

# **AutomationDirect ECOM Driver Help**

© 2012 Kepware Technologies

# Table of Contents

Table of Contents .....	2
AutomationDirect ECOM Driver Help .....	4
Overview .....	4
Device Setup .....	5
Communication Parameters .....	6
Tag Import Settings .....	6
Automatic Tag Database Generation .....	7
Tag Hierarchy .....	7
Import File-to-Server Name Conversions .....	7
Importing DirectSoft Elements .....	7
Import Preparation: DirectSoft Steps .....	7
Import Preparation: OPC Server Steps .....	10
Optimizing the AutomationDirect ECOM Communications .....	11
Data Types Description .....	12
Address Descriptions .....	13
DL-05 Addressing .....	13
DL-06 Addressing .....	14
DL-240 Addressing .....	15
DL-250(-1) Addressing .....	16
DL-260 Addressing .....	17
DL-430 Addressing .....	19
DL-440 Addressing .....	20
DL-450 Addressing .....	21
Error Descriptions .....	23
Address Validation .....	23
Missing address .....	23
Device address '<address>' contains a syntax error .....	23
Address '<address>' is out of range for the specified device or register .....	23
Device address '<address>' is not supported by model '<model name>' .....	24
Data Type '<type>' is not valid for device address '<address>' .....	24
Device address '<address>' is Read Only .....	24
Device Status Messages .....	24
Device '<device name>' not responding .....	24
Unable to write to '<address>' on device '<device name>' .....	25
Device Specific Messages .....	25
Winsock initialization failed (OS Error = n) .....	25
Winsock V1.1 or higher must be installed to use the AutomationDirect ECOM device driver .....	25
Bad address in block [<start address> to <end address>] on device '<device name>' .....	25
Automatic Tag Database Generation Messages .....	26

Unable to generate a tag database for device <device name>. Reason: Low memory resources..... 26

Unable to generate a tag database for device <device name>. Reason: Import file is invalid or corrupt... 26

**Index** ..... 27

## AutomationDirect ECOM Driver Help

---

Help version 1.013

### CONTENTS

#### [Overview](#)

What is the AutomationDirect ECOM Driver?

#### [Device Setup](#)

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

#### [Automatic Tag Database Generation](#)

How can I easily configure tags for the AutomationDirect ECOM driver?

#### [Optimizing the AutomationDirect ECOM Communications](#)

How do I get the best performance from the AutomationDirect ECOM driver?

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

What data types does this driver support?

#### [Address Descriptions](#)

How do I address a data location on an AutomationDirect device?

#### [Error Descriptions](#)

What error messages does the AutomationDirect ECOM driver produce?

### Overview

---

The AutomationDirect ECOM Driver provides an easy and reliable way to connect AutomationDirect ECOM controllers to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It is intended for use with AutomationDirect Programmable Logic Controllers that may be accessed via an ECOM Ethernet module.

## Device Setup

---

### Supported Devices\*

DL-05  
DL-06  
DL-240  
DL-250(-1)  
DL-260  
DL-430  
DL-440  
DL-450

\*All PLCs via an Hx-ECOM module.

### Communication Protocol

Ethernet using Winsock V1.1 or higher.

### Connection Timeout

This parameter 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 60 seconds.

### Request Timeout

This parameter specifies the time 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 250 milliseconds. The valid range is 50 to 9999 milliseconds.

### Retry Attempts

This parameter 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 IDs

Up to 1024 devices may be defined on a given channel. Each device on the channel must be uniquely identified by its own IP address. In general the Device ID has the following format `YYY.YYY.YYY.YYY` where `YYY` designates the device IP address. Each `YYY` byte should be in the range of 0 to 255.

An ECOM module's IP address can be determined using NetEdit, an AutomationDirect device configuration utility. To launch NetEdit, select the Device ID Wizard button on the General tab in Device Properties.



**Note:** NetEdit has the ability to query the network, configure network parameters and update firmware for ECOM devices.

### Automatic Tag Database Generation

[Tag Import Settings](#)

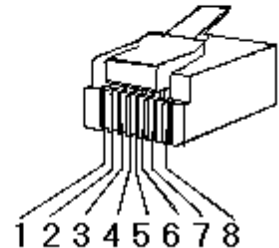
### Cable Diagrams

## Patch Cable (Straight Through)

TD + 1	OR/WHT	OR/WHT	1 TD +
TD - 2	OR	OR	2 TD -
RD + 3	GRN/WHT	GRN/WHT	3 RD +
4	BLU	BLU	4
5	BLU/WHT	BLU/WHT	5
RD - 6	GRN	GRN	6 RD -
7	BRN/WHT	BRN/WHT	7
8	BRN	BRN	8

RJ45 RJ45

## 10 BaseT



## Crossover Cable

TD + 1	OR/WHT	GRN/WHT	1 TD +
TD - 2	OR	GRN	2 TD -
RD + 3	GRN/WHT	OR/WHT	3 RD +
4	BLU	BLU	4
5	BLU/WHT	BLU/WHT	5
RD - 6	GRN	OR	6 RD -
7	BRN/WHT	BRN/WHT	7
8	BRN	BRN	8

RJ45 RJ45

## 8-pin RJ45

## Communication Parameters

### Port

This parameter specifies the port number that the remote device is configured to use. The default port number is 28784 (0x7070).

### Tag Import Settings

#### Tag Import File

This parameter is used to specify the exact location of the DirectSoft export file from which tags will be imported. This file will be used when Automatic Tag Database Generation is instructed to create the tag database. The two files types that can be imported are Supported and Not Supported.

#### Supported Import Files

Program (via Export), .txt extension  
Element Documentation (via Export), Standard Format, .csv extension

#### Import Files Not Supported

Element Documentation (via Export), Standard Format, .txt extension  
Element Documentation (via Export), EZ-Touch Format, .csv and .txt extension  
Element Documentation (auto created), .esd extension  
DirectSoft Project, .prj extension

#### Display Descriptions

When checked, this option will import tag descriptions. A description will be given to tags with long names that states the original tag name when necessary.

**See Also:** [Automatic Tag Database Generation](#)

## Automatic Tag Database Generation

---

The AutomationDirect ECOM Driver generates its tags offline, meaning that a connection to the device is not required to generate tags. Automatic Tag Database Generation is a two step procedure. First, DirectSoft is used to create a tag export file. Then, the export file is accessed from within the OPC server in order to generate tags. For more information on creating an export file (\*.txt or \*.csv) from DirectSoft, refer to [Import Preparation: DirectSoft Steps](#). For more information on configuring the OPC server to use the DirectSoft file for automatic tag database generation, refer to [Import Preparation: OPC Server Steps](#).

See Also: [Tag Import Settings](#)

## Tag Hierarchy

---

The tags created by Automatic Tag Database Generation follow a specific hierarchy. The root level groups (or subgroup level of the group specified in "Add generated tags to the following group") are determined by the tag's memory type referenced (such as X, C, V and etc.).

### Example

Every variable that is of address type "X" will be placed in a root level group called "X." The only exception applies to counter and timer accumulator addresses CTA and TA respectively. In these cases, the address is converted to a V-memory reference (TA0 = V0) but the tags generated will be assigned to the root level group CTA or TA, not V. But explicit V-memory references to CTA and TA locations will be assigned to the root level group V as intended.

## Import File-to-Server Name Conversions

---

### Leading Underscores

Leading underscores (\_) in tag names will be removed. This is required since the server does not accept tag names beginning with an underscore.

### Invalid Characters in Name

The only characters allowed in the server tag name are A-Z, a-z, 0-9, and underscore (\_). As mentioned above, a tag name cannot begin with an underscore. All other invalid characters encountered will be removed from the tag name.

## Importing DirectSoft Elements

---

This driver uses files generated from DirectSoft via the Program or Element Documentation Export feature to generate the tag database. In both methods, the items of interest are the Elements (such as nickname, address and description) that were created in the DirectSoft Documentation Editor.

### How do I create a DirectSoft tag import file (\*.txt or \*.csv)?

See [Import Preparation: DirectSoft Steps](#)

### How do I configure the OPC Server to use this import file for Automatic Tag Database Generation?

See [Import Preparation: OPC Server Steps](#)

## Import Preparation: DirectSoft Steps

---

There are two supported methods for generating an export file in DirectSoft for the driver to use as a tag import file: Program Export (\*.txt extension) and Element Documentation Export, Standard Format (\*.csv extension).

### Step 1: Create Nicknames

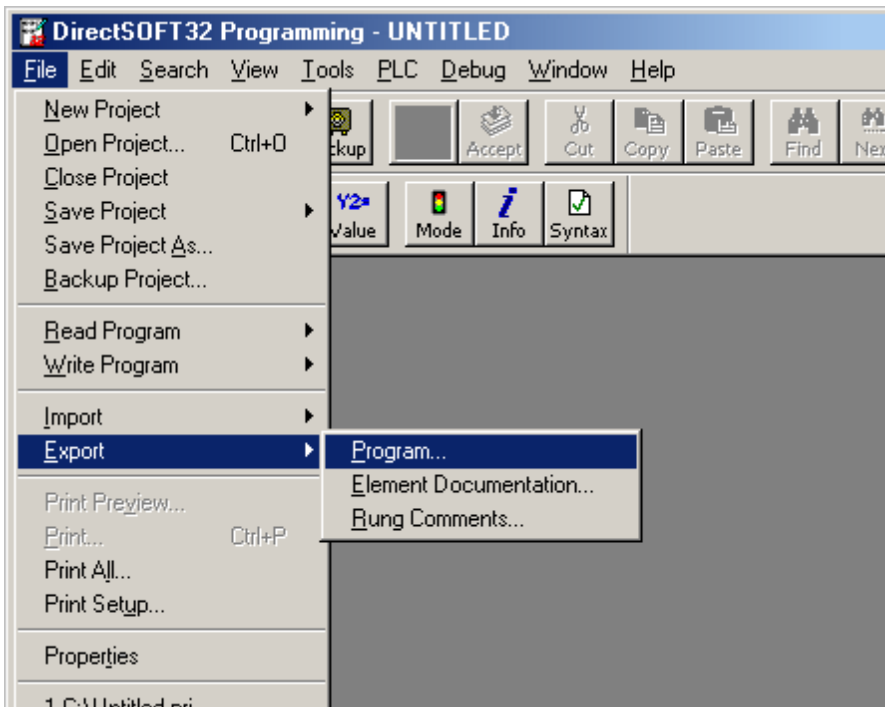
1. Open the DirectSoft project containing the tags (elements) that will be ported to the OPC Server.
2. Launch the **Documentation Editor** by clicking **Tools | Documentation Editor**.
3. For each **Memory Reference** of interest, enter both a **Nickname** and a **Description**.

Element	Nickname	Wiring Info	Description
X27	Alarm1		Alarm 1 set
X30			
X31			
X32			
X33			
X34			
X35			
X36			
X37			
X40			
X41			
X42			
X43			
X44			
X45			
X46			
X47			
X50			
X51			
X52			
X53			
X54			
X55			

## Step 2: Export the Elements

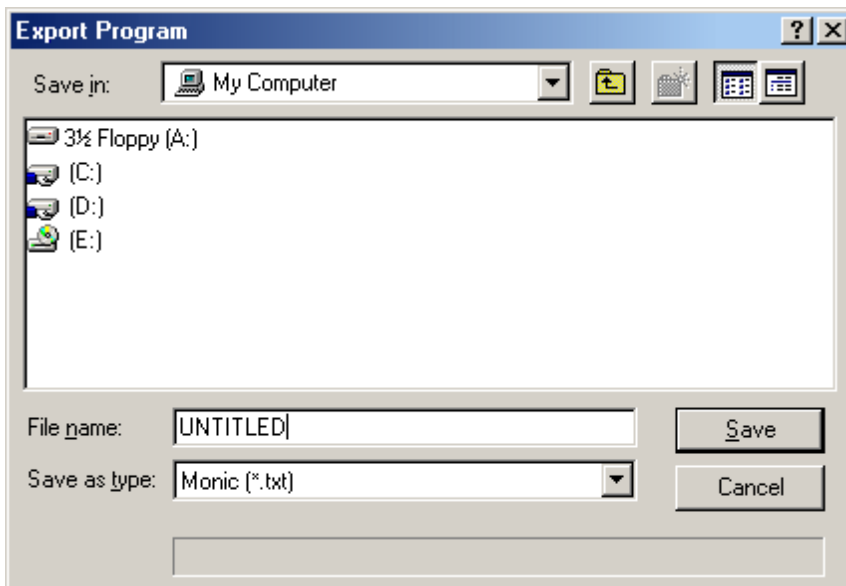
### Program Export (.txt)

1. In DirectSoft, click **File** | **Export**. Then, select **Program**.



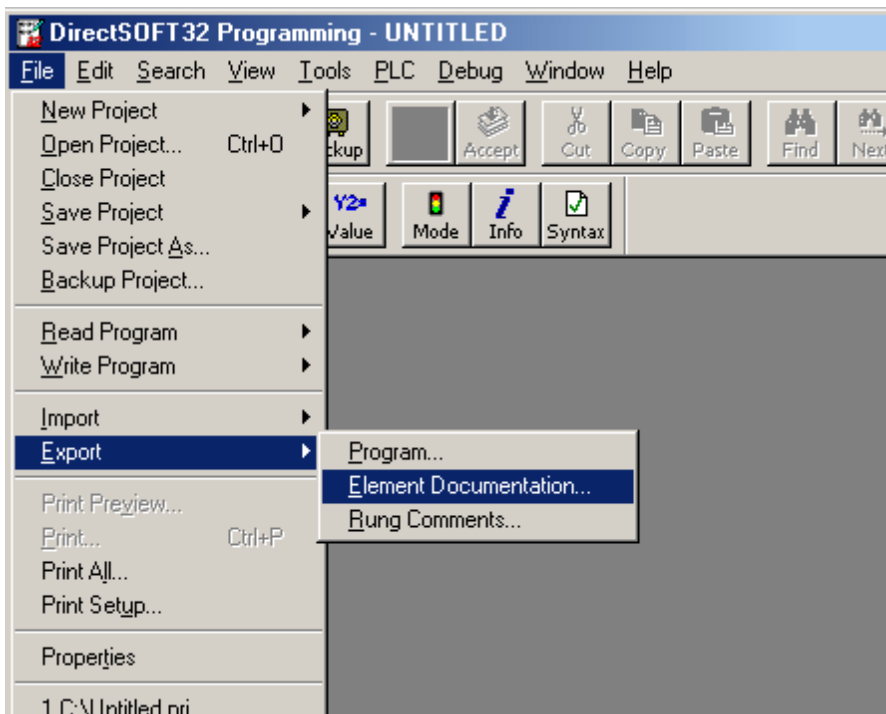


2. The Save dialog will appear, showing the file in text (\*.txt) format.

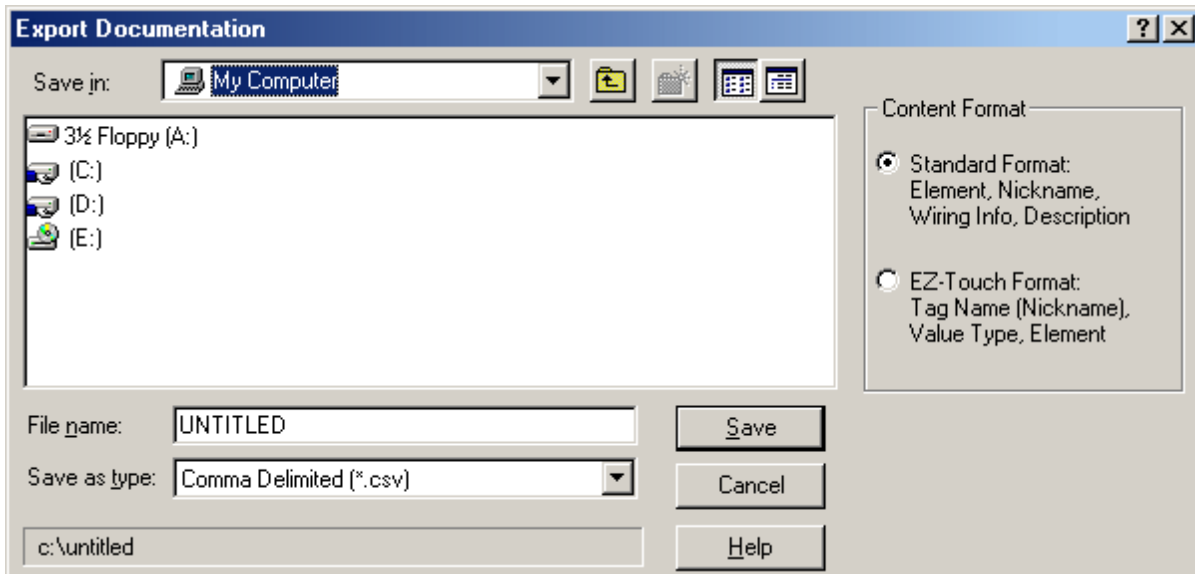


### Element Documentation Export (.csv)

1. In DirectSoft, click **File | Export**. Then, select **Element Documentation**.



2. The Save dialog will appear. Select **Comma Delimited (\*.csv)** and **Standard Format**.



**Note:** Any other format or file type will not import properly. The file will be in comma separated variable format.

### Import Preparation: OPC Server Steps

An export file from DirectSoft must be created before completing the following steps. For more information, refer to [Import Preparation: DirectSoft Steps](#).

1. In the driver, click on the device of interest (for which tags will be generated).
2. Next, open its **Device Properties** dialog and select the **Tag Import Settings** tab.
3. Browse and select the location of the DirectSoft export file previously created and then click **Apply**.
4. Next, select the **Database Creation** tab and configure the Database Creation settings.
5. Click **Auto Create** to create the tag database.
6. The OPC Server will then attempt to create the tag database while posting messages to the event log on the status of the import. When finished, it will state that the tag import has completed. All elements exported out of DirectSoft will appear in the OPC Server in the layout discussed in [Tag Hierarchy](#).

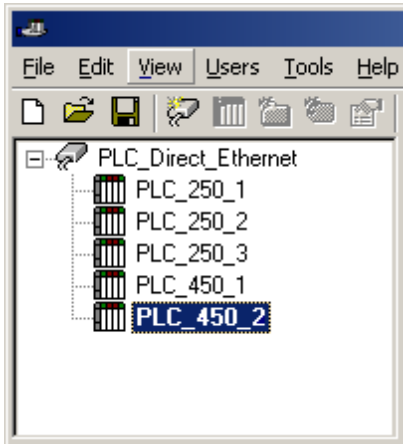
**Note:** The OPC tags generated are given meaningful names in the OPC Server that are based on the variables imported. These tags are also placed in meaningful tag groups to provide a structured and manageable interface. The end result is a well-organized OPC Server project that directly reflects the variable import file.

**See Also:** [Import File-To-Server Name Conversions](#) and [Tag Import Settings](#).

## Optimizing the AutomationDirect ECOM Communications

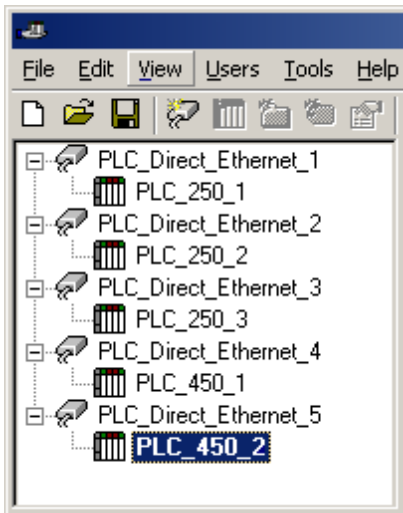
The AutomationDirect ECOM driver has been designed to provide the best performance with the least amount of impact on the system's overall performance. While the AutomationDirect ECOM driver is fast, there are a couple of guidelines that can be used in order to control and optimize the application and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

Our server refers to communications protocols like AutomationDirect ECOM 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 AutomationDirect ECOM 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 ECOM 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 AutomationDirect ECOM 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 ECOM driver could only define one single channel, then the example shown above would be the only option available; however, the AutomationDirect ECOM driver can define up to 16 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 16 or fewer devices, it can be optimized exactly how it is shown here.

The performance will improve even if the application has more than 16 devices. While 16 or fewer devices may be ideal, the application will still benefit from additional channels. Although by 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 low bit bit 15 is the high bit
Short	Signed 16 bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32 bit value bit 0 is the low bit bit 31 is the high bit
Long	Signed 32 bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
Float	32 bit floating point value.  The driver interprets two consecutive registers as a floating-point value by making the second register the high word and the first register the low word.
BCD	Two byte packed BCD  Value range is 0-9999. Behavior is undefined for values beyond this range.
String	Null terminated ASCII string. 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.

[DL-05](#)

[DL-06](#)

[DL-240](#)

[DL-250\(-1\)](#)

[DL-260](#)

[DL-430](#)

[DL-440](#)

[DL-450](#)

## DL-05 Addressing

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

Memory Type	Reference	Data Types
I/O	X,Y	<b>Boolean</b>
Devices	C,SP,T,CT,S	<b>Boolean</b>
Data Words	V	Boolean, Word, <b>Short</b> , DWord, Long, LBCD, Float, String, <b>BCD*</b>

\*The default is for Timers and Counters only.

### Bit Access to V Memory

Bit information can be directly accessed within V memory registers. To access a bit within a V memory register, a bit number can be appended to any V memory address. V memory addressing with bit access would appear as follows: V<xxxxx>.<yy> where xxxxx is the V memory register location and yy is the bit number (0 to 15) within that register. If the V memory location is either a Long or DWord, the bit number yy can be (0 to 31). For more information, refer to the examples below.

### Array Support for Data Words

This driver supports array notation for V memory Data Word addresses. To specify an array, append the array size to the address specification as follows: address[array size] or address[rows][cols]. Array size is limited to 63 elements when referenced as a Word, Short, and BCD, and 31 elements when referenced as a DWord, Long, Float, and LBCD.

### Examples

- V1400 [63] @ Word - Array of 63 Words (Maximum allowed) starting at V1400.
- V1400 [31] @ DWord - Array of 31 DWords (Maximum allowed) starting at V1400.

### String Access to Data Words

This driver supports reading and writing V memory Data Words as an ASCII string. When using V memory 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 126 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.

**Note:** The references are in Octal format.

Memory Type	Discrete Memory Reference	Word Memory Reference
Input Points	X0-X377	V40400-V40417
Output Points	Y0-Y377	V40500-V40517
Control Relays	C0-C777	V40600-V40637
Special Relays	SP0-SP777	V41200-V41237
Timer Status Bits	T0-T177	V41100-V41107
Timer Current Values	N/A	V0-V177
Counter Status Bits	CT0-CT177	V41140-V41147
Counter Current Values	N/A	V1000-V1177
Data Words	N/A	V1200-V7377
Data Words	N/A	V1200.2H-V7377.126H

String Access HiLo Byte Ordering		.Bit is string length, range 2 to 126 bytes.
Data Words	N/A	V1200.2L-V7377.126L
String Access LoHi Byte Ordering		.Bit is string length, range 2 to 126 bytes.
Data Words (Non-volatile)	N/A	V7400-V7577
Data Words (Non-volatile)	N/A	V7400.2H-V7577.126H
String Access HiLo Byte Ordering		.Bit is string length, range 2 to 126 bytes.
Data Words (Non-volatile)	N/A	V7400.2L-V7577.126L
String Access LoHi Byte Ordering		.Bit is string length, range 2 to 126 bytes.
Stages	S0-S377	V41000-V41017
System Parameters	N/A	V7600-V7777

### Examples

Example	Description
V40401	Bits 20-27 (octal) of X Input.
V41100	Timer status bits 0-17 (octal).
V1200.1	Bit access to V1200 bit 1.
V1200.100H	String with length 100 and HiLo byte ordering starting at V1200.
V1500.78L	String with length 78 and LoHi byte ordering starting at V1500.

### DL-06 Addressing

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

Memory Type	Reference	Data Types
I/O	X,Y,GX,GY	<b>Boolean</b>
Devices	C,SP,T,CT,S	<b>Boolean</b>
Data Words	V	Boolean, Word, <b>Short</b> , DWord, Long, LBCD, Float, String, <b>BCD*</b>

\*The default is for Timers and Counters only.

### Bit Access to V Memory

Bit information can be directly accessed within V memory registers. To access a bit within a V memory register, a bit number can be appended to any V memory address. V memory addressing with bit access would appear as follows: V<xxxxx>.<yy> where xxxxx is the V memory register location and yy is the bit number (0 to 15) within that register. If the V memory location is either a Long or DWord, the bit number yy can be (0 to 31). For more information, refer to the examples below.

### Array Support for Data Words

This driver supports array notation for V memory Data Word addresses. To specify an array, append the array size to the address specification as follows: address[array size] or address[rows][cols]. Array size is limited to 63 elements when referenced as a Word, Short, and BCD, and 31 elements when referenced as a DWord, Long, Float, and LBCD.

### Examples

- V1200 [63] @ Word - Array of 63 Words (Maximum allowed) starting at V1200.
- V1200 [31] @ DWord - Array of 31 DWords (Maximum allowed) starting at V1200.

### String Access to Data Words

This driver supports reading and writing V memory Data Words as an ASCII string. When using V memory 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 126 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.

**Note:** The references are in Octal format.

Memory Type	Discrete Memory Reference	Word Memory Reference
Input Points	X0-X777	V40400-V40437
Output Points	Y0-Y777	V40500-V40537
Control Relays	C0-C1777	V40600-V40677

Special Relays	SP0-SP777	V41200-V41237
Timer Status Bits	T0-T377	V41100-V41117
Timer Current Values	N/A	V0-V377
Counter Status Bits	CT0-CT177	V41040-V41147
Counter Current Values	N/A	V1000-V1177
Data Words	N/A	V400-V677 V1200-V7377 V10000-V17777
Data Words String Access HiLo Byte Ordering	N/A	V400.2H-V677.126H V1200.2H-V7377.126H V10000.2H-V17777.126H .Bit is string length, range 2 to 126 bytes.
Data Words String Access LoHi Byte Ordering	N/A	V400.2L-V677.126L V1200.2L-V7377.126L V10000.2L-V17777.126L .Bit is string length, range 2 to 126 bytes.
Data Words (Non-volatile)	N/A	V7400-V7577
Data Words (Non-volatile) String Access HiLo Byte Ordering	N/A	V7400.2H-V7577.126H .Bit is string length, range 2 to 126 bytes.
Data Words (Non-volatile) String Access LoHi Byte Ordering	N/A	V7400.2L-V7577.126L .Bit is string length, range 2 to 126 bytes.
Stages	S0-S1777	V41000-V41077
Remote I/O	GX0-GX3777 GY0-GY3777	V40000-V40177 V40200-V40377
System Parameters	N/A	V700-V737 V7600-V7777 V36000-V37777

### Examples

Example	Description
V40401	Bits 20-27 (octal) of X Input.
V41100	Timer status bits 0-17 (octal).
V700	System parameter word 700.
V2000.1	Bit access to V2000 bit 1.
V1200.100H	String with length 100 and HiLo byte ordering starting at V1200.
V1500.78L	String with length 78 and LoHi byte ordering starting at V1500.

### DL-240 Addressing

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

Memory Type	Reference	Data Types
I/O	X,Y	<b>Boolean</b>
Devices	C,SP,T,CT,S	<b>Boolean</b>
Data Words	V	Boolean, Word, <b>Short</b> , DWord, Long, LBCD, Float, String, <b>BCD*</b>

\*The default is for Timers and Counters only.

#### Bit Access to V Memory

Bit information can be directly accessed within V memory registers. To access a bit within a V memory register, a bit number can be appended to any V memory address. V memory addressing with bit access would appear as follows: V<xxxxx>.<yy> where xxxxx is the V memory register location and yy is the bit number (0 to 15) within that register. If the V memory location is either a Long or DWord, the bit number yy can be (0 to 31). For more information, refer to the examples below.

#### Array Support for Data Words

This driver supports array notation for V memory Data Word addresses. To specify an array, append the array size to the address specification as follows: address[array size] or address[rows][cols]. Array size is limited to 63 elements when referenced as a Word, Short, and BCD, and 31 elements when referenced as a DWord, Long, Float, and LBCD.

**Examples**

1. V2000 [63] @ Word - Array of 63 Words (Maximum allowed) starting at V2000.
2. V2000 [31] @ DWord - Array of 31 DWords (Maximum allowed) starting at V2000.

**String Access to Data Words**

This driver supports reading and writing V memory Data Words as an ASCII string. When using V memory 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 126 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.

**Note:** The references are in Octal format.

Memory Type	Discrete Memory Reference	Word Memory Reference
Input Points	X0-X477	V40400-V40423
Output Points	Y0-Y477	V40500-V40523
Control Relays	C0-C377	V40600-V40617
Special Relays	SP0-SP137 SP540-SP617	V41200-V41205 V41226-V41230
Timer Status Bits	T0-T177	V41100-V41107
Timer Current Values	N/A	V0-V177
Counter Status Bits	CT0-CT177	V41140-V41147
Counter Current Values	N/A	V1000-V1177
Data Words	N/A	V2000-V3777
Data Words String Access HiLo Byte Ordering	N/A	V2000.2H-V3777.126H .Bit is string length, range 2 to 126 bytes.
Data Words String Access LoHi Byte Ordering	N/A	V2000.2L-V3777.126L .Bit is string length, range 2 to 126 bytes.
Data Words (Non-volatile)	N/A	V4000-V4377
Data Words (Non-volatile) String Access HiLo Byte Ordering	N/A	V4000.2H-V4377.126H .Bit is string length, range 2 to 126 bytes.
Data Words (Non-volatile) String Access LoHi Byte Ordering	N/A	V4000.2L-V4377.126L .Bit is string length, range 2 to 126 bytes.
Stages	S0-S777	V41000-V41037
System Parameters	N/A	V7620-V7737 V7746-V7777

**Examples**

Example	Description
V40500	bits 0-17 (octal) of Y Output.
CT65	Counter contact 65.
S57	Stage control bit 57.
V2000.1	Bit access to V2000 bit 1.
V2000.100H	String with length 100 and HiLo byte ordering starting at V2000.
V2000.78L	String with length 78 and LoHi byte ordering starting at V2000.

**DL-250(-1) Addressing**

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

Memory Type	Reference	Data Types
I/O	X,Y	<b>Boolean</b>
Devices	C,SP,T,CT,S	<b>Boolean</b>
Data Words	V	Boolean, Word, <b>Short</b> , DWord, Long, LBCD, Float, String, <b>BCD*</b>

\*The default is for Timers and Counters only.

**Bit Access to V Memory**



Bit information can be directly accessed within V memory registers. To access a bit within a V memory register, a bit number can be appended to any V memory address. V memory addressing with bit access would appear as follows: V<xxxxx>.<yy> where xxxxx is the V memory register location and yy is the bit number (0 to 15) within that register. If the V memory location is either a Long or DWord, the bit number yy can be (0 to 31). For more information, refer to the examples below.

### Array Support for Data Words

This driver supports array notation for V memory Data Word addresses. To specify an array, append the array size to the address specification as follows: address[array size] or address[rows][cols]. Array size is limited to 63 elements when referenced as a Word, Short, and BCD, and 31 elements when referenced as a DWord, Long, Float, and LBCD.

#### Examples

1. V1400 [63] @ Word - Array of 63 Words (Maximum allowed) starting at V1400.
2. V1400 [31] @ DWord - Array of 31 DWords (Maximum allowed) starting at V1400.

### String Access to Data Words

This driver supports reading and writing V memory Data Words as an ASCII string. When using V memory 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 126 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.

**Note:** The references are in Octal format.

Memory Type	Discrete Memory Reference	Word Memory Reference
Input Points	X0-X777	V40400-V40437
Output Points	Y0-Y777	V40500-V40537
Control Relays	C0-C1777	V40600-V40677
Special Relays	SP0-SP777	V41200-V41237
Timer Status Bits	T0-T377	V41100-V41117
Timer Current Values	N/A	V0-V377
Counter Status Bits	CT0-CT177	V41140-V41147
Counter Current Values	N/A	V1000-V1177
Data Words	N/A	V1400-V7377 V10000-V17777
Data Words String Access HiLo Byte Ordering	N/A	V1400.2H-V7377.126H V10000.2H-V17777.126H .Bit is string length, range 2 to 126 bytes.
Data Words String Access LoHi Byte Ordering	N/A	V1400.2L-V7377.126L V10000.2L-V17777.126L .Bit is string length, range 2 to 126 bytes.
Stages	S0-S1777	V41000-V41077
System Parameters	N/A	V7400-V7777 V37000-V37777

### Examples

Example	Description
V40401	Bits 20-27 (octal) of X Input.
V41100	Timer status bits 0-17 (octal).
V7400	Stage control bit 57.
V1400.1	Bit access to V1400 bit 1.
V1400.100H	String with length 100 and HiLo byte ordering starting at V1400.
V1500.78L	String with length 78 and LoHi byte ordering starting at V1500.

### DL-260 Addressing

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

Memory Type	Reference	Data Types
I/O	X,Y,GX,GY	<b>Boolean</b>

Devices	C,SP,T,CT,S	<b>Boolean</b>
Data Words	V	Boolean, Word, <b>Short</b> , DWord, Long, LBCD, Float, String, <b>BCD*</b>

\*The default is for Timers and Counters only.

### Bit Access to V Memory

Bit information can be directly accessed within V memory registers. To access a bit within a V memory register, a bit number can be appended to any V memory address. V memory addressing with bit access would appear as follows: V<xxxxx>.<yy> where xxxxx is the V memory register location and yy is the bit number (0 to 15) within that register. If the V memory location is either a Long or DWord, the bit number yy can be (0 to 31). For more information, refer to the examples below.

### Array Support for Data Words

This driver supports array notation for V memory Data Word addresses. To specify an array, append the array size to the address specification as follows: address[array size] or address[rows][cols]. Array size is limited to 63 elements when referenced as a Word, Short, and BCD, and 31 elements when referenced as a DWord, Long, Float, and LBCD.

### Examples

- V1400 [63] @ Word - Array of 63 Words (Maximum allowed) starting at V1400.
- V1400 [31] @ DWord - Array of 31 DWords (Maximum allowed) starting at V1400.

### String Access to Data Words

This driver supports reading and writing V memory Data Words as an ASCII string. When using V memory 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 126 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.

**Note:** The references are in Octal format.

Memory Type	Discrete Memory Reference	Word Memory Reference
Input Points	X0-X1777	V40400-V40477
Output Points	Y0-Y1777	V40500-V40577
Control Relays	C0-C3777	V40600-V40777
Special Relays	SP0-SP777	V41200-V41237
Timer Status Bits	T0-T377	V41100-V41117
Timer Current Values	N/A	V0-V377
Counter Status Bits	CT0-CT377	V41140-V41157
Counter Current Values	N/A	V1000-V1377
Data Words	N/A	V400-V777 V1400-V7577 V10000-V35777
Data Words String Access HiLo Byte Ordering	N/A	V400.2H-V777.126H V1400.2H-V7577.126H V10000.2H-V35777.126H .Bit is string length, range 2 to 126 bytes.
Data Words String Access LoHi Byte Ordering	N/A	V400.2L-V777.126L V1400.2L-V7577.126L V10000.2L-V35777.126L .Bit is string length, range 2 to 126 bytes.
Remote I/O	GX0-GX3777 GY0-GY3777	V40000-V40177 V40200-V40377
Stages	S0-S1777	V41000-V41077
System Parameters	N/A	V7600-V7777 V36000-V37777

### Examples

Example	Description
V40401	Bits 20-27 (octal) of X Input.

V41100	Timer status bits 0-17 (octal).
V7600	System parameter word 7600.
V2000.1	Bit access to V2000 bit 1.
V400.100H	String with length 100 and HiLo byte ordering starting at V400.
V1400.78L	String with length 78 and LoHi byte ordering starting at V1400.

## DL-430 Addressing

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

Memory Type	Reference	Data Types
I/O	X,Y,GX	<b>Boolean</b>
Devices	C,SP,T,CT,S	<b>Boolean</b>
Data Words	V	Boolean, Word, <b>Short</b> , DWord, Long, LBCD, Float, String, <b>BCD*</b>

\*The default is for Timers and Counters only.

### Bit Access to V Memory

Bit information can be directly accessed within V memory registers. To access a bit within a V memory register, a bit number can be appended to any V memory address. V memory addressing with bit access would appear as follows: V<xxxx>.<yy> where xxxx is the V memory register location and yy is the bit number (0 to 15) within that register. If the V memory location is either a Long or DWord, the bit number yy can be (0 to 31). For more information, refer to the examples below.

### Array Support for Data Words

This driver supports array notation for V memory Data Word addresses. To specify an array, append the array size to the address specification as follows: address[array size] or address[rows][cols]. Array size is limited to 63 elements when referenced as a Word, Short, and BCD, and 31 elements when referenced as a DWord, Long, Float, and LBCD.

#### Examples

- V1400 [63] @ Word - Array of 63 Words (Maximum allowed) starting at V1400.
- V1400 [31] @ DWord - Array of 31 DWords (Maximum allowed) starting at V1400.

### String Access to Data Words

This driver supports reading and writing V memory Data Words as an ASCII string. When using V memory 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 126 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.

**Note:** The references are in Octal format.

Memory Type	Discrete Memory Reference	Word Memory Reference
Input Points	X0-X477	V40400-V40423
Output Points	Y0-Y477	V40500-V40523
Control Relays	C0-C737	V40600-V40635
Special Relays	SP0-SP137 SP320-SP617	V41200-V41205 V41215-V41230
Timer Status Bits	T0-T177	V41100-V41107
Timer Current Values	N/A	V0-V177
Counter Status Bits	CT0-CT177	V41040-V41147
Counter Current Values	N/A	V1000-V1177
Data Words	N/A	V1400-V7377
Data Words String Access HiLo Byte Ordering	N/A	V1400.2H-V7377.126H .Bit is string length, range 2 to 126 bytes.
Data Words String Access LoHi Byte Ordering	N/A	V1400.2L-V7377.126L .Bit is string length, range 2 to 126 bytes.
Remote I/O	GX0-GX737	V40000-V40037
Stages	S0-S577	V41000-V41027

System Parameters	N/A	V7400-V7777
-------------------	-----	-------------

### Examples

Example	Description
V40401	Bits 20-27 (octal) of X Input.
T172	Timer contact 172.
GX5	Remote I/O bit 5.
V2000.1	Bit access to V2000 bit 1.
V1400.100H	String with length 100 and HiLo byte ordering starting at V1400.
V1500.78L	String with length 78 and LoHi byte ordering starting at V1500.

### DL-440 Addressing

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

Memory Type	Reference	Data Types
I/O	X,Y,GX	<b>Boolean</b>
Devices	C,SP,T,CT,S	<b>Boolean</b>
Data Words	V	Boolean, Word, <b>Short</b> , DWord, Long, LBCD, Float, String, <b>BCD*</b>

\*The default is for Timers and Counters only.

### Bit Access to V Memory

Bit information can be directly accessed within V memory registers. To access a bit within a V memory register, a bit number can be appended to any V memory address. V memory addressing with bit access would appear as follows: V<xxxxx>.<yy> where xxxxx is the V memory register location and yy is the bit number (0 to 15) within that register. If the V memory location is either a Long or DWord, the bit number yy can be (0 to 31). For more information, refer to the examples below.

### Array Support for Data Words

This driver supports array notation for V memory Data Word addresses. To specify an array, append the array size to the address specification as follows: address[array size] or address[rows][cols]. Array size is limited to 63 elements when referenced as a Word, Short, and BCD, and 31 elements when referenced as a DWord, Long, Float, and LBCD.

### Examples

- V1400 [63] @ Word - Array of 63 Words (Maximum allowed) starting at V1400.
- V1400 [31] @ DWord - Array of 31 DWords (Maximum allowed) starting at V1400.

### String Access to Data Words

This driver supports reading and writing V memory Data Words as an ASCII string. When using V memory 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 126 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.

**Note:** The references are in Octal format.

Memory Type	Discrete Memory Reference	Word Memory Reference
Input Points	X0-X477	V40400-V40423
Output Points	Y0-Y477	V40500-V40523
Control Relays	C0-C1777	V40600-V40677
Special Relays	SP0-SP137 SP320-SP617 SP620-SP717	V41200-V41205 V41215-V41230 V41231-V41234
Timer Status Bits	T0-T377	V41100-V41117
Timer Current Values	N/A	V0-V377
Counter Status Bits	CT0-CT177	V41040-V41147
Counter Current Values	N/A	V1000-V1177
Data Words	N/A	V1400-V7377 V10000-V17777

Data Words String Access HiLo Byte Ordering	N/A	V1400.2H-V7377.126H V10000.2H-V17777.126H .Bit is string length, range 2 to 126 bytes.
Data Words String Access LoHi Byte Ordering	N/A	V1400.2L-V7377.126L V10000.2L-V17777.126L .Bit is string length, range 2 to 126 bytes.
Remote I/O	GX0-GX1777	V40000-V40077
Stages	S0-S1777	V41000-V41077
System Parameters	N/A	V700-V737 V7400-V7777

### Examples

Example	Description
V40401	Bits 20-27 (octal) of X Input.
V41100	Timer status bits 0-17 (octal).
V700	System parameter word 700.
V2000.1	Bit access to V2000 bit 1.
V1400.100H	String with length 100 and HiLo byte ordering starting at V1400.
V1500.78L	String with length 78 and LoHi byte ordering starting at V1500.

### DL-450 Addressing

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

Memory Type	Reference	Data Types
I/O	X,Y,GX,GY	<b>Boolean</b>
Devices	C,SP,T,CT,S	<b>Boolean</b>
Data Words	V	Boolean, Word, <b>Short</b> , DWord, Long, LBCD, Float, String, <b>BCD*</b>

\*The default is for Timers and Counters only.

#### Bit Access to V Memory

Bit information can be directly accessed within V memory registers. To access a bit within a V memory register, a bit number can be appended to any V memory address. V memory addressing with bit access would appear as follows: V<xxxxx>.<yy> where xxxxx is the V memory register location and yy is the bit number (0 to 15) within that register. If the V memory location is either a Long or DWord, the bit number yy can be (0 to 31). For more information, refer to the examples below.

#### Array Support for Data Words

This driver supports array notation for V memory Data Word addresses. To specify an array, append the array size to the address specification as follows: address[array size] or address[rows][cols]. Array size is limited to 63 elements when referenced as a Word, Short, and BCD, and 31 elements when referenced as a DWord, Long, Float, and LBCD.

#### Examples

- V1400 [63] @ Word - Array of 63 Words (Maximum allowed) starting at V1400.
- V1400 [31] @ DWord - Array of 31 DWords (Maximum allowed) starting at V1400.

#### String Access to Data Words

This driver supports reading and writing V memory Data Words as an ASCII string. When using V memory 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 126 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.

**Note:** The references are in Octal format.

Memory Type	Discrete Memory Reference	Word Memory Reference
Input Points	X0-X1777	V40400-V40477
Output Points	Y0-Y1777	V40500-V40577

Control Relays	C0-C3777	V40600-V40777
Special Relays	SP0-SP137 SP320-SP717	V41200-V41237
Timer Status Bits	T0-T377	V41100-V41117
Timer Current Values	N/A	V0-V377
Counter Status Bits	CT0-CT377	V41040-V41157
Counter Current Values	N/A	V1000-V1377
Data Words	N/A	V1400-V7377 V10000-V37777
Data Words String Access HiLo Byte Ordering	N/A	V1400.2H-V7377.126H V10000.2H-V37777.126H .Bit is string length, range 2 to 126 bytes.
Data Words String Access LoHi Byte Ordering	N/A	V1400.2L-V7377.126L V10000.2L-V37777.126L .Bit is string length, range 2 to 126 bytes.
Remote I/O	GX0-GX3777 GY0-GY3777	V40000-V40177 V40200-V40377
System Parameters	N/A	V700-V737 V7400-V7777

### Examples

Example	Description
V40401	Bits 20-27 (octal) of X Input.
V41100	Timer status bits 0-17 (octal).
V700	System parameter word 700.
V2000.1	Bit access to V2000 bit 1.
V1400.100H	String with length 100 and HiLo byte ordering starting at V1400.
V1500.78L	String with length 78 and LoHi byte ordering starting at V1500.

## 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

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

[Unable to write to '<address>' on device '<device name>'](#)

### Device Specific Messages

[Winsock initialization failed \(OS Error = n\)](#)

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

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

### Automatic Tag Database Generation Messages

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

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

## Address Validation

---

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](#)

### 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 address '<address>' contains a syntax error

---

#### Error Type:

Warning

#### Possible Cause:

1. A tag address contains one or more invalid characters.
2. The Bit addressing notation conflicts with the assigned data type.

#### Solution:

Re-enter the address in the client application.

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

---

#### Error Type:

Warning

**Possible Cause:**

A tag address that has been specified dynamically references a location that is beyond the range of supported locations for the device.

**Solution:**

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

---

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

---

**Error Type:**

Warning

**Possible Cause:**

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

**Solution:**

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

---

**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>' 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.

---

**Device Status Messages**

---

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

**Device Status Messages**

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

[Unable to write to '<address>' on device '<device name>'](#)

---

**Device '<device name>' not responding**

---

**Error Type:**

Serious

**Possible Cause:**

1. The Ethernet connection between the device and the Host PC is broken.
2. The named device may have been assigned an incorrect IP address.
3. The requested address is not available in the device.
4. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.



**Solution:**

1. Verify the cabling between the PC and the ECOM device network.
2. Verify that the IP address given to the named device matches that of the actual device.
3. Verify that the device supports the requested address.
4. Increase the Request Timeout setting so that the entire response can be handled.

**Unable to write to '<address>' on device '<device name>'****Error Type:**

Serious

**Possible Cause:**

1. The serial connection between the device and the Host PC is broken.
2. The communications parameters for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.

**Solution:**

1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.

**Device Specific Messages**

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

**Device Specific Messages**

[Winsock initialization failed \(OS Error = n\)](#)

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

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

**Winsock initialization failed (OS Error = n)****Error Type:**

Fatal

OS Error	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 AutomationDirect ECOM 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.

**Bad address in block [<start address> to <end address>] on device '<device name>'****Error Type:**

Serious

**Possible Cause:**

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

**Solution:**

Verify the tags assigned to addresses in the specified range on the device. Eliminate those that reference invalid locations.

### **Automatic Tag Database Generation Messages**

---

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

#### **Automatic Tag Database Generation Messages**

[Unable to generate a tag database for device <device name>. Reason: Low memory resources](#)  
[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: Low memory resources**

---

**Error Type:**

Warning

**Possible Cause:**

Memory required for database generation could not be allocated. The process is aborted.

**Solution:**

Close any unused applications and/or increase the amount of virtual memory and try again.

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

---

**Error Type:**

Warning

**Possible Cause:**

The file specified as the Tag Import File in the Database Settings tab in Device Properties is an improperly formatted txt or csv file.

**Solution:**

If importing Element Documentation, verify that the export file was saved in "Standard Format" with a .csv extension. If problem resumes, try re-exporting the file.

**See Also:**

[Importing DirectSoft Elements](#)

# Index

## A

Address '<address>' is out of range for the specified device or register.....	23
Address Descriptions.....	13
Address Validation.....	23
Automatic Tag Database Generation.....	7, 26

## B

Bad address in block [<start address> to <end address>] on device '<device name>'.....	25
BCD.....	12
Boolean.....	12

## C

Communication Parameters.....	6
-------------------------------	---

## D

Data Type '<type>' is not valid for device address '<address>'.....	24
Data Types Description.....	12
Device '<device name>' not responding.....	24
Device address '<address>' contains a syntax error.....	23
Device address '<address>' is not supported by model '<model name>'.....	24
Device address '<address>' is Read Only.....	24
Device ID.....	5
Device Setup.....	5
Device Specific Messages.....	25
Device Status Messages.....	24
DirectSoft Steps: Import Preparation.....	7
DL-05 Addressing.....	13
DL-06 Addressing.....	14
DL-240 Addressing.....	15
DL-250(-1) Addressing.....	16

DL-260 Addressing .....	17
DL-430 Addressing .....	19
DL-440 Addressing .....	20
DL-450 Addressing .....	21
DWord .....	12

## E

ECOM .....	5
Element Documentation .....	6
Error Descriptions .....	23

## F

Float .....	12
-------------	----

## I

Import File-To-Server Name Conversions .....	7
Importing DirectSoft Elements .....	7
IP Address .....	5

## L

Long .....	12
------------	----

## M

Missing address .....	23
-----------------------	----

## N

Network .....	5
---------------	---

**O**

OPC Server Steps: Import Preparation .....	10
Optimizing Your AutomationDirect ECOM Communications .....	11
Overview .....	4

**S**

Short .....	12
-------------	----

**T**

Tag Hierarchy .....	7
Tag Import Settings .....	6

**U**

Unable to generate a tag database for device <device name>. Reason: Low memory resources .....	26
Unable to generate a tag database for device <device name>. Reason: Import file is invalid or corrupt .....	26
Unable to write tag '<address>' on device '<device name>' .....	25

**W**

Winsock initialization failed (OS Error = n) .....	25
Winsock V1.1 or higher must be installed to use the AutomationDirect ECOM device driver .....	25
Word .....	12