter the Ethernet module's slot number and the desired port number. Then, enter the slot number and port number in both the Tool column and the Module column.

   Note: Not all screens will appear as the image shown below.
9. Ensure the device is set to **Halt Mode** by turning the key, and then click **Write** from the module and parameters window. A write successful message will appear upon completion. Close the window.



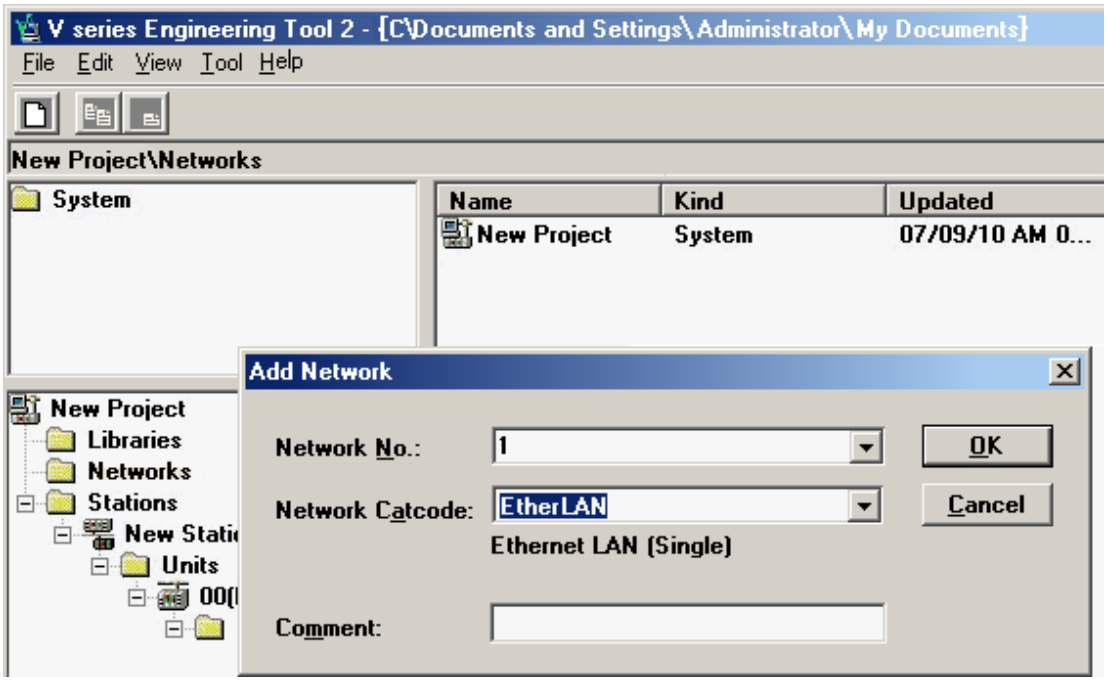
10. Right-click on the Ethernet module and then select **Module Parameters.**

11. Next, click **Tool and Module.** In the Tool column, enter the desired IP and subnet mask. Leave the **IP Address Type** set to Class B.

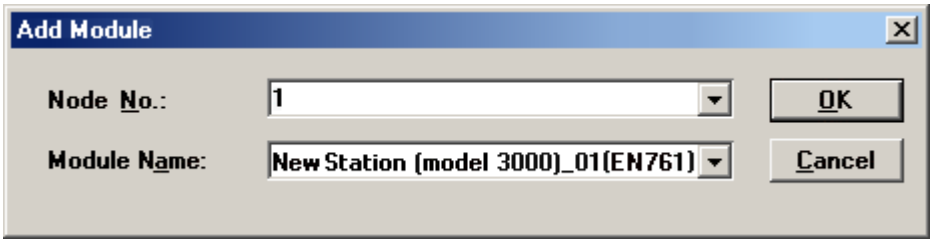


12. Click **Download** in order to write the information to the module. Then, close the window.

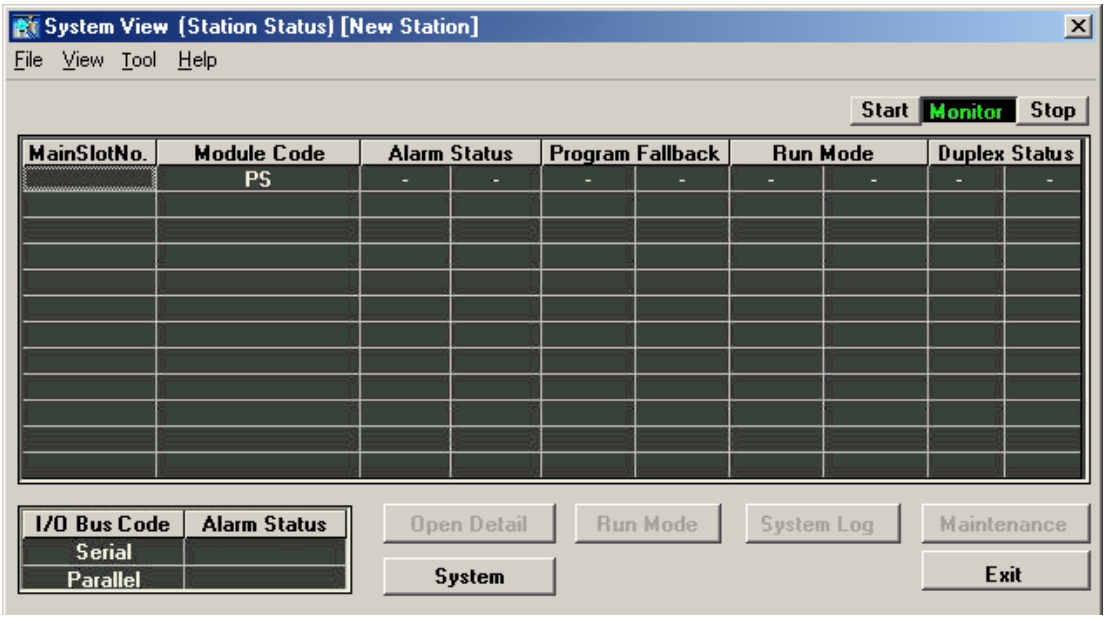
13. Click the **Network** folder from the tree and then select **New**. For a 3000 series project using an EN model Ethernet, select "EtherLAN" for the network. This will create a Module subfolder.



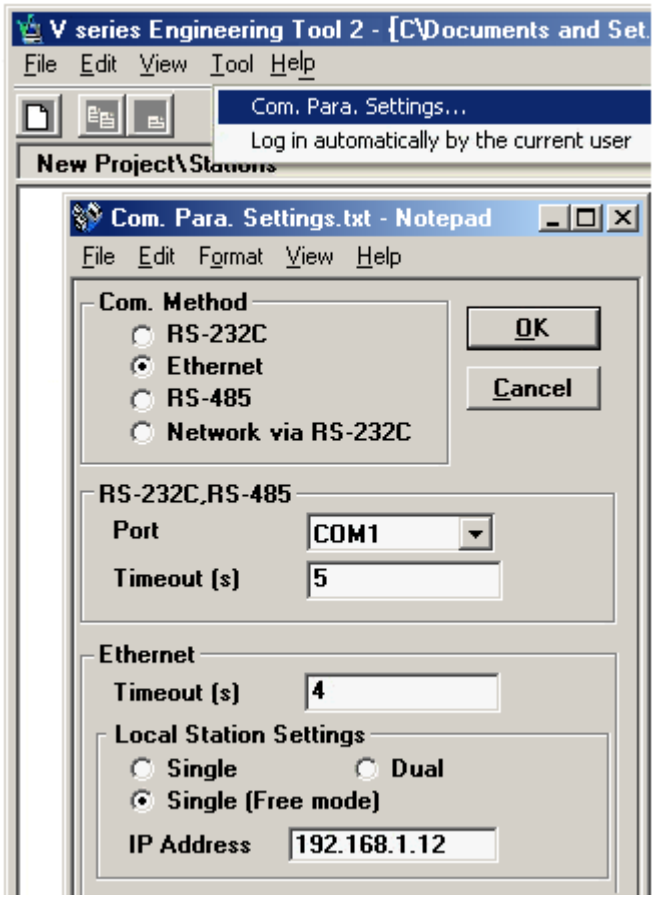
14. Right-click on **Module** and select **New**. Select the Ethernet module and then close the window.



15. To verify that connections have been set up successfully, right-click the first subgroup under the **Stations** folder. For example, test (model 3000) and select "System View" from the list. This will show the status of the devices.



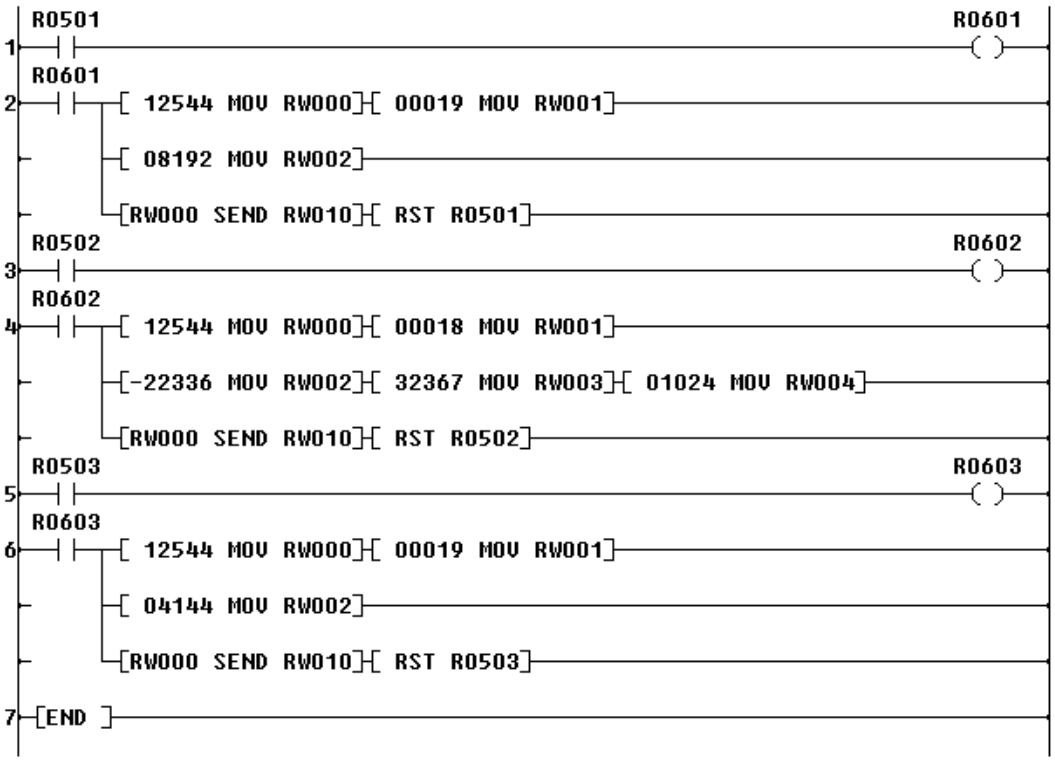
16. From the **Tool** drop-down list, select **Com. Para. Settings**. In the IP address box at the bottom of the window, enter the PC's IP. Then, choose Ethernet as the form of communications and close the window.



17. To test the connection, return to the System View and verify that the devices' statuses are visible without any communication errors.

T2 and T3 Setup

Before this driver can be used, the PLC must be configured with an IP and port number. This can be done using a simple ladder program, which must be run each time the PLC is powered up. An example program that sets IP = 192.168.111.126 and port = 1024 is shown below. Users may wish to modify this program to run automatically whenever the PLC powers up.



The first block of code at rung 2 places the PLC in Standby Mode when bit R0501 is set. This must be done before the IP and port can be set. The second block of code at rung 4 sets the IP and port number when bit R0502 is set. The final block of code at rung 6 puts the PLC back in Run Mode when bit R0603 is set.

Each block of code places parameters in memory starting at RW000, issues a SEND command using those parameters, and then resets the corresponding trigger bit. The parameters in each block are defined as shown in the tables below.

Set Standby Mode

Parameter	Value	Description
RW000	0x3100 (12544)	Specifies module.
RW001	0x0013 (19)	Change mode command code.
RW002	0x2000 (8192)	Specifies standby mode.

Set IP and Port

Parameter	Value	Description
RW000	0x3100 (12544)	Specifies module.
RW001	0x0012 (18)	Set IP and port command code.
RW002	User Defined	Specifies upper 2 octals of IP address (network byte order).
RW003	User Defined	Specifies lower 2 octals of IP address (network byte order).
RW004	User Defined	Specifies port number.

The IP address parameters must be in the correct format. In the example IP, the four octals that follow are expressed in hex: 0xC0 (192), 0xA8 (168), 0x6F (111) and 0x7E (126). Therefore, the value placed in RW002 is 0xA8C0 (or -22336 as a signed integer). The value placed in RW003 is 0x7E6F (or 32367 as a signed integer).

Set Run Mode

Parameter	Value	Description
RW000	0x3100 (12544)	Specifies module.
RW001	0x0013 (19)	Change mode command code.
RW002	0x1030 (4144)	Specifies Run Mode.