# AutomationDirect Productivity 3000 Ethernet Driver Help

# Table of Contents

# AutomationDirect Productivity 3000 Ethernet Driver Help

Help version 1.032

**CONTENTS**

## Overview

The AutomationDirect Productivity 3000 Ethernet Driver provides an easy and reliable way to connect AutomationDirect Productivity 3000 Ethernet devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications.

## Device Setup

### Supported Devices

P3-550

**Important:** This device cannot be used as a MODBUS TCP server if DHCP IP addressing is being used.

### Firmware Versions

P3-550: ver.1.0.7.2

### Hardware Setup

It is strongly recommended that the device be configured with "No exception response for non-existing Modbus address requests" checked. This is the default setting. For more information, refer to the device's programming software under "Project Properties: Modbus Server Settings."

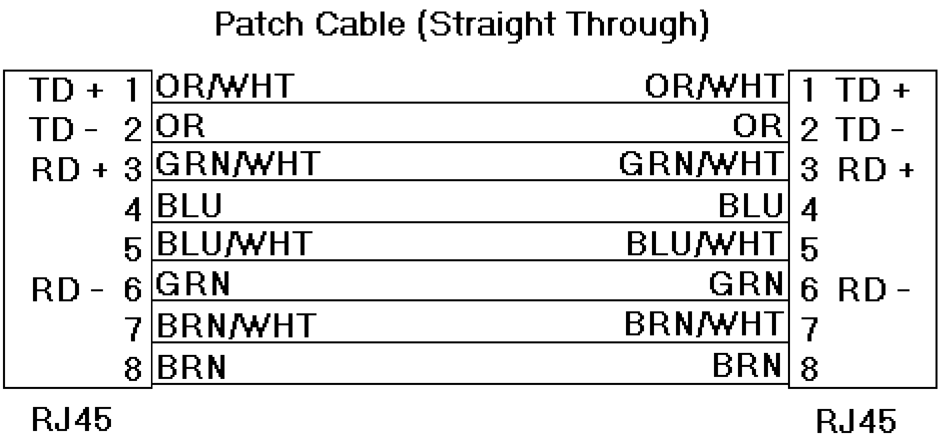
### Maximum Number of Channels and Devices

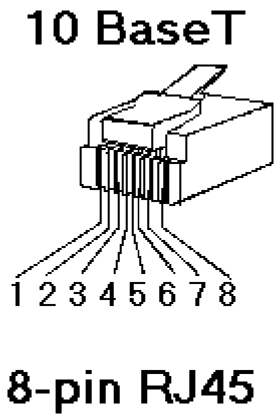
Maximum number of channels and devices supported by this driver are as follows:
100 channels
256 devices

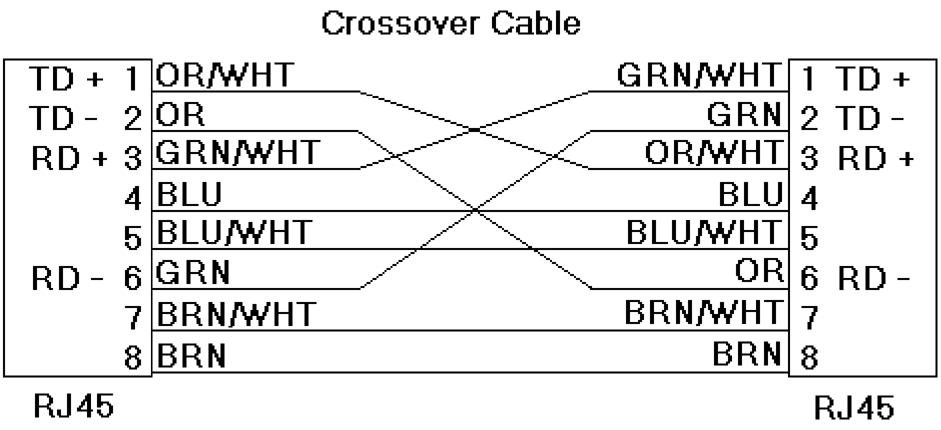
### Device ID

This parameter specifies the IP address of the device with the extension **:#** where **#** is the Slave ID.

### Cable Diagrams



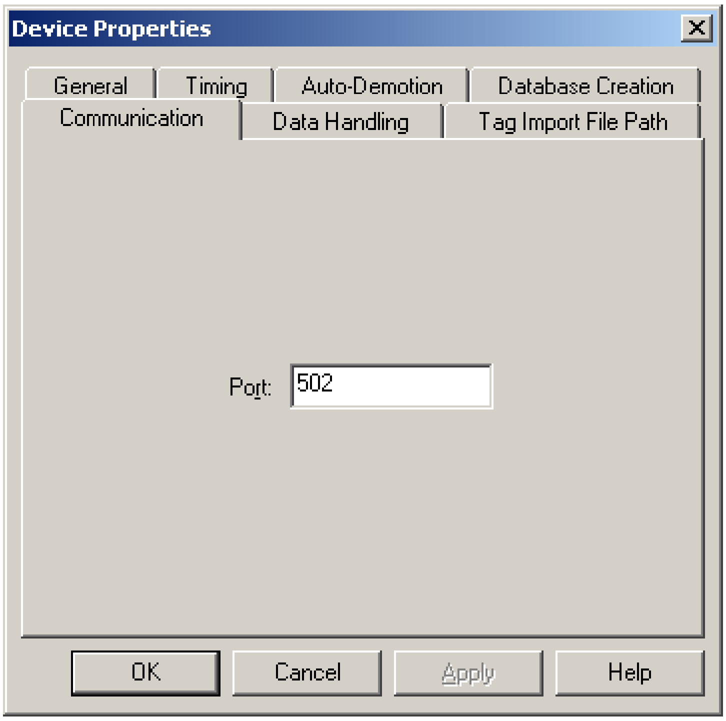




## Communication

### Port

This parameter is used to specify the port number. The default setting is 502.



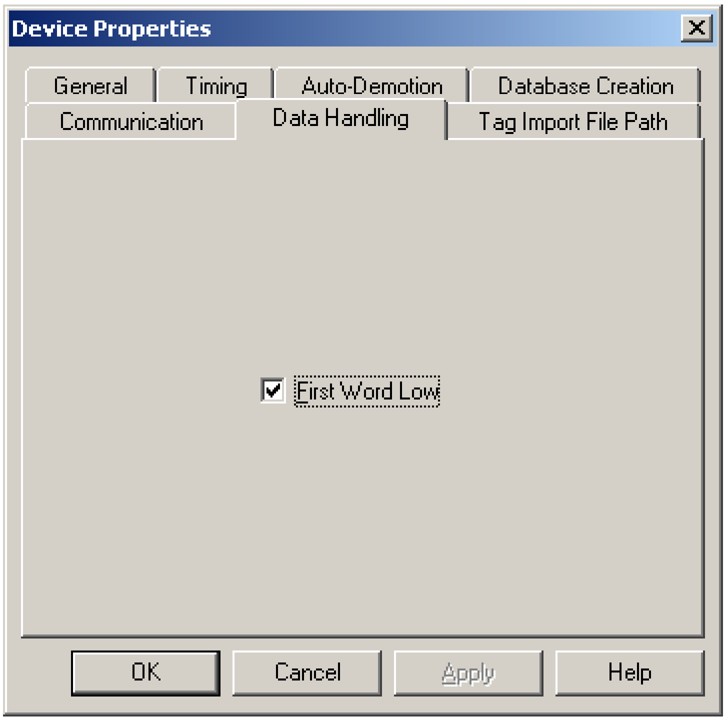
Port: 502

**Note:** The default communication protocol is Modbus TCP/IP.

## Data Handling

### First Word Low

When First Word Low is checked, the driver will assume that the first register used to construct a 32 bit value is the low word. When this setting is unchecked, the driver will assume that the first register is the high word.

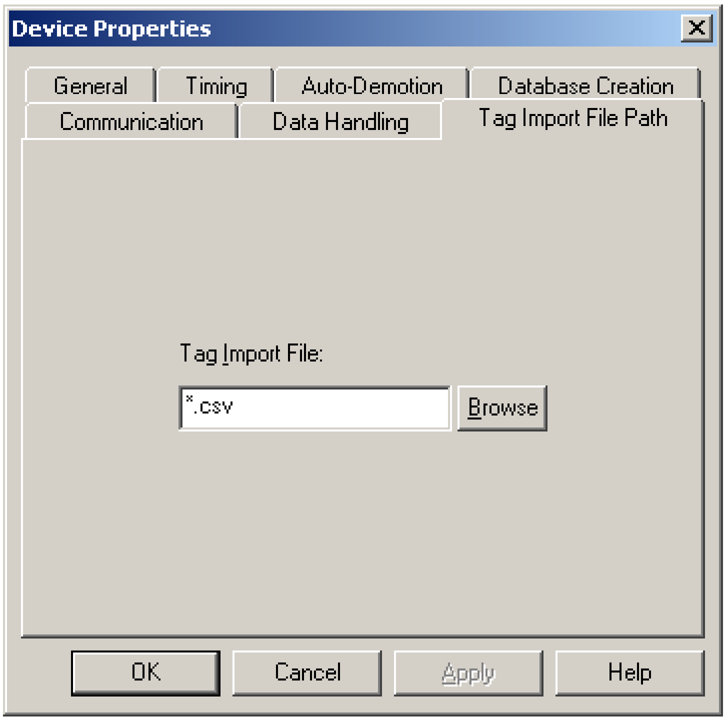


☑ First Word Low

## Tag Import File Path

### Tag Import File

Enter the exact location of the tag import file (*filename*.csv). The CSV file needs to be created in the Productivity Suite Programming Software. The tag import file will be used by the Automatic Tag Database Generation feature to create the tag database. All tags will be imported and expanded according to their respective data types.



Tag Import File: *.csv

**See Also:** **Automatic Tag Database Generation**

## Automatic Tag Database Generation

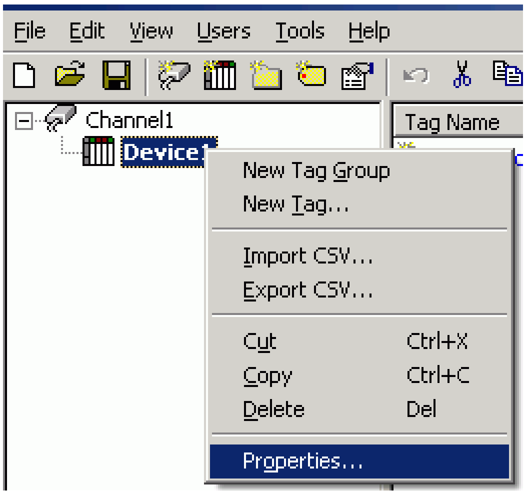
The AutomationDirect Productivity 3000 Ethernet driver makes use of the OPC server's Automatic Tag Database Generation feature. This enables the driver to automatically create tags that access data points used in a device configuration.

The OPC server will use the tag import file to create the tag database. For more information, refer to **Tag Import File Path**.

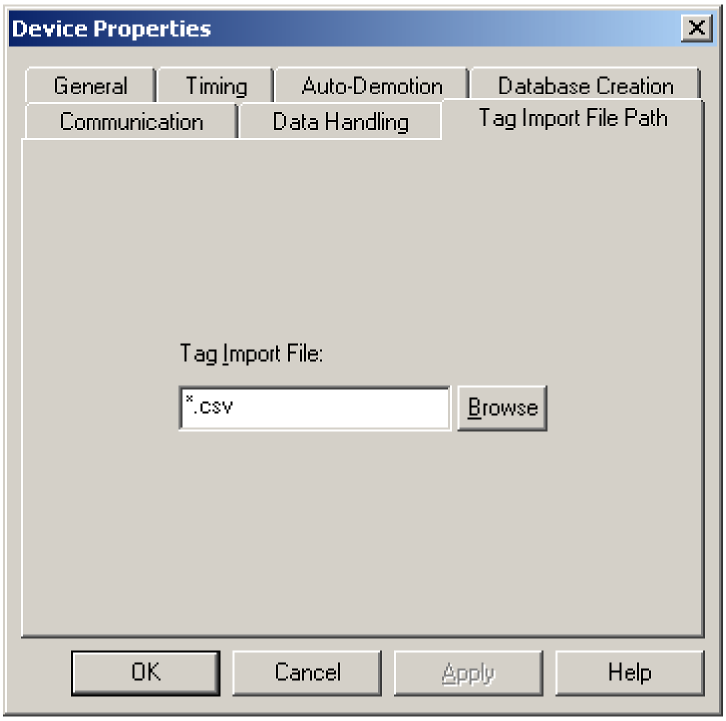
**Note:** The tag import file (*.csv) needs to be created in the Productivity Suite Programming Software.

To run the tag database generation for a device, perform the following steps.

1. In the OPC server project, right-click on the device. Select **Properties**.



2. Click on the **Tag Settings** tab.



Tag Import File: *.csv Browse

3. Use **Browse** to locate the folder containing the import file (*.csv). Then, select the import file.
4. Click **OKto start the import and tag generation process.**

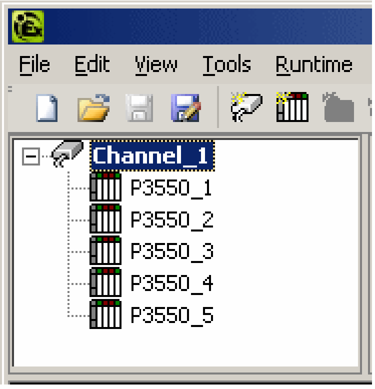
5. The OPC Server's event log will show when the tag generation process started, in addition to any errors that occurred while processing and when the process completed.

**Note:** For more information, refer to the section on Automatic OPC Tag Database Generation in the OPC Server's help file.

## Optimizing Your AutomationDirect Productivity 3000 Ethernet Communications

The AutomationDirect Productivity 3000 Ethernet driver has been designed to provide the best performance with the least amount of impact on the system's overall performance. While the AutomationDirect Productivity 3000 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.

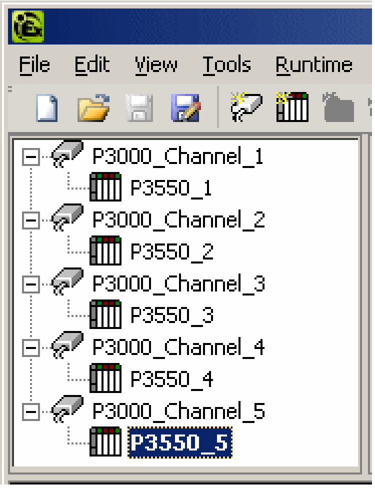
Our server refers to communications protocols like AutomationDirect Productivity 3000 Ethernet driver as a channel. Each channel defined in the application represents a separate path of execution in the server. Once a channel has been defined, a series of devices must then be defined under that channel. Each of these devices represents a single 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 AutomationDirect Productivity 3000 Ethernet driver or the network. An example of how the application may appear when configured using a single channel is shown below.



Channel_1
- P3550_1
- P3550_2
- P3550_3
- P3550_4
- P3550_5

Each device appears under a single AutomationDirect Productivity 3000 Ethernet driver 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 AutomationDirect Productivity 3000 Ethernet driver could only define one single channel, then the example shown above would be the only option available; however, the AutomationDirect Productivity 3000 Ethernet driver can define up to 256 devices per 100 channel. 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.



- P3000_Channel_1
  - P3550_1
- P3000_Channel_2
  - P3550_2
- P3000_Channel_3
  - P3550_3
- P3000_Channel_4
  - P3550_4
- P3000_Channel_5
  - P3550_5

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 256 or fewer devices, it can be optimized exactly how it is shown here.

The performance will improve even if the application has more than 256 devices. While 256 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.

**See Also:** **Data Handling**

## Data Types Description

| Data Type | Description |
|---|---|
| Boolean | Single bit. |
| Byte | Unsigned 8 bit value.<br>bit 0 is the low bit.<br>bit 7 is the high bit. |
| Word | Unsigned 16 bit value.<br>bit 0 is the low bit.<br>bit 15 is the high bit. |
| Short | Signed 16 bit value.<br>bit 0 is the low bit.<br>bit 14 is the high bit.<br>bit 15 is the sign bit. |
| DWord | Unsigned 32 bit value.<br>bit 0 is the low bit.<br>bit 31 is the high 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. |
| BCD | Two byte packed BCD.<br>Value range is 0-9999. Behavior is undefined for values beyond this range. |
| LBCD | Four byte packed BCD.<br>Value range is 0-99999999. Behavior is undefined for values beyond this range. |
| Float | 32 bit floating point value.<br>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.<br>Support includes HiLo LoHi byte order selection. |

## Address Descriptions

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

### System Index Addresses

| Device Type | Range | Data Type | Access |
|---|---|---|---|
| Analog Input Float 32 | AIF32-00.00.00.01-AIF32-99.05.11.32 | **Float** | Read Only |
| Analog Input Signed 32 | AIS32-00.00.00.01-AIS32-99.05.11.32 | **Long** | Read Only |
| Analog Output Float 32 | AOF32-00.00.00.01-AOF32-99.05.11.32 | **Float** | Read/Write |
| Analog Output Signed 32 | AOS32-00.00.00.01-AOS32-99.05.11.32 | **Long** | Read/Write |
| Discrete Inputs | DI-00.00.00.01-DI-99.05.11.128 | **Boolean** | Read Only |
| Discrete Outputs | DO-00.00.00.01-DO-99.05.11.128 | **Boolean** | Read/Write |
| I/O Module Status Bits | MST-00.00.00.01-MST-99.05.11.128 | **Boolean** | Read Only |

### Continuous Addresses

| Device Type | Range | Data Type | Access |
|---|---|---|---|
| Internal BCD 16 | BCD16-000001-BCD16-999999 | **BCD** | Read/Write |
| Internal BCD 32 | BCD32-000001-BCD32-999999 | **LBCD** | Read/Write |
| Internal Bits | C-000001-C-999999 | **Boolean** | Read/Write |
| Internal Float 32 | F32-000001-F32-999999 | **Float** | Read/Write |
| Internal Unsigned 8 | US8-000001-US8-999999 | **Byte** | Read/Write |
| Internal Signed 16 | S16-000001-S16-999999 | **Short** | Read/Write |
| Internal Signed 32 | S32-000001-S32-999999 | **Long** | Read/Write |
| String | STR-000001:1-STR-999999:128 | **String** | Read/Write |
| System String | SSTR-000001:1-SSTR-999999:50 | **String** | Read Only |
| System Read Only Bit | SBR-000001-SBR-999999 | **Boolean** | Read Only |
| System Read/Write Bit | SBRW-000001-SBRW-999999 | **Boolean** | Read/Write |
| System Read Only Word | SWR-000001-SWR-999999 | **Word** | Read Only |
| System Read/Write Word | SWRW-000001-SWRW-999999 | **Word** | Read/Write |
| Internal Unsigned 16 | US16-000001-US16-999999 | **Word** | Read/Write |

### One Dimensional Array Elements

| Device Type | Range | Data Type | Access |
|---|---|---|---|
| 1 Dimensional Internal BCD 16 | AR1BCD16-00001(1)-AR1BCD16-999999(65535) | **BCD** | Read/Write |
| 1 Dimensional Internal BCD 32 | AR1BCD32-00001(1)-AR1BCD32-999999 (65535) | **LBCD** | Read/Write |
| 1 Dimensional Internal Bits | AR1C-000001(1)-AR1C-999999(65535) | **Boolean** | Read/Write |
| 1 Dimensional Internal Float 32 | AR1F32-00001(1)-AR1F32-999999(65535) | **Float** | Read/Write |
| 1 Dimensional Internal Unsigned 8 | AR1US8-00001(1)-AR1US8-999999(65535) | **Byte** | Read/Write |
| 1 Dimensional Signed 16 | AR1S16-00001(1)-AR1S16-999999(65535) | **Short** | Read/Write |
| 1 Dimensional Signed 32 | AR1S32-00001(1)-AR1S32-999999(65535) | **Long** | Read/Write |
| 1 Dimensional String | AR1STR-00001:1(1)-AR1STR-999999:128 (65535) | **String** | Read/Write |
| 1 Dimensional Internal Unsigned 16 | AR1US16-00001(1)-AR1US16-999999(65535) | **Word** | Read/Write |

### Two Dimensional Array Elements

| Device Type | Range | Data Type | Access |
|---|---|---|---|
| 2 Dimensional Internal BCD 16 | AR2BCD16-00001(1),(1)-AR2BCD16-999999 (65535),(65535) | **BCD** | Read/Write |
| 2 Dimensional Internal BCD 32 | AR2BCD32-00001(1),(1)–AR2BCD32-999999 (65535),(65535) | **LBCD** | Read/Write |
| 2 Dimensional Internal Bits | AR2C-000001(1),(1)-AR2C-999999 (65535),(65535) | **Boolean** | Read/Write |
| 2 Dimensional Internal Float 32 | AR2F32-00001(1),(1)-AR2F32-999999 (65535),(65535) | **Float** | Read/Write |
| 2 Dimensional Unsigned Hex 8 | AR2US8-00001(1),(1)-AR2US8-999999 | **Byte** | Read/Write |

| | | | |
|---|---|---|---|
| | (65535),(65535) | | |
| 2 Dimensional Signed 16 | AR2S16-00001(1),(1)-AR2S16-999999 (65535),(65535) | **Short** | Read/Write |
| 2 Dimensional Signed 32 | AR2S32-00001(1),(1)-AR2S32-999999 (65535),(65535) | **Long** | Read/Write |
| 2 Dimensional String | AR2STR-00001:1(1),(1)–AR2STR-999999:128 (65535),(65535) | **String** | Read/Write |
| 2 Dimensional Internal Unsigned 16 | AR2US16-00001(1),(1)-AR2US16-999999 (65535),(65535) | **Word** | Read/Write |

**See Also:** Ordering of Array Data

## Ordering of Array Data

**1. Dimensional Arrays - array [dim1]**
1 dimensional array data is passed to and from the controller in ascending order:

```
for (dim1=0; dim1<dim1_max; dim1++)
```

**Example:** 3 element array

array [0]
array [1]
array [2]

**2. Dimensional Arrays - array [dim1, dim2]**
2 dimensional array data is passed to and from the controller in ascending order:

```
for (dim1=0; dim1<dim1_max; dim1++)
for (dim2=0; dim2<dim2_max; dim2++)
```

**Example:** 3X3 element array

array [0, 0]
array [0, 1]
array [0, 2]
array [1, 0]
array [1, 1]
array [1, 2]
array [2, 0]
array [2, 1]
array [2, 2]

## Address Formats

### Tag Names

The default tag names are assigned when hardware is added. Tag names for I/O modules generally follow the AAAAA-B.C.D.E format, which is described in the table below.

| Notation | Example | Notes |
|---|---|---|
| AAAAA-B.C.D.E<br><br>AAAAA = Category<br>B = Base Group#<br>C = Base#<br>D = Slot#<br>E = Point or Channel Number | DI-0.1.1.1 | Discrete Input P3-550 |

### Category

The category of data may also include the data type. Descriptions of the categories are as follows.

- **Discrete I/O points:**
  DI = Discrete Input Point
  DO = Discrete Output Point

- **Boolean:** This category includes data items other than I/O points. For example, a blown fuse or out-of-range error bits.
  MST = Module Status
- **Analog points:** This category uses five digits. The first two digits designate the type of point and the last three digits describe the data type being used.
  AIxxx = Analog Input Point
  AOxxx = Analog Output Point
  xxF32 = Floating Point 32 bit data type
  xxS32 = Signed 32 bit data type

### Base Group Number

The base group number is where the module resides. For the P3-550, the base group is group 0.

### Base Number

The base number within the group is where the module resides. For the P3-550 or P3-RS, the base is base 1.

### Slot Number

The slot number in the base is where the module resides. The slot to the right of the CPU is slot 1.

### Point or Channel Number

If the tag name is for an I/O point of a module, this digit will be the channel number. If the tag name is for a status indication, this digit will differentiate between them.

For example, a P3-16TR has one fuse on both commons and a blown fuse indicator bit for each. Their default tag names are MST-x.x.x.1 and MST-x.x.x.2. MST-x.x.x.1 is the blown fuse indicator for the fuse on the common of channels 1 through 8.

## Error Descriptions

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

### Automatic Tag Database Generation Error Messages

Tag name <tag name> encountered validation error and will not be imported
Unable to generate a tag database for device <device name>. Reason: Auto tag generation aborted
Unable to generate a tag database for device <device name>. Reason: Import file is invalid or corrupt
Unable to generate a tag database for device <device name>. Reason: Import file not found
Unable to generate a tag database for device <device name>. Reason: Low memory resources

### Driver Error Messages

Unable to bind to adapter: <network adapter>. Connect failed. Winsock Err # <Error number>
Winsock initialization failed (OS Error = <error code>)
Winsock shut down failed (OS Error = <error code>)
Winsock V1.1 or higher must be installed to use the Productivity 3000 Ethernet device driver

### Read Errors

Cannot read <tag count> items starting at tag '<tag address>': System ID '<System ID>' does not exist in device '<device name>'
Cannot read tag '<tag address>': System ID '<System ID>' does not exist in device '<device name>'
Cannot read <tag count> items starting at tag '<tag address>': address does not exist in device '<device name>'
Cannot read tag '<tag address>': address does not exist in device '<device name>'
Cannot read <tag count> items starting at tag '<tag address>': value is invalid for data type '<data type>' in device '<device name>'
Cannot read tag '<tag address>': value is invalid for data type '<data type>' in device '<device name>'
Cannot read <tag count> items starting at tag '<tag address>': error receiving response frame from device '<device name>'
Cannot read tag '<tag address>': error receiving response frame from device '<device name>'
Cannot read <tag count> items starting at tag '<tag address>': device '<device name>' returned error code <error code>
Cannot read tag '<tag address>': device '<device name>' returned with error code <error code>

### Write Errors

Cannot write to tag '<tag address>': System ID '<System ID>' does not exist in device '<device name>'
Cannot write to tag '<tag address>': address does not exist in device '<device name>
Cannot write to tag '<tag address>': value is invalid for data type '<data type>' in device '<device name>'
Cannot write to tag '<tag address>': error receiving response frame from device '<device name>'
Cannot write to tag '<tag address>': device '<device name>' returned with error code <error code>

### See Also:

Modbus Exception Codes

## Modbus Exception Codes

The following data is from Modbus Application Protocol Specifications documentation.

| Code Dec/Hex | Name | Description |
|---|---|---|
| 01/0x01 | ILLEGAL FUNCTION | The function code received in the query is not an allowable action for the server (or slave). This may be because the function code is only applicable to newer devices and was not implemented in the unit selected. It could also indicate that the server (or slave) is in the wrong state to process a request of this type. For example, because it is unconfigured and is being asked to return register values. |
| 02/0x02 | ILLEGAL DATA ADDRESS | The data address received in the query is not an allowable address for the server (or slave). More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed, a request with offset 96 and length 5 will generate exception 02. |

| | | |
|---|---|---|
| 03/0x03 | ILLEGAL DATA VALUE | A value contained in the query data field is not an allowable value for server (or slave). This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does not mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the Modbus protocol is unaware of the significance of any particular value of any particular register. |

## Automatic Tag Database Generation Error Messages

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

**Automatic Tag Database Generation Error Messages**
**Tag name <tag name> encountered validation error and will not be imported**
**Unable to generate a tag database for device <device name>. Reason: Auto tag generation aborted**
**Unable to generate a tag database for device <device name>. Reason: Import file is invalid or corrupt**
**Unable to generate a tag database for device <device name>. Reason: Import file not found**
**Unable to generate a tag database for device <device name>. Reason: Low memory resources**

## Tag name <tag name> encountered validation error and will not be imported

**Error Type:**
Warning

**Possible Cause:**
The tag import file contains tag(s) that have invalid character(s). This error message will be returned for each tag that contains an invalid character.

**Solution:**
Remove the invalid character(s).

## Unable to generate a tag database for device <device name>. Reason: Auto tag generation aborted

**Error Type:**
Warning

**Possible Cause:**
The Automatic Tag Generation process was aborted.

**Solution:**
Retry the Automatic Tag Generation process.

**See Also:** Automatic Tag Database Generation

## Unable to generate a tag database for device <device name>. Reason: Import file is invalid or corrupt

**Error Type:**
Warning

**Possible Cause:**
The Tag Import File is a corrupt project file.

**Solution:**
In the OPC server project, right-click on the device and then select **Properties** from the context menu. Next, click on the **Tag Import File Path** tab. Select a valid, properly formatted **Productivity Suite Programming Software Import File**, or produce a new import file by retrying the tag export process in the application.

## Unable to generate a tag database for device <device name>. Reason: Import file not found

**Error Type:**
Warning

### Possible Cause:
The Tag Import File could not be found.

### Solution:
In the OPC server project, right-click on the device and then select **Properties** from the context menu. Next, click on the **Tag Import File Path** tab. Select a valid, properly formatted **Productivity Suite Programming Software Import File**, or produce a new import file by retrying the tag export process in the application.

## Unable to generate a tag database for device <device name>. Reason: Low memory resources

### Error Type:
Warning

### Possible Cause:
Memory required for database generation could not be allocated. The process is aborted.

### Solution:
Close unused applications and/or increase the amount of virtual memory and try again.

## Driver Error Messages

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

### Driver Error Messages
Unable to bind to adapter: <network adapter>. Connect failed. Winsock Err # <Error number>
Winsock initialization failed (OS Error = <error code>)
Winsock shut down failed (OS Error = <error code>)
Winsock V1.1 or higher must be installed to use the Productivity 3000 Ethernet device driver

## Unable to bind to adapter: <network adapter>. Connect failed. Winsock Err# <error number>

### Error Type:
Fatal

### Possible Cause:
1. The operating system could not find an unused port to use for communication with this device.
2. Network system failure, such as Winsock or network adapter failure.
3. Other applications have claimed all available ports (possible but unlikely).

### Solution:
1. Reboot the computer and check the network adapter.
2. Check for applications that could be causing conflicts and shut them down.

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

### Error Type:
Fatal

| OS Error | Possible Solution |
|---|---|
| 10036 | The network subsystem is still busy with unfinished processing. Wait a few seconds and restart the driver. |
| 10050 | The network subsystem has failed. Refer to the Network Administrator. |
| 10093 | The network subsystem was not initialized before the shutdown was attempted. Wait a few seconds and try again. |

## Winsock V1.1 or higher must be installed to use the Productivity 3000 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.

## Read Errors

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

**Read Errors**
**Cannot read <tag count> items starting at tag '<tag address>': System ID '<System ID>' does not exist in device '<device name>'**
**Cannot read tag '<tag address>': System ID '<System ID>' does not exist in device '<device name>'**
**Cannot read <tag count> items starting at tag '<tag address>': address does not exist in device '<device name>'**
**Cannot read tag '<tag address>': address does not exist in device '<device name>'**
**Cannot read <tag count> items starting at tag '<tag address>': value is invalid for data type '<data type>' in device '<device name>'**
**Cannot read tag '<tag address>': value is invalid for data type '<data type>' in device '<device name>'**
**Cannot read <tag count> items starting at tag '<tag address>': error receiving response frame from device '<device name>'**
**Cannot read tag '<tag address>': error receiving response frame from device '<device name>'**
**Cannot read <tag count> items starting at tag '<tag address>': device '<device name>' returned error code <error code>**
**Cannot read tag '<tag address>': device '<device name>' returned with error code <error code>**

## Cannot read <tag count> items starting at tag '<tag address>': System ID '<System ID>' does not exist in device '<device name>'

**Error Type:**
Warning

**Possible Cause:**
The PLC returned an error code of 0x01 (illegal function) for a blocked Read transaction.

**Solution:**
Replace the current System ID with a valid one (as listed in the device protocol).

## Cannot read tag '<tag address>': System ID '<System ID>' does not exist in device '<device name>'

**Error Type:**
Warning

**Possible Cause:**
The PLC returned an error code of 0x01 (illegal function) for an unblocked Read transaction.

**Solution:**
Replace the current System ID with a valid one (as listed in the device protocol).

### Cannot read <tag count> items starting at tag '<tag address>': address does not exist in device '<device name>'

**Error Type:**
Warning

**Possible Cause:**
The PLC returned an error code of 0x02 (illegal data address) for a blocked Read transaction.

**Solution:**
Replace the current data address with one within the valid range (as listed in the device protocol).

### Cannot read tag '<tag address>': address does not exist in device '<device name>'

**Error Type:**
Warning

**Possible Cause:**
The PLC returned an error code of 0x02 (illegal data address) for an unblocked Read transaction.

**Solution:**
Replace the current data address with one within the valid range (as listed in the device protocol).

### Cannot read <tag count> items starting at tag '<tag address>': value is invalid for data type '<data type>' in device '<device name>'

**Error Type:**
Warning

**Possible Cause:**
The PLC returned an error code of 0x03 (illegal data value) for a blocked Read transaction.

**Solution:**
Replace the current data with valid values (depending on the data type).

### Cannot read tag '<tag address>': value is invalid for data type '<data type>' in device '<device name>'

**Error Type:**
Warning

**Possible Cause:**
The PLC returned an error code of 0x03 (illegal data value) for an unblocked Read transaction.

**Solution:**
Replace the current data with valid values (depending on the data type).

### Cannot read <tag count> items starting at tag '<tag address>': error receiving response frame from device '<device name>'

**Error Type:**
Warning

**Possible Cause:**
The response frame received from the PLC contains an error.

**Solution:**
Check the connection to the PLC and then resend the data.

## Cannot read tag '<tag address>': error receiving response frame from device '<device name>'

**Error Type:**
Warning

**Possible Cause:**
The response frame received from the PLC contains an error.

**Solution:**
Check the connection to the PLC and then resend the data.

## Cannot read <tag count> items starting at tag '<tag address>': device '<device name>' returned error code <error code>

**Error Type:**
Warning

**Possible Cause:**
The PLC has returned an error code.

**Solution:**
Refer to the list of error code descriptions located in the PLC's manual.

## Cannot read tag '<tag address>': device '<device name>' returned with error code <error code>

**Error Type:**
Warning

**Possible Cause:**
The PLC has returned an error code.

**Solution:**
Refer to the list of error code descriptions located in the PLC's manual.

## Write Errors

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

**Write Errors**
Cannot write to tag '<tag address>': System ID '<System ID>' does not exist in device '<device name>'
Cannot write to tag '<tag address>': address does not exist in device '<device name>
Cannot write to tag '<tag address>': value is invalid for data type '<data type>' in device '<device name>'
Cannot write to tag '<tag address>': error receiving response frame from device '<device name>'
Cannot write to tag '<tag address>': device '<device name>' returned with error code <error code>

## Cannot write to tag '<tag address>': System ID '<System ID>' does not exist in device '<device name>'

**Error Type:**
Warning

**Possible Cause:**
The PLC returned an error code of 0x01 (illegal function) for a Write transaction.

**Solution:**
Replace the current System ID with a valid one (as listed in the device protocol).

## Cannot write to tag '<tag address>': address does not exist in device '<device name>'

**Error Type:**
Warning

**Possible Cause:**
The PLC returned an error code of 0x02 (illegal data address) for a Write transaction.

**Solution:**
Replace the current data address with one within the valid range (as listed in the device protocol).

## Cannot write to tag '<tag address>': value is invalid for data type '<data type>' in device '<device name>'

**Error Type:**
Warning

**Possible Cause:**
The PLC returned an error code of 0x03 (illegal data value) for a Write transaction.

**Solution:**
Replace the current data with valid values (depending on the data type).

## Cannot write to tag '<tag address>': error receiving response frame from device '<device name>'

**Error Type:**
Warning

**Possible Cause:**
The response frame received from the PLC contains an error.

**Solution:**
Check the connection to the PLC and then resend the data.

## Cannot write to tag '<tag address>': device '<device name>' returned with error code <error code>

**Error Type:**
Warning

**Possible Cause:**
The PLC has returned an error code.

**Solution:**
Refer to the list of error code descriptions located in the PLC's manual.

# Index