Omron FINS Ethernet Driver Help

Table of Contents

Omron FINS Ethernet Driver Help

Help version 1.045

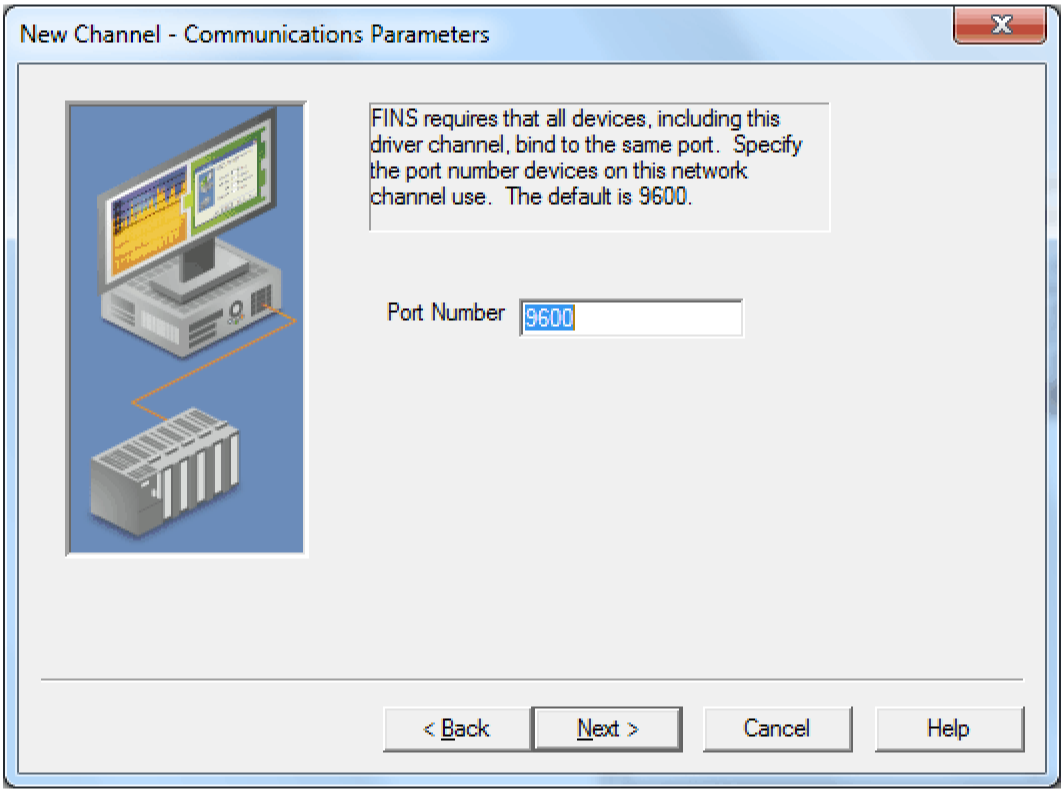
CONTENTS

Overview

The Omron FINS Ethernet Driver provides an easy and reliable way to connect Omron FINS Ethernet controller to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications.

Channel Setup

Communications Parameters



FINS requires that all devices, including this driver channel, bind to the same port. Specify the port number devices on this network channel use. The default is 9600.

Port Number 9600

Descriptions of the parameters are as follows:

- **Port Number:** This parameter specifies the port number that will be used by the devices on the local Ethernet network. FINS requires that the source and destination port number be the same. The valid range is

1 to 65535. The default setting is 9600.

Optimizing Communication Performance

Users can create multiple channels in the OPC server that utilize a unique port number in order to optimize communication performance. Examples are as follows:

- Channel1.device1 will talk on Port 9601.
- Channel2.device2 will talk on Port 9602.
- CX programmer will talk on Port 9600.

Note: Communication to multiple physical devices (with different port numbers configured in the devices) from the same channel in the OPC server is not allowed. This is because both the port number and adapter are configured at the channel level.

Important: In order to communicate to the same device (or multiple physical devices with the same port number configured in the devices) across multiple OPC server channels, the same port number must be used for each channel. Each channel, however, must be configured to use a unique network adapter and IP. For more information, refer to **Multihoming**.

Device Setup

Supported Devices

This driver supports the FINS protocol via UDP/IP. For a list of models that support the FINS Communications Service, refer to the manufacturer's web site.

Communication Protocol

Omron FINS using UDP/IP.

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

Device IDs

The Device ID specifies the IP address of the controller that is in direct contact with the host computer. This controller may not necessarily be the destination device, but instead be a relay device. The destination device is specified by its FINS network parameters. For all parameters specified in Device Properties (except the Device ID when a relay device is used) refer to the destination device.

Request Size

This parameter specifies the number of bytes that may be requested from a device at one time. To refine the driver's performance, the request size may be configured to one of the following settings: 32, 64, 128, 256, 512, 1024, or 1984 bytes. The default setting is 512 bytes.

Source Network Address Number

This parameter specifies the network address number of the source device (which is the computer running this driver). The Source Network Address Number is sometimes referred to as SNA in Omron FINS documentation. The valid range is 0 to 127. Actual network numbers can range from 1 to 127. FINS interprets 0 as use the local network; as such, 0 must not be used when data will be routed through gateway PLCs.

Source Node Number

This parameter specifies the node number of the source device. The Source Node Number is sometimes referred to as SA1 in Omron FINS documentation. The valid range is 0 to 254. If the destination PLC is configured to use automatic address generation, then this number must be the host number portion of the host computer's IP address.

For example, if the host computer has an IP of 111.222.333.123 and the subnet mask is 255.255.255.000, the source node number should be 123. If the target PLC is configured to use an address table, then the table must have an entry for the host computer's IP. The node number in this table entry must agree with the source node number entered for the driver.

Note: When a channel is duplicated using the Copy and Paste commands, the new channel will retain the original channel's FINS network configuration parameters. Unless the Source Node Number is changed to a unique value, however, correct data transmission cannot be guaranteed.

Destination Network Address Number

This parameter specifies the network address number of the destination device. The Destination Network Address Number is sometimes referred to as DNA in Omron FINS documentation. The valid range is 0 to 127. Actual network numbers can range from 1 to 127. FINS interprets 0 as use the local network; as such, 0 must not be used when data will be routed through gateway PLCs.

Destination Node Number

This parameter specifies the node number of the destination device. The Destination Node Number is sometimes referred to as DA1 in Omron FINS documentation. The valid range is 0 to 254.

Note: The source unit number SA2 is assumed to be 0.

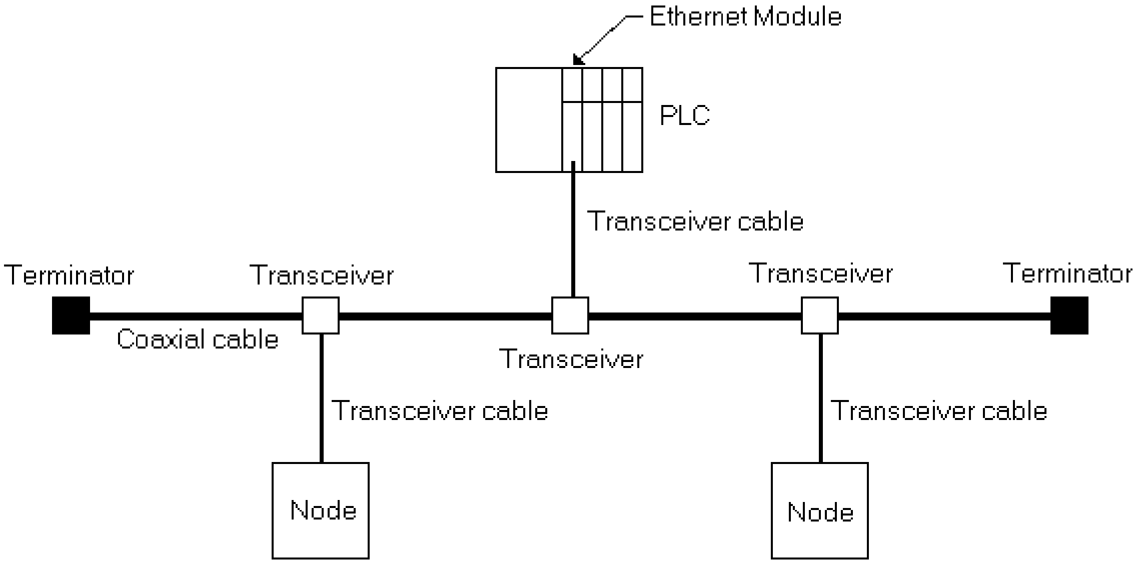
Caution: Pay particular attention to the FINS network and node numbers configured for each device, as well as the IP address calculation method used by the target PLCs. Inconsistent values will result in communications problems. For example, when using the default Source and Destination Network Address of 0 (Local network), the Destination Node Number and Source Node Number cannot be the same.

Destination Unit Number

This parameter specifies the device unit number, which is sometimes referred to as DA2. The valid range is 0 to 255. The default number for DA2 is 0.

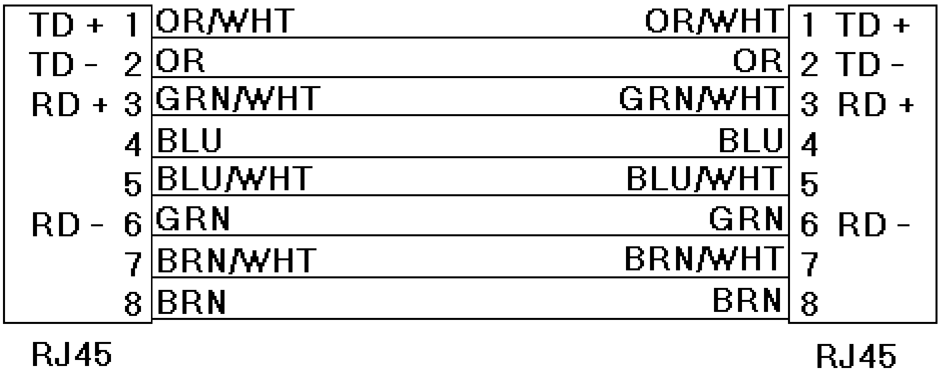
Network Configuration

For optimum reliability, care should be taken to properly isolate all cables and components from electrical noise. Coaxial cables should have lengths of integral multiples of 2.5 meters up to a maximum of 500 meters. Depending on the transceiver used, 10BASE-T twisted-pair cable can be substituted for 10BASE5 coax.



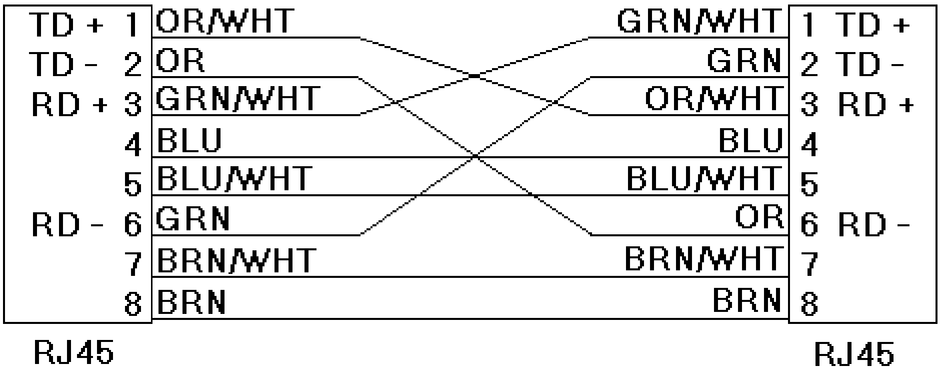
10BASE-T Cable (if used)

Patch Cable (Straight Through)

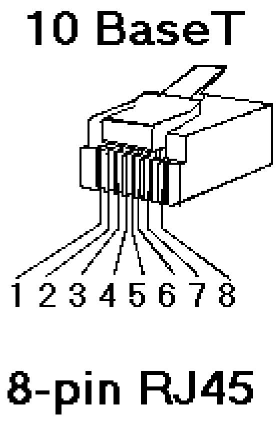


TD +	1	OR/WHT	OR/WHT	1	TD +
TD -	2	OR	OR	2	TD -
RD +	3	GRN/WHT	GRN/WHT	3	RD +
	4	BLU	BLU	4	
	5	BLU/WHT	BLU/WHT	5	
RD -	6	GRN	GRN	6	RD -
	7	BRN/WHT	BRN/WHT	7	
	8	BRN	BRN	8	
RJ45					RJ45

Crossover Cable

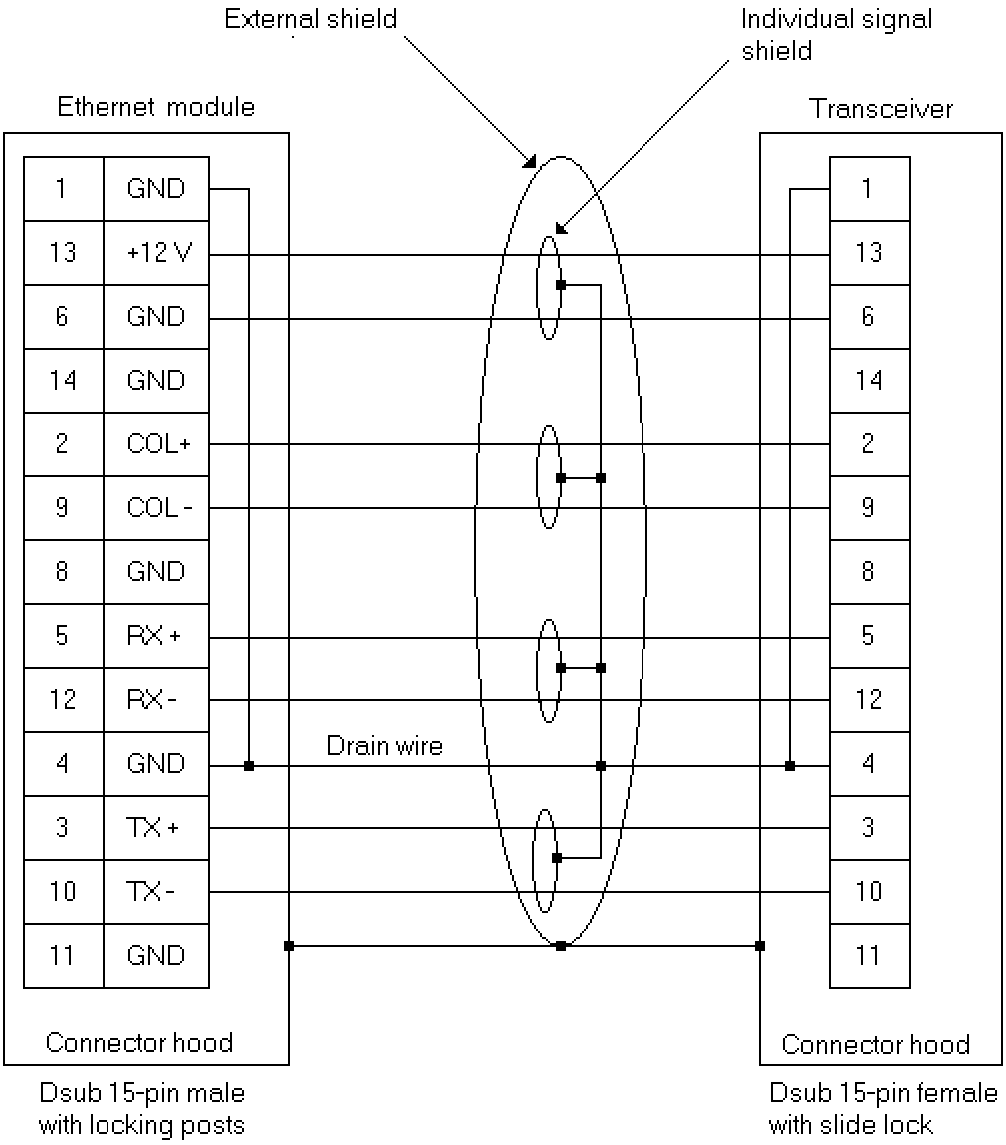


TD +	1	OR/WHT	GRN/WHT	1	TD +
TD -	2	OR	GRN	2	TD -
RD +	3	GRN/WHT	OR/WHT	3	RD +
	4	BLU	BLU	4	
	5	BLU/WHT	BLU/WHT	5	
RD -	6	GRN	OR	6	RD -
	7	BRN/WHT	BRN/WHT	7	
	8	BRN	BRN	8	
RJ45					RJ45



Transceiver Cable

All PLC to transceiver cabling should conform to the IEEE802.3 standards, where each of the three signal wire pairs are individually shielded in addition to an external shield.



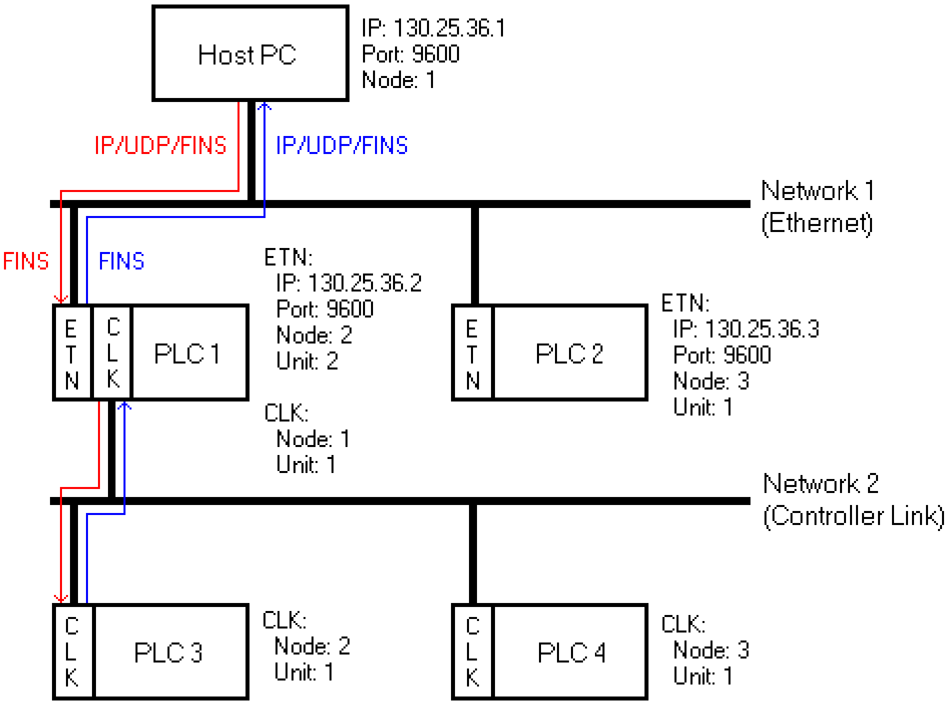
Note: For more information, refer to **FINS Networks**.

FINS Networks

The FINS communications service was developed by Omron to provide a consistent way for PLCs and computers on various networks to communicate. Compatible network types include Ethernet, Host Link, Controller Link, SYSMAC LINK, SYSMAC WAY, and Toolbus. FINS allows communications between nodes up to three network levels. A direct connection between a computer and a PLC via Host Link is not considered a network level.

The diagram below shows a FINS network comprised of interconnected Ethernet and Controller Link networks, and will serve as an example for the following discussion. FINS allows communication between any pair of devices in the diagram. PLC 1 acts as a gateway between the two networks. The Host PC sends a data request command (shown in red) to PLC 3 via PLC 1. The response is shown in blue.

Note: In principle, PLC 3 or 4 could act as a gateway to a third network layer that could also be reached by the Host PC.



FINS Messages

FINS messages contain two parts: a header and a data portion. The header contains source and destination information, among other things. The data portion contains command codes and optional command parameters. The six source and destination parameters contained in the header are as follows:

- **DNA:** Destination Network Address.
- **DA1:** Destination Node Number.
- **DA2:** Destination Module Address.
- **SNA:** Source Network Address.
- **SA1:** Source Node Number.
- **SA2:** Source Module Address.

This driver will always set DA2 and SA2 to zero; meaning, communication will be between the host computer and the destination node's CPU module.

Ethernet Communications

This driver is able to communicate with any FINS compatible device on an Ethernet network. The FINS device can process a given FINS command if it is the destination node or relay the message to another device if it is not. In the figure above, it is desired to send a data request from the Host PC to PLC 3. The driver constructs a FINS message with appropriate source and destination parameters set in the header, encapsulates that message in a UDP/IP datagram, and then sends it off to PLC 1. The message cannot be sent directly to PLC 3 because it is not on the same network as the PC. PLC 1 examines the FINS message header and determines that node 1 on network 2 (which is PLC 3) is the intended destination. PLC 1 relays the FINS message to PLC 3. Since PLC 3 is not on an Ethernet network, the UDP/IP wrapper is removed. PLC 3 then sends its reply back to the Host PC via PLC 1.

A problem arises when PLC 1 must forward the reply from PLC 3 to the Host PC. The reply from PLC 3 addresses the destination using FINS network parameters only: it does not explicitly give the destination IP address. Omron has devised three methods for a PLC to determine destination IPs from FINS network parameters. The method is chosen when the PLC is configured and will determine what node numbers should be assigned to each device on the Ethernet network. The three methods are Automatic Address Generation, IP Address Table, and Combined Address Conversion. Descriptions of the methods are as follows:

- **Automatic Address Generation:** This method uses the relay device to construct the destination IP from its own IP, the subnet mask, and the FINS destination node number. IP addresses are made up of two parts: the network number (which is the same for all nodes on the local network) and the host number (which is unique to each node on the local network). The Automatic Address Generation method requires that the host number portion of each node's IP be the same as its FINS node number.

Note: In this example, PLC 1 would calculate the network number by first performing a logical AND operation between its own IP address and subnet mask. In Boolean algebra terms, 130.25.36.2 AND 255.255.255.0 equals 130.25.36.0. The destination IP would then be calculated by adding the FINS destination node number to the network number. For the Destination IP, 130.25.36.0 + 1 = 130.25.36.1.

- **IP Address Table:** This method uses an IP address table, which is simply a list of FINS node numbers and their associated IP addresses. In this method, the host number portion of the IP address does not need to be the same as the FINS node number. If used, the IP address tables must be programmed into the PLCs on the Ethernet networks.
- **Combined Address Conversion:** This method uses a device to look up the FINS destination node number in its IP address table. If the node number is found, the corresponding IP address will be taken from the table and used to construct the UDP/IP datagram. If the node number is not found, the destination IP will be computed using the Automatic Address Generation method.

Note: PLC 1 also needs to know on what port the Host PC is listening. Since FINS requires that all nodes on an Ethernet network listen on the same port number, the PLC will simply use the port number on which it has been configured to listen.

Server Configuration Example

1. To start, create a channel that uses this driver to communicate with devices on the Ethernet network. FINS requires that all nodes on an Ethernet network use the same port number to send and receive data. For this example, use the default value 9600.
2. Next, create a device on that channel that represents the destination node. For this example, use PLC 3.
3. Set the Device ID, which must be the IP address of the PLC with which the driver will be in direct communication. For this example, use PLC 1 (130.25.36.2).

   Note: The Device ID should be the IP address of the destination device if that device is on the Host PC's local Ethernet network. This would be the case with PLC 2.
4. Next, set the FINS network parameters to identify the source (Host PC) and destination (PLC 3). In this example, the settings are as follows:

   - **Source Network Address Number (SNA):** 1.*
   - **Source Node Number (SA1):** 1.
   - **Destination Network Address Number (DNA):** 2.*
   - **Destination Node Number (DA1):** 1.

   *The default value for network addresses is zero. This is interpreted by FINS as to use the local network. Actual network numbers can range from 1 to 127. Using zero is convenient if there is only one network level. When using gateway devices, however, users must specify the actual network number (1–127) to avoid routing ambiguities.

Note 1: SA2 and DA2 are automatically set to zero by the driver.

Note 2: Similar device objects would need to be created for PLC 1, 2, and 4.

See Also: Channel Setup

Routing Tables

With multi-level networks, additional information must be programmed into the PLCs so they can send messages to other nodes in the system. This is accomplished using FINS routing tables, which come in two types: Local and Remote. Local routing tables associate a network number with one of the communication modules or Special Input/Output Units (SIOU) installed in the PLC's rack. Remote routing tables give directions on how to reach one of the next network levels. The routing tables for the given example would appear as shown below.

PLC 1 (Local)

Network Number	Unit
1	2
2	1

There are two local networks for PLC 1. The Ethernet and Controller Link network numbers are assigned as 1 and 2 respectively. The Ethernet and Controller Link modules are assigned unit numbers 2 and 1 respectively. The network node number of each of these modules is configured by the user and must be unique within its associated network.

Note: Both of the communications modules in PLC 1 node 10 could have been called since they are on different networks. There are no remote networks for PLC 1.

PLC 2 (Local)

Network Number	Unit
1	1

PLC 2 has only one communications module: the Ethernet module. Thus, it only has one entry in its local routing table.

PLC 2 (Remote)

Remote Network Number	Relay Network	Relay Node
2	1	2

Network 2 is a remote network for PLC 2. To send a message to a node on network 2, PLC 2 must send that message to a relay node (the gateway) on one of its local networks. This local network is called the relay network and must be network 1 in the given example. The Ethernet module in the gateway (PLC 1) is node number 2. Therefore, the relay node for PLC 2 is 2.

Likewise, routing tables for PLC 3 and 4 must appear like the following. They happen to be the same in the given example because the Controller Link modules in both PLCs are unit number 1. These modules must be assigned node numbers that are unique in network 2.

PLC 3 and 4 (Local)

Network Number	Unit
2	1

PLC 3 and 4 (Remote)

Remote Network Number	Relay Network	Relay Node
1	2	1

Duplicate Devices

It is generally recommended that the communications load be distributed over multiple channels. Significant performance gains can be achieved since the Windows operation system allows each channel to operate on an independent thread. The idea is to have as few devices on each channel as possible, so that no one device can significantly impact the update rate of the others. For more information, refer to **Optimizing Your Omron FINS Ethernet Communications**.

Although it is not usually advantageous to create multiple server device objects that communicate with a single physical device, it is necessary to do so when using a multi-level FINS network. In the example above, device objects are needed for PLC 3 and PLC 4. The driver will actually be in direct communications with PLC 1. This means that the device number configured for both device objects must be the IP address of PLC 1. With most Ethernet drivers, this sort of thing is perfectly safe. Because FINS requires that all nodes on an Ethernet network use the same port number to send and receive data, however, a problem arises.

If the device objects for PLC 3 and 4 were created on the same channel, everything will work fine. Users will encounter the problem when attempting to improve performance by placing the device objects on separate channels. It is important to know that this driver will give each one of these channels an Ethernet communications socket. For example, if the Host PC has only one IP associated with it, then both of these sockets must use that IP as its source address and both of these sockets will need to use the same port number according to the FINS protocol. If the driver allowed users to do this, the UDP/IP datagrams from PLC 1 would contain the same source and destination address information (where source is 130.25.36.2/9600 and destination is 130.25.36.1/9600) regardless of the intended destination device object. Users cannot be guaranteed that the operating system will send the datagrams to the correct socket if both happen to be waiting for responses from PLC 3 and 4. The only way to remove the ambiguity is to associate each channel/socket with a unique IP address.

In order to associate a unique IP address with each channel, users must multihome the host computer. This means that the computer must have multiple NICs installed or users must associate multiple IP addresses with a single NIC. Things will work if both devices are on the same channel because all transactions are carried out serially on a given channel. It doesn't matter if the source and destination addresses (IP/port) in responses from PLC 3 and 4 look the same because the socket will be waiting for a response from only one PLC at a time. This is only relevant when multi-level FINS networks are being used.

Caution: Pay particular attention to the FINS network and node numbers configured for each device, as well as the IP address calculation method being used by the target PLCs. Inconsistent values will result in communications problems.

Note: For more information on FINS networking, refer to Omron documentation.

Multihoming

Some applications require users to associate a unique IP address with each channel. In these cases, the host computer must be multihomed; meaning, it must be configured to have more than one IP address. This may be accomplished by installing multiple Network Adapter Cards (NIC) in the computer or by assigning multiple IP addresses to a single NIC.

Adding IP Addresses to a Single NIC on Windows NT

1. To start, click **My Computer** | **Control Panel** | **Network**.
2. Click the **Protocols** tab and then select **TCP/IP Protocol**.
3. Next, click **Properties** and then select the **IP Address** tab.
4. Click **Advanced** | **Add**.
5. Enter the additional IP address and subnet mask.
6. Click **OK**.

Adding IP Addresses to a Single NIC on Windows 2000, XP, and 2003

1. To start, click **My Computer** | **Control Panel** | **Network and Dial-Up Connections**.
2. Click **Local Area Connection** (or any other icon associated with the NIC of interest).
3. Next, click **Properties** and then select **Internet Protocol (TCP/IP)**.
4. Click **Properties** | **Advanced**.
5. Select the **IP Settings** tab and then click **Add**.
6. Enter the additional IP address and subnet mask.
7. Click **OK**.

Adding IP Addresses to a Single NIC on Windows Vista, 2008, and 7

1. Click **Start** and then open **Network Connections**.
2. Next, click **Control Panel** | **Network and Internet**. Then, select the connection that will be changed (such as the Local Area Connection).
3. Click **Properties**, and then provide the administrator password or confirmation (if prompted).
4. Select the **Networking** tab. Beneath **This connection uses the following items**, click **Internet Protocol Version 4 (TCP/IPv4)**.
5. Next, click **Properties**. Ensure that the connection is set to use an IP address by clicking **Use the following IP address**.
6. Next, specify the IP address settings in the **IP Address**, **Subnet Mask**, and **Default Gateway** fields.
7. To add a second IP address, click **Advanced** | **IP Settings**. Beneath **IP Address**, click **Add**. Then, enter a new IP address and subnet mask.

Note 1: Windows NT can add up to five IP addresses for each NIC via the control panel. More IP addresses can be added to the registry manually as necessary. To browse, look under **HEKY_LOCAL_MACHINE** | **SYSTEM** | **CurrentControlSet** | **Services**. Then, select the service associated with the adapter card in question. Under the service, go to the **Parameters** | **Tcpip** subkey. Add the IP addresses to **IPAddress** and then edit

SubnetMask and add an entry for each new IP address. Windows 2000 does not impose a limit on the number of IP addresses users add via the Control Panel, although the system may need to be rebooted before the new IPs can be used.

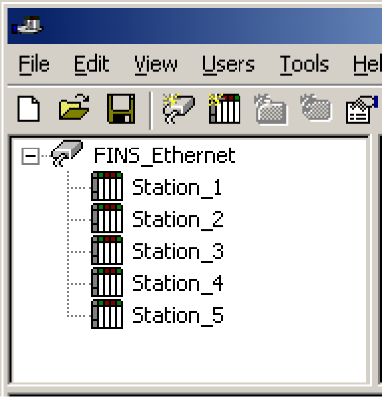
Note 2: There will be additional operating system overhead when running on a multihomed system. Unless users are using very fast devices, this overhead should not entirely cancel out the performance gain achieved from distributing the communications load over multiple channels.

See Also: Optimizing Your Omron FINS Ethernet Communications

Optimizing Your Omron FINS Ethernet Communications

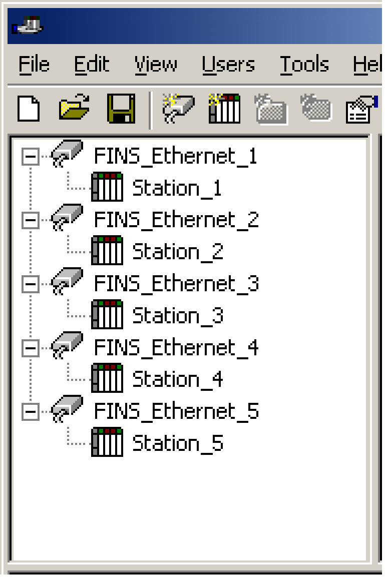
The Omron FINS Ethernet Driver has been designed to provide the best performance with the least amount of impact on the system's overall performance. While the Omron FINS Ethernet Driver is fast, there are a couple of guidelines that can be used in order to control and optimize the application and gain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

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



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

The performance will improve even if the application has more than 32 devices. While 32 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.

The Device Request Size parameter can also affect the Omron FINS Ethernet Driver's performance. The request size refers to the number of bytes that may be requested from a device at one time. If data is being read from a large number of contiguous addresses, it may be advantageous to use a large request size. To refine the performance of this driver, the request size may be configured from 32 to 512 bytes for each device.

Important: If using a multi-leveled FINS network, multiple device objects may need to be created in the server that will all be in direct communication with the same gateway PLC. Since FINS requires all nodes on an Ethernet network to use the same port number to send and receive data, this driver must impose some constraints that become relevant when distributing the communications load over multiple channels. For more information, refer to **Channel Setup**.

Data Types Description

Data Type	Description
Boolean	Single bit
Short	Signed 16 bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
Word	Unsigned 16 bit value bit 0 is the low bit bit 15 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
DWord	Unsigned 32 bit value bit 0 is the low bit bit 31 is the high bit
Float	32-bit real
BCD	Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.
LBCD	Four byte packed BCD Value range is 0-99999999. Behavior is undefined for values beyond this range.
String	Null terminated ASCII string. Support includes string lengths up to 512 characters, and selection of HiLo byte order, LoHi byte order, Only High byte and Only Low byte.

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.

C200H Addressing
C500 Addressing
C1000H Addressing
C2000H Addressing
CV500 Addressing
CV1000 Addressing
CV2000 Addressing
CVM1-CPU01 Addressing
CVM1-CPU11 Addressing
CVM1-CPU21 Addressing
CS1 Addressing
CJ1 Addressing
CJ2 Addressing

C200H Addressing

The default data types for dynamically defined tags are shown in **bold**. For more information, refer below the table for notes and restrictions on **BCD Support**, **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Auxiliary Relay	AR00-AR27 AR00-AR26 ARxx.00-ARxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Auxiliary Relay as string with HiLo byte order	AR00.056H-AR27.002H .l is string length, range 2 to 56 chars	**String**	Read/Write
Auxiliary Relay as string with LoHi byte order	AR00.056L-AR27.002L .l is string length, range 2 to 56 chars	**String**	Read/Write
Auxiliary Relay as string Using Only the High Order byte of each word	AR00.028D-AR27.001D .l is string length, range 1 to 28 chars	**String**	Read/Write
Auxiliary Relay as string Using Only the Low Order byte of each word	AR00.028E-AR27.001E .l is string length, range 1 to 28 chars	**String**	Read/Write
Data Memory	DM0000-DM6655 DM0000-DM6654 DMxxxx.00-DMxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Data Memory as string with HiLo byte order	DM0000.512H-DM6655.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string with LoHi byte order	DM0000.512L-DM6655.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string Using Only the High Order byte of each word	DM0000.256D-DM6655.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory as string Using Only the Low Order byte of each word	DM0000.256E-DM6655.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory (current bank)	EM0000-EM6143 EM0000-EM6142 EMxxxx.00-EMxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	EM0000.512H-EM6143.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string	EM0000.512L-EM6143.002L	**String**	Read/Write

with LoHi byte order	.l is string length, range 2 to 512 chars		
Expansion Data Memory (current bank) as string Using Only the High Order byte of each word	EM0000.256D-EM6143.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string Using Only the Low Order byte of each word	EM0000.256E-EM6143.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory	EM00:0000-EM07:6143 EM00:0000-EM07:6142 EMx:x.00-EMxx:xxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory as string with HiLo byte order	EM00:0000.512H-EM07:6143.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory as string with LoHi byte order	EM00:0000.512L-EM07:6143.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory as string Using Only the High Order byte of each word	EM00:0000.256D-EM07:6143.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory as string Using Only the Low Order byte of each word	EM00:0000.256E-EM07:6143.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Holding Relay	HR00-HR99 HR00-HR98 HRxx.00-HRxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Holding Relay as string with HiLo byte order	HR00.200H-HR99.002H .l is string length, range 2 to 200 chars	**String**	Read/Write
Holding Relay as string with LoHi byte order	HR00.200L-HR99.002L .l is string length, range 2 to 200 chars	**String**	Read/Write
Holding Relay as string Using Only the High Order byte of each word	HR00.100D-HR99.001D .l is string length, range 1 to 100 chars	**String**	Read/Write
Holding Relay as string Using Only the Low Order byte of each word	HR00.100E-HR99.001E .l is string length, range 1 to 100 chars	**String**	Read/Write
Internal Relay	IR000-IR511 IR000-IR510 IRxxx.00-IRxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Internal Relay as a string with HiLo byte order	IR000.512H-IR511.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Internal Relay as a string with LoHi byte order	IR000.512L-IR511.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Internal Relay as a string Using Only the High Order byte of each word	IR000.256D-IR511.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Internal Relay as a string Using Only the Low Order byte of each word	IR000.256E-IR511.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Link Relays	LR00-LR63 LR00-LR62 LRxx.00-LRxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Link Relay as string with HiLo byte order	LR00.128H-LR63.002H .l is string length, range 2 to 128 chars	**String**	Read/Write
Link Relay as string with LoHI byte order	LR00.128L-LR63.002L	**String**	Read/Write

	.l is string length, range 2 to 128 chars		
Link Relay as string Using Only the High Order byte of each word	LR00.064D-LR63.001D .l is string length, range 1 to 64 chars	**String**	Read/Write
Link Relay as string Using Only the Low Order byte of each word	LR00.064E-LR63.001E .l is string length, range 1 to 64 chars	**String**	Read/Write
Temporary Relay	TR TR0-TR7	**Word**, Short **Boolean**	Read/Write
Timer/Counter	TC000-TC511	**BCD**, Word, Short	Read/Write
Timer/Counter as string with HiLo byte order	TC000.512H-TC511.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer/Counter as string with LoHi byte order	TC000.512L-TC511.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer/Counter as string Using Only the High Order byte of each word	TC000.256D-TC511.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer/Counter as string Using Only the Low Order byte of each word	TC000.256E-TC511.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer/Counter Status	TS000-TS511	**Boolean**	Read/Write

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range is as follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range is as follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example
IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The C200H model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) 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 512 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 512. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register, the length of the string can be from 1 to 256 characters and is entered in place of a bit number. The string length times 2 cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 256. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address. For more information, refer to **Device Setup**.

Examples
1. To address a string starting at DM1000 with a length of 100 bytes and HiLo byte order, enter:
DM1000.100H

2. To address a string starting at DM1100 with a length of 78 bytes and LoHi byte order, enter:
DM1100.078L

3. To address a string starting at DM2000 with a length of 55 bytes and Only the High Order byte, enter:
DM2000.055D

4. To address a string starting at DM2200 with a length of 37 bytes and Only the Low Order byte, enter:
DM2200.037E

Array Support

Arrays are supported for all data types except Boolean. There are two methods of addressing an array. Examples are given using data memory locations.

DMxxxx [rows] [cols]
*DMxxxx [cols]**

*This method assumes that "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes (2 for Word, Short and BCD; 4 for DWord, Long, LBCD and Float) cannot exceed the request size that has been assigned to the device. For example, a 10 X 10 array of words results in an array size of 200 bytes, which would require a request size of at least 256.

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD and Float). Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Thus, writing to DM0 will also modify the value held in DM1. It is recommended that users utilize these data types so that overlapping does not occur. For example, when using DWords, users may want to use DM0, DM2, DM4 and so on to prevent overlapping Words.

C500 Addressing

The default data types for dynamically defined tags are shown in **bold**. For more information, refer below the table for notes and restrictions on **BCD Support**, **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Data Memory	DM000-DM511 DM000-DM510 DMxxx.00-DMxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Data Memory as string with HiLo byte order	DM000.512H-DM511.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string with LoHi byte order	DM000.512L-DM511.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string Using Only the High Order byte of each word	DM000.256D-DM511.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory as string Using Only the Low Order byte of each word	DM000.256E-DM511.001E .l is string length, range 1 to 256	**String**	Read/Write

	chars		
Holding Relay	HR00-HR31 HR00-HR30 HRxx.00-HRxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Holding Relay as string with HiLo byte order	HR00.064H-HR31.002H .l is string length, range 2 to 64 chars	**String**	Read/Write
Holding Relay as string with LoHi byte order	HR00.064L-HR31.002L .l is string length, range 2 to 64 chars	**String**	Read/Write
Holding Relay as string Using Only the High Order byte of each word	HR00.032D-HR31.001D .l is string length, range 1 to 32 chars	**String**	Read/Write
Holding Relay as string Using Only the Low Order byte of each word	HR00.032E-HR31.001E .l is string length, range 1 to 32 chars	**String**	Read/Write
Internal Relay	IR00-IR63 IR00-IR62 IRxx.00-IRxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Internal Relay as a string with HiLo byte order	IR00.128H-IR63.002H .l is string length, range 2 to 128 chars	**String**	Read/Write
Internal Relay as a string with LoHi byte order	IR00.128L-IR63.002L .l is string length, range 2 to 128 chars	**String**	Read/Write
Internal Relay as a string Using Only the High Order byte of each word	IR00.064D-IR63.001D .l is string length, range 1 to 64 chars	**String**	Read/Write
Internal Relay as a string Using Only the Low Order byte of each word	IR00.064E-IR63.001E .l is string length, range 1 to 64 chars	**String**	Read/Write
Link Relays	LR00-LR31 LR00-LR30 LRxx.00-LRxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Link Relays as string with HiLo byte order	LR00.064H-LR31.002H .l is string length, range 2 to 64 chars	**String**	Read/Write
Link Relays as string with LoHi byte order	LR00.064L-LR31.002L .l is string length, range 2 to 64 chars	**String**	Read/Write
Link Relays as string Using Only the High Order byte of each word	LR00.032D-LR31.001D .l is string length, range 1 to 32 chars	**String**	Read/Write
Link Relays as string Using Only the Low Order byte of each word	LR00.032E-LR31.001E .l is string length, range 1 to 32 chars	**String**	Read/Write
Temporary Relay	TR TR0-TR7	**Word**, Short **Boolean**	Read/Write
Timer/Counter	TC000-TC127	**BCD**, Word, Short	Read/Write
Timer/Counter as string	TC000.256H-TC127.002H	**String**	Read/Write

with HiLo byte order	.l is string length, range 2 to 256 chars		
Timer/Counter as string with LoHi byte order	TC000.256L-TC127.002L .l is string length, range 2 to 256 chars	**String**	Read/Write
Timer/Counter as string Using Only the High Order byte of each word	TC000.128D-TC127.001D .l is string length, range 1 to 128 chars	**String**	Read/Write
Timer/Counter as string Using Only the Low Order byte of each word	TC000.128E-TC127.001E .l is string length, range 1 to 128 chars	**String**	Read/Write
Timer/Counter Status	TS000-TS127	**Boolean**	Read/Write

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range is as follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range is as follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example
IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The C500 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) 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 512 characters and is entered in place of a bit number.The string length cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 512. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register, the length of the string can be from 1 to 256 characters and is entered in place of a bit number. The string length times 2 cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 256. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address. For more information, refer to **Device Setup**.

Examples
1. To address a string starting at DM100 with a length of 100 bytes and HiLo byte order, enter:
DM100.100H

2. To address a string starting at DM110 with a length of 78 bytes and LoHi byte order, enter:
DM110.078L

3. To address a string starting at DM200 with a length of 55 bytes and Only the High Order byte, enter:
DM200.055D

4. To address a string starting at DM220 with a length of 37 bytes and Only the Low Order byte, enter:
DM220.037E

Array Support

Arrays are supported for all data types except Boolean. There are two methods of addressing an array. Examples are given using data memory locations.

DMxxxx [rows] [cols]
*DMxxxx [cols]**

*This method assumes that "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes (2 for Word, Short and BCD; 4 for DWord, Long, LBCD and Float) cannot exceed the request size that has been assigned to the device. For example, a 10 X 10 array of words results in an array size of 200 bytes, which would require a request size of at least 256.

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD and Float). Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Thus, writing to DM0 will also modify the value held in DM1. It is recommended that users utilize these data types so that overlapping does not occur. For example, when using DWords, users may want to use DM0, DM2, DM4 and so on to prevent overlapping Words.

C1000H Addressing

The default data types for dynamically defined tags are shown in **bold**. For more information, refer below the table for notes and restrictions on **BCD Support**, **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Auxiliary Relay	AR00-AR27 AR00-AR26 ARxx.00-ARxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Auxiliary Relay as string with HiLo byte order	AR00.056H-AR27.002H .l is string length, range 2 to 56 chars	**String**	Read/Write
Auxiliary Relay as string with LoHi byte order	AR00.056L-AR27.002L .l is string length, range 2 to 56 chars	**String**	Read/Write
Auxiliary Relay as string Using Only the High Order byte of each word	AR00.028D-AR27.001D .l is string length, range 1 to 28 chars	**String**	Read/Write
Auxiliary Relay as string Using Only the Low Order byte of each word	AR00.028E-AR27.001E .l is string length, range 1 to 28 chars	**String**	Read/Write
Data Memory	DM0000-DM4095 DM0000-DM4094 DMxxxx.00-DMxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Data Memory as string	DM0000.512H-DM4095.002H	**String**	Read/Write

with HiLo byte order	.l is string length, range 2 to 512 chars		
Data Memory as string with LoHi byte order	DM0000.512L-DM4095.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string Using Only the High Order byte of each word	DM0000.256D-DM4095.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory as string Using Only the Low Order byte of each word	DM0000.256E-DM4095.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Holding Relay	HR00-HR99 HR00-HR98 HRxx.00-HRxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Holding Relay as string with HiLo byte order	HR00.200H-HR99.002H .l is string length, range 2 to 200 chars	**String**	Read/Write
Holding Relay as string with LoHi byte order	HR00.200L-HR99.002L .l is string length, range 2 to 200 chars	**String**	Read/Write
Holding Relay as string Using Only the High Order byte of each word	HR00.100D-HR99.001D .l is string length, range 1 to 100 chars	**String**	Read/Write
Holding Relay as string Using Only the Low Order byte of each word	HR00.100E-HR99.001E .l is string length, range 1 to 100 chars	**String**	Read/Write
Internal Relay	IR000-IR255 IR000-IR254 IRxxx.00-IRxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Internal Relay as a string with HiLo byte order	IR000.512H-IR255.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Internal Relay as a string with LoHi byte order	IR000.512L-IR255.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Internal Relay as a string Using Only the High Order byte of each word	IR000.256D-IR255.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Internal Relay as a string Using Only the Low Order byte of each word	IR000.256E-IR255.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Link Relays	LR00-LR63 LR00-LR62 LRxx.00-LRxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write

Link Relays as string with HiLo byte order	LR00.128H-LR63.002H .l is string length, range 2 to 128 chars	**String**	Read/Write
Link Relays as string with LoHi byte order	LR00.128L-LR63.002L .l is string length, range 2 to 128 chars	**String**	Read/Write
Link Relays as string Using Only the High Order byte of each word	LR00.064D-LR63.001D .l is string length, range 1 to 64 chars	**String**	Read/Write
Link Relays as string Using Only the Low Order byte of each word	LR00.064E-LR63.001E .l is string length, range 1 to 64 chars	**String**	Read/Write
Temporary Relay	TR TR0-TR7	**Word**, Short **Boolean**	Read/Write
Timer/Counter	TC000-TC511	**BCD**, Word, Short	Read/Write
Timer/Counter as string with HiLo byte order	TC000.512H-TC511.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer/Counter as string with LoHi byte order	TC000.512L-TC511.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer/Counter as string Using Only the High Order byte of each word	TC000.256D-TC511.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer/Counter as string Using Only the Low Order byte of each word	TC000.256E-TC511.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer/Counter Status	TS000-TS511	**Boolean**	Read/Write

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values. Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range is as follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range is as follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example:
IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The C1000H model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) 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 512 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 512. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register, the length of the string can be from 1 to 256 characters and is entered in place of a bit number. The string length times 2 cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 256. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address. For more information, refer to **Device Setup**.

Examples

1. To address a string starting at DM1000 with a length of 100 bytes and HiLo byte order, enter:
DM1000.100H

2. To address a string starting at DM1100 with a length of 78 bytes and LoHi byte order, enter:
DM1100.078L

3. To address a string starting at DM2000 with a length of 55 bytes and Only the High Order byte, enter:
DM2000.055D

4. To address a string starting at DM2200 with a length of 37 bytes and Only the Low Order byte, enter:
DM2200.037E

Array Support

Arrays are supported for all data types except Boolean. There are two methods of addressing an array. Examples are given using data memory locations.

DMxxxx [rows] [cols]
*DMxxxx [cols]**

*This method assumes that "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes (2 for Word, Short and BCD; 4 for DWord, Long, LBCD and Float) cannot exceed the request size that has been assigned to the device. For example, a 10 X 10 array of words results in an array size of 200 bytes, which would require a request size of at least 256.

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD and Float). Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Thus, writing to DM0 will also modify the value held in DM1. It is recommended that users utilize these data types so that overlapping does not occur. For example, when using DWords, users may want to use DM0, DM2, DM4 and so on to prevent overlapping Words.

C2000H Addressing

The default data types for dynamically defined tags are shown in **bold**. For more information, refer below the table for notes and restrictions on **BCD Support**, **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Auxiliary Relay	AR00-AR27 AR00-AR26 ARxx.00-ARxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Auxiliary Relay as string with HiLo byte order	AR00.056H-AR27.002H .l is string length, range 2 to 56	**String**	Read/Write

	chars		
Auxiliary Relay as string with LoHi byte order	AR00.056L-AR27.002L .l is string length, range 2 to 56 chars	**String**	Read/Write
Auxiliary Relay as string Using Only the High Order byte of each word	AR00.028D-AR27.001D .l is string length, range 1 to 28 chars	**String**	Read/Write
Auxiliary Relay as string Using Only the Low Order byte of each word	AR00.028E-AR27.001E .l is string length, range 1 to 28 chars	**String**	Read/Write
Data Memory	DM0000-DM6655 DM0000-DM6654 DMxxxx.00-DMxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Data Memory as string with HiLo byte order	DM0000.512H-DM6655.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string with LoHi byte order	DM0000.512L-DM6655.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string Using Only the High Order byte of each word	DM0000.256D-DM6655.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory as string Using Only the Low Order byte of each word	DM0000.256E-DM6655.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Holding Relay	HR00-HR99 HR00-HR98 HRxx.00-HRxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Holding Relay as string with HiLo byte order	HR00.200H-HR99.002H .l is string length, range 2 to 200 chars	**String**	Read/Write
Holding Relay as string with LoHi byte order	HR00.200L-HR99.002L .l is string length, range 2 to 200 chars	**String**	Read/Write
Holding Relay as string Using Only the High Order byte of each word	HR00.100D-HR99.001D .l is string length, range 1 to 100 chars	**String**	Read/Write
Holding Relay as string Using Only the Low Order byte of each word	HR00.100E-HR99.001E .l is string length, range 1 to 100 chars	**String**	Read/Write
Internal Relay	IR000-IR255 IR000-IR254 IRxxx.00-IRxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Internal Relay as a string with HiLo byte order	IR000.512H-IR255.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Internal Relay as a string with LoHi byte order	IR000.512L-IR255.002L	**String**	Read/Write

	.l is string length, range 2 to 512 chars		
Internal Relay as a string Using Only the High Order byte of each word	IR000.256D-IR255.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Internal Relay as a string Using Only the Low Order byte of each word	IR000.256E-IR255.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Link Relays	LR00-LR63 LR00-LR62 LRxx.00-LRxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Link Relays as string with HiLo byte order	LR00.128H-LR63.002H .l is string length, range 2 to 128 chars	**String**	Read/Write
Link Relays as string with LoHi byte order	LR00.128L-LR63.002L .l is string length, range 2 to 128 chars	**String**	Read/Write
Link Relays as string Using Only the High Order byte of each word	LR00.064D-LR63.001D .l is string length, range 1 to 64 chars	**String**	Read/Write
Link Relays as string Using Only the Low Order byte of each word	LR00.064E-LR63.001E .l is string length, range 1 to 64 chars	**String**	Read/Write
Temporary Relay	TR TR0-TR7	**Word**, Short **Boolean**	Read/Write
Timer/Counter	TC000-TC511	**BCD**, Word, Short	Read/Write
Timer/Counter as string with HiLo byte order	TC000.512H-TC511.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer/Counter as string with LoHi byte order	TC000.512L-TC511.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer/Counter as string Using Only the High Order byte of each word	TC000.256D-TC511.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer/Counter as string Using Only the Low Order byte of each word	TC000.256E-TC511.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer/Counter Status	TS000-TS511	**Boolean**	Read/Write

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values. Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range is as follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range is as follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example
IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The C2000H model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) 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 512 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 512. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register, the length of the string can be from 1 to 256 characters and is entered in place of a bit number. The string length times 2 cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 256. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address. For more information, refer to **Device Setup**.

Examples
1. To address a string starting at DM1000 with a length of 100 bytes and HiLo byte order, enter:
DM1000.100H

2. To address a string starting at DM1100 with a length of 78 bytes and LoHi byte order, enter:
DM1100.078L

3. To address a string starting at DM2000 with a length of 55 bytes and Only the High Order byte, enter:
DM2000.055D

4. To address a string starting at DM2200 with a length of 37 bytes and Only the Low Order byte, enter:
DM2200.037E

Array Support

Arrays are supported for all data types except Boolean. There are two methods of addressing an array. Examples are given using data memory locations.

DMxxxx [rows] [cols]
*DMxxxx [cols]**

*This method assumes that "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes (2 for Word, Short and BCD; 4 for DWord, Long, LBCD and Float) cannot exceed the request size that has been assigned to the device. For example, a 10 X 10 array of words results in an array size of 200 bytes, which would require a request size of at least 256.

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD and Float). Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Thus, writing to DM0 will also modify the value held in DM1. It is recommended that users utilize these data types so that overlapping does not occur. For example, when using DWords, users may want to use DM0, DM2, DM4 and so on to prevent overlapping Words.

CV500 Addressing

The default data types for dynamically defined tags are shown in **bold**. For more information, refer below the table for notes and restrictions on **BCD Support**, **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Action Flag	AC0000-AC1023	**Boolean**	Read Only
Auxiliary Relay	A000-A255 A000-A254 A256-A511 A256-A510 A000.00-A000.15-A255.00-A255.15 A256.00-A256.15-A511.00-A511.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Word**, Short, BCD Long, DWord, LBCD, Float **Boolean** **Boolean**	Read/Write Read Only Read/Write Read Only
Auxiliary Relay as string with HiLo byte order	A000.512H-A255.002H A256.512H-A511.002H .l is string length, range 2 to 512 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string with LoHi byte order	A000.512L-A255.002L A256.512L-A511.002L .l is string length, range 2 to 512 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string Using Only the High Order byte of each word	A000.256D-A255.001D A256.256D-A511.001D .l is string length, range 1 to 256 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.256E-A255.001E A256.256E-A511.001E .l is string length, range 1 to 256 chars	**String** **String**	Read/Write Read Only
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxxx.00-CIOxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.512H-CIO2555.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.512L-CIO2555.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.256D-CIO2555.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.256E-CIO2555.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter	C000-C511	**BCD**, Word, Short	Read/Write
Counter as string with HiLo byte order	C000.512H-C511.002H	**String**	Read/Write

	.l is string length, range 2 to 512 chars		
Counter as string with LoHi byte order	C000.512L-C511.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Counter as string Using Only the High Order byte of each word	C000.256D-C511.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter as string Using Only the Low Order byte of each word	C000.256E-C511.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter Status	CS000-CS511	**Boolean**	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CPU Bus Link as string with HiLo byte order	G000.512H-G255.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CPU Bus Link as string with LoHi byte order	G000.512L-G255.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CPU Bus Link as string Using Only the High Order byte of each word	G000.256D-G255.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CPU Bus Link as string Using Only the Low Order byte of each word	G000.256E-G255.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory	D0000-D8191 D0000-D8190 Dxxxx.00-Dxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Data Memory as string with HiLo byte order	D0000.512H-D8191.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string with LoHi byte order	D0000.512L-D8191.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string Using Only the High Order byte of each word	D0000.256D-D8191.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D0000.256E-D8191.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Register	DR0-DR2	**Word**, Short, BCD*	Read/Write
Index Register	IR0-IR2	**Word**, Short, BCD*	Read/Write
Step Timer	ST000-ST511	**Word**, Short, BCD*	Read/Write
Step Timer Status	STS000-STS511	**Boolean**	Read/Write

Temporary Relay	TR TR0-TR7	**Word**, Short **Boolean**	Read/Write
Timer	T000-T511	**BCD**, Word, Short	Read/Write
Timer as string with HiLo byte order	T000.512H-T511.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer as string with LoHi byte order	T000.512L-T511.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer as string Using Only the High Order byte of each word	T000.256D-T511.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer as string Using Only the Low Order byte of each word	T000.256E-T511.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer Status	TS000-TS511	**Boolean**	Read/Write
Transition Flag	TN000-TN511	**Boolean**	Read/Write

*Arrays are not supported.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range is as follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range is as follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example
IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CV500 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) 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 512 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 512. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register, the length of the string can be from 1 to 256 characters and is entered in place of a bit number. The string length times 2 cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 256. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address. For more information, refer to **Device Setup**.

Examples
1. To address a string starting at D1000 with a length of 100 bytes and HiLo byte order, enter:
D1000.100H

2. To address a string starting at D1100 with a length of 78 bytes and LoHi byte order, enter:
D1100.078L

3. To address a string starting at D2000 with a length of 55 bytes and Only the High Order byte, enter:
D2000.055D

4. To address a string starting at D2200 with a length of 37 bytes and Only the Low Order byte, enter:
D2200.037E

Array Support

Arrays are supported for all data types except Boolean, Data Register, Index Register and Step Timer. There are two methods of addressing an array. Examples are given using data memory locations.

Dxxxx [rows] [cols]
*Dxxxx [cols]**

*This method assumes that "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes (2 for Word, Short and BCD; 4 for DWord, Long, LBCD and Float) cannot exceed the request size that has been assigned to the device. For example, a 10 X 10 array of words results in an array size of 200 bytes, which would require a request size of at least 256.

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD and Float). Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Thus, writing to DM0 will also modify the value held in DM1. It is recommended that users utilize these data types so that overlapping does not occur. For example, when using DWords, users may want to use DM0, DM2, DM4 and so on to prevent overlapping Words.

CV1000 Addressing

The default data types for dynamically defined tags are shown in **bold**. For more information, refer below the table for notes and restrictions on **BCD Support**, **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Action Flag	AC0000-AC2047	**Boolean**	Read Only
Auxiliary Relay	A000-A255 A000-A254 A256-A511 A256-A510 A000.00-A000.15-A255.00-A255.15 A256.00-A256.15-A511.00-A511.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Word**, Short, BCD Long, DWord, LBCD, Float **Boolean** **Boolean**	Read/Write Read Only Read/Write Read Only
Auxiliary Relay as string with HiLo byte order	A000.512H-A255.002H A256.512H-A511.002H .l is string length, range 2 to 512 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string with LoHi byte order	A000.512L-A255.002L	**String** **String**	Read/Write

	A256.512L-A511.002L .l is string length, range 2 to 512 chars		Read Only
Auxiliary Relay as string Using Only the High Order byte of each word	A000.256D-A255.001D A256.256D-A511.001D .l is string length, range 1 to 256 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.256E-A255.001E A256.256E-A511.001E .l is string length, range 1 to 256 chars	**String** **String**	Read/Write Read Only
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxxx.00-CIOxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.512H-CIO2555.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.512L-CIO2555.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.256D-CIO2555.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.256E-CIO2555.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter	C0000-C1023	**BCD**, Word, Short	Read/Write
Counter as string with HiLo byte order	C0000.512H-C1023.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Counter as string with LoHi byte order	C0000.512L-C1023.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Counter as string Using Only the High Order byte of each word	C0000.256D-C1023.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter as string Using Only the Low Order byte of each word	C0000.256E-C1023.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter Status	CS0000-CS1023	**Boolean**	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CPU Bus Link as string with HiLo byte order	G000.512H-G255.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CPU Bus Link as string with LoHi byte order	G000.512L-G255.002L .l is string length, range 2 to 512	**String**	Read/Write

	chars		
CPU Bus Link as string Using Only the High Order byte of each word	G000.256D-G255.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CPU Bus Link as string Using Only the Low Order byte of each word	G000.256E-G255.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory	D00000-D24575 D00000-D24574 Dxxxxx.00-Dxxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Data Memory as string with HiLo byte order	D00000.512H-D24575.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string with LoHi byte order	D00000.512L-D24575.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string Using Only the High Order byte of each word	D00000.256E-D24575.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D00000.256D-D24575.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Register	DR0-DR2	**Word**, Short, BCD*	Read/Write
Expansion Data Memory (current bank)	E00000-E32765 E00000-E32764 Exxxxx.00-Exxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.512H-E32765.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.512L-E32765.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory (current bank) Using Only the High Order byte of each word	E00000.256D-E32765.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory (current bank) Using Only the Low Order byte of each word	E00000.256E-E32765.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory	E00:00000-E07:32765 E00:00000-E07:32764 Ex:x.00-Exx:xxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory as string with HiLo byte order	E00:00000.512H-E07:32765.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory as string with LoHi byte order	E00:00000.512L-E07:32765.002L .l is string length, range 2 to	**String**	Read/Write

	512 chars		
Expansion Data Memory as string Using Only the High Order byte of each word	E00:00000.256D-E07:32765.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory as string Using Only the Low Order byte of each word	E00:00000.256E-E07:32765.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Index Register	IR0-IR2	**Word**, Short, BCD*	Read/Write
Step Timer	ST0000-ST1023	**Word**, Short, BCD*	Read/Write
Step Timer Status	STS0000-STS1023	**Boolean**	Read/Write
Temporary Relay	TR TR0-TR7	**Word**, Short **Boolean**	Read/Write
Timer	T0000-T1023	**BCD**, Word, Short	Read/Write
Timer as string with HiLo byte order	T0000.512H-T1023.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer as string with LoHi byte order	T0000.512L-T1023.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer as string Using Only the High Order byte of each word	T0000.256D-T1023.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer as string Using Only the Low Order byte of each word	T0000.256E-T1023.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer Status	TS0000-TS1023	**Boolean**	Read/Write
Transition Flag	TN0000-TN1023	**Boolean**	Read/Write

*Arrays are not supported.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values. Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range is as follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range is as follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example
IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50

IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CV1000 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) 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 512 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 512. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register, the length of the string can be from 1 to 256 characters and is entered in place of a bit number. The string length times 2 cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 256. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address. For more information, refer to **Device Setup**.

Examples
1. To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H

2. To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter:
D01100.078L

3. To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter:
D02000.055D

4. To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter:
D02200.037E

Array Support

Arrays are supported for all data types except Boolean, Data Register, Index Register and Step Timer. There are two methods of addressing an array. Examples are given using data memory locations.

Dxxxx [rows] [cols]
*Dxxxx [cols]**

*This method assumes that "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes (2 for Word, Short and BCD; 4 for DWord, Long, LBCD and Float) cannot exceed the request size that has been assigned to the device. For example, a 10 X 10 array of words results in an array size of 200 bytes, which would require a request size of at least 256.

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD and Float). Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1. Thus, writing to D0 will also modify the value held in D1. It is recommended that users utilize these data types so that overlapping does not occur. For example, when using DWords, users may want to use D0, D2, D4 and so on to prevent overlapping Words.

CV2000 Addressing

The default data types for dynamically defined tags are shown in **bold**. For more information, refer below the table for notes and restrictions on **BCD Support**, **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Action Flag	AC0000-AC2047	**Boolean**	Read Only
Auxiliary Relay	A000-A255 A000-A254 A256-A511	**Word**, Short, BCD Long, DWord, LBCD, Float	Read/Write Read Only

	A256-A510 A000.00-A000.15-A255.00-A255.15 A256.00-A256.15-A511.00-A511.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean** **Boolean**	Read/Write Read Only
Auxiliary Relay as string with HiLo byte order	A000.512H-A255.002H A256.512H-A511.002H .l is string length, range 2 to 512 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string with LoHi byte order	A000.512L-A255.002L A256.512L-A511.002L .l is string length, range 2 to 512 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string Using Only the High Order byte of each word	A000.256D-A255.001D A256.256D-A511.001D .l is string length, range 1 to 256 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.256E-A255.001E A256.256E-A511.001E .l is string length, range 1 to 256 chars	**String** **String**	Read/Write Read Only
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxxx.00-CIOxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.512H-CIO2555.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.512L-CIO2555.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.256D-CIO2555.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.256E-CIO2555.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter	C0000-C1023	**BCD**, Word, Short	Read/Write
Counter as string with HiLo byte order	C0000.512H-C1023.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Counter as string with LoHi byte order	C0000.512L-C1023.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Counter as string Using Only the High Order byte of each word	C0000.256D-C1023.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter as string Using Only the Low Order byte of each word	C0000.256E-C1023.001E .l is string length, range 1 to 256	**String**	Read/Write

	chars		
Counter Status	CS0000-CS1023	**Boolean**	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CPU Bus Link as string with HiLo byte order	G000.512H-G255.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CPU Bus Link as string with LoHi byte order	G000.512L-G255.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CPU Bus Link as string Using Only the High Order byte of each word	G000.256D-G255.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CPU Bus Link as string Using Only the Low Order byte of each word	G000.256E-G255.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory	D00000-D24575 D00000-D24574 Dxxxxx.00-Dxxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Data Memory as string with HiLo byte order	D00000.512H-D24575.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string with LoHi byte order	D00000.512L-D24575.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string Using Only the High Order byte of each word	D00000.256D-D24575.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D00000.256E-D24575.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Register	DR0-DR2	**Word**, Short, BCD*	Read/Write
Expansion Data Memory (current bank)	E00000-E32765 E00000-E32764 Exxxxx.00-Exxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.512H-E32765.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.512L-E32765.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string Using Only the High Order byte of each word	E00000.256D-E32765.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string Using Only the Low Order byte of each word	E00000.256E-E32765.001E .l is string length, range 1 to 256 chars	**String**	Read/Write

Expansion Data Memory	E00:00000-E07:32765 E00:00000-E07:32764 Ex:x.00-Exx:xxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory as string with HiLo byte order	E00:00000.512H - E07:32765.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory as string with LoHi byte order	E00:00000.512L - E07:32765.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory as string Using Only the High Order byte of each word	E00:00000.256D-E07:32765.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory as string Using Only the Low Order byte of each word	E00:00000.256E-E07:32765.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Index Register	IR0-IR2	**Word**, Short, BCD*	Read/Write
Step Timer	ST0000-ST1023	**Word**, Short, BCD*	Read/Write
Step Timer Status	STS0000-STS1023	**Boolean**	Read/Write
Temporary Relay	TR TR0-TR7	**Word**, Short **Boolean**	Read/Write
Timer	T0000-T1023	**BCD**, Word, Short	Read/Write
Timer as string with HiLo byte order	T0000.512H-T1023.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer as string with LoHi byte order	T0000.512L-T1023.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer as string Using Only the High Order byte of each word	T0000.256D-T1023.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer as string Using Only the Low Order byte of each word	T0000.256E-T1023.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer Status	TS0000-TS1023	**Boolean**	Read/Write
Transition Flag	TN0000-TN1023	**Boolean**	Read/Write

*Arrays are not supported.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range is as follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range is as follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example
IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CV2000 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) 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 512 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 512. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register, the length of the string can be from 1 to 256 characters and is entered in place of a bit number. The string length times 2 cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 256. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address. For more information, refer to **Device Setup**.

Examples
1. To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H

2. To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter:
D01100.078L

3. To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter:
D02000.055D

4. To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter:
D02200.037E

Array Support

Arrays are supported for all data types except Boolean, Data Register, Index Register and Step Timer. There are two methods of addressing an array. Examples are given using data memory locations.

Dxxxx [rows] [cols]
*Dxxxx [cols]**

*This method assumes that "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes (2 for Word, Short and BCD; 4 for DWord, Long, LBCD and Float) cannot exceed the request size that has been assigned to the device. For example, a 10 X 10 array of words results in an array size of 200 bytes, which would require a request size of at least 256.

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD and Float). Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1. Thus, writing to D0 will also modify the value held in D1. It is recommended that users utilize these data types so that overlapping does not occur. For example, when using DWords, users may want to use D0, D2, D4 and so on to prevent overlapping Words.

CVM1-CPU01 Addressing

The default data types for dynamically defined tags are shown in **bold**. For more information, refer below the table for notes and restrictions on **BCD Support**, **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Auxiliary Relay	A000-A255 A000-A254 A256-A511 A256-A510 A000.00-A000.15-A255.00-A255.15 A256.00-A256.15-A511.00-A511.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Word**, Short, BCD Long, DWord, LBCD, Float **Boolean** **Boolean**	Read/Write Read Only Read/Write Read Only
Auxiliary Relay as string with HiLo byte order	A000.512H-A255.002H A256.512H-A511.002H .l is string length, range 2 to 512 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string with LoHi byte order	A000.512L-A255.002L A256.512L-A511.002L .l is string length, range 2 to 512 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string Using Only the High Order byte of each word	A000.256D-A255.001D A256.256D-A511.001D .l is string length, range 1 to 256 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.256E-A255.001E A256.256E-A511.001E .l is string length, range 1 to 256 chars	**String** **String**	Read/Write Read Only
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxxx.00-CIOxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.512H-CIO2555.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.512L-CIO2555.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.256D-CIO2555.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.256E-CIO2555.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter	C000-C511	**BCD**, Word, Short	Read/Write
Counter as string with HiLo byte order	C000.512H-C511.002H	**String**	Read/Write

	.l is string length, range 2 to 512 chars		
Counter as string with LoHi byte order	C000.512L-C511.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Counter as string Using Only the High Order byte of each word	C000.256D-C511.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter as string Using Only the Low Order byte of each word	C000.256E-C511.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter Status	CS000-CS511	**Boolean**	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CPU Bus Link as string with HiLo byte order	G000.512H-G255.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CPU Bus Link as string with LoHi byte order	G000.512L-G255.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CPU Bus Link as string Using Only the High Order byte of each word	G000.256D-G255.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CPU Bus Link as string Using Only the Low Order byte of each word	G000.256E-G255.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory	D0000-D8191 D0000-D8190 Dxxxx.00-Dxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Data Memory as string with HiLo byte order	D0000.512H-D8191.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string with LoHi byte order	D0000.512L- D8191.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string Using Only the High Order byte of each word	D0000.256D-D8191.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D0000.256E-D8191.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Register	DR0-DR2	**Word**, Short, BCD*	Read/Write
Index Register	IR0-IR2	**Word**, Short, BCD*	Read/Write
Temporary Relay	TR TR0-TR7	**Word**, Short **Boolean**	Read/Write
Timer	T000-T511	**BCD**, Word, Short	Read/Write
Timer as string with HiLo byte order	T000.512H-T511.002H	**String**	Read/Write

	.l is string length, range 2 to 512 chars		
Timer as string with LoHi byte order	T000.512L-T511.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer as string Using Only the High Order byte of each word	T000.256D-T511.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer as string Using Only the Low Order byte of each word	T000.256E-T511.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer Status	TS000-TS511	**Boolean**	Read/Write

*Arrays are not supported.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range is as follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range is as follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example
IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CVM1-CPU01 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) 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 512 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 512. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register, the length of the string can be from 1 to 256 characters and is entered in place of a bit number. The string length times 2 cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 256. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address. For more information, refer to **Device Setup**.

Examples
1. To address a string starting at D1000 with a length of 100 bytes and HiLo byte order, enter:
D1000.100H

2. To address a string starting at D1100 with a length of 78 bytes and LoHi byte order, enter:
D1100.078L

3. To address a string starting at D2000 with a length of 55 bytes and Only the High Order byte, enter:
D2000.055D

4. To address a string starting at D2200 with a length of 37 bytes and Only the Low Order byte, enter:
D2200.037E

Array Support

Arrays are supported for all data types except Boolean, Data Register and Index Register. There are two methods of addressing an array. Examples are given using data memory locations.

Dxxxx [rows] [cols]
*Dxxxx [cols]**

*This method assumes that "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes (2 for Word, Short and BCD; 4 for DWord, Long, LBCD and Float) cannot exceed the request size that has been assigned to the device. For example, a 10 X 10 array of words results in an array size of 200 bytes, which would require a request size of at least 256.

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD and Float). Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Thus, writing to DM0 will also modify the value held in DM1. It is recommended that users utilize these data types so that overlapping does not occur. For example, when using DWords, users may want to use DM0, DM2, DM4 and so on to prevent overlapping Words.

CVM1-CPU11 Addressing

The default data types for dynamically defined tags are shown in **bold**. For more information, refer below the table for notes and restrictions on **BCD Support**, **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Auxiliary Relay	A000-A255 A000-A254 A256-A511 A256-A510 A000.00-A000.15-A255.00-A255.15 A256.00-A256.15-A511.00-A511.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Word**, Short, BCD Long, DWord, LBCD, Float **Boolean** **Boolean**	Read/Write Read Only Read/Write Read Only
Auxiliary Relay as string with HiLo byte order	A000.512H-A255.002H A256.512H-A511.002H .l is string length, range 2 to 512 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string with LoHi byte order	A000.512L-A255.002L A256.512L-A511.002L .l is string length, range 2 to 512 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string Using Only the High Order byte of each word	A000.256D-A255.001D A256.256D-A511.001D .l is string length, range 1 to 256	**String** **String**	Read/Write Read Only

	chars		
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.256E-A255.001E A256.256E-A511.001E .l is string length, range 1 to 256 chars	**String** **String**	Read/Write Read Only
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxxx.00-CIOxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.512H-CIO2555.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.512L-CIO2555.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.256D-CIO2555.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.256E-CIO2555.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter	C0000-C1023	**BCD**, Word, Short	Read/Write
Counter as string with HiLo byte order	C0000.512H-C1023.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Counter as string with LoHi byte order	C0000.512L-C1023.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Counter as string Using Only the High Order byte of each word	C0000.256D-C1023.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter as string Using Only the Low Order byte of each word	C0000.256E-C1023.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter Status	CS0000-CS1023	**Boolean**	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CPU Bus Link as string with HiLo byte order	G000.512H-G255.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CPU Bus Link as string with LoHi byte order	G000.512L-G255.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CPU Bus Link as string Using Only the High Order byte of each word	G000.256D-G255.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CPU Bus Link as string Using Only the Low Order byte of each word	G000.256E-G255.001E	**String**	Read/Write

	.l is string length, range 1 to 256 chars		
Data Memory	D00000-D24575 D00000-D24574 Dxxxxx.00-Dxxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Data Memory as string with HiLo byte order	D00000.512H-D24575.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string with LoHi byte order	D00000.512L-D24575.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string Using Only the High Order byte of each word	D00000.256D-D24575.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D00000.256E-D24575.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Register	DR0-DR2	**Word**, Short, BCD*	Read/Write
Index Register	IR0-IR2	**Word**, Short, BCD*	Read/Write
Temporary Relay	TR TR0-TR7	**Word**, Short **Boolean**	Read/Write
Timer	T0000-T1023	**BCD**, Word, Short	Read/Write
Timer as string with HiLo byte order	T0000.512H-T1023.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer as string with LoHi byte order	T0000.512L-T1023.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer as string Using Only the High Order byte of each word	T0000.256D-T1023.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer as string Using Only the Low Order byte of each word	T0000.256E-T1023.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer Status	TS0000-TS1023	**Boolean**	Read/Write

*Arrays are not supported.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range is as follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range is as follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example
IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CVM1-CPU11 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) 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 512 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 512. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register, the length of the string can be from 1 to 256 characters and is entered in place of a bit number. The string length times 2 cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 256. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address. For more information, refer to **Device Setup**.

Examples
1. To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H

2. To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter:
D01100.078L

3. To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter:
D02000.055D

4. To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter:
D02200.037E

Array Support

Arrays are supported for all data types except Boolean, Data Register and Index Register. There are two methods of addressing an array. Examples are given using data memory locations.

Dxxxx [rows] [cols]
*Dxxxx [cols]**

*This method assumes that "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes (2 for Word, Short and BCD; 4 for DWord, Long, LBCD and Float) cannot exceed the request size that has been assigned to the device. For example, a 10 X 10 array of words results in an array size of 200 bytes, which would require a request size of at least 256.

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD and Float). Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Thus, writing to DM0 will also modify the value held in DM1. It is recommended that users utilize these data types so that overlapping does not occur. For example, when using DWords, users may want to use DM0, DM2, DM4 and so on to prevent overlapping Words.

CVM1-CPU21 Addressing

The default data types for dynamically defined tags are shown in **bold**. For more information, refer below the table for notes and restrictions on **BCD Support**, **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Auxiliary Relay	A000-A255 A000-A254 A256-A511 A256-A510 A000.00-A000.15-A255.00-A255.15 A256.00-A256.15-A511.00-A511.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Word**, Short, BCD Long, DWord, LBCD, Float **Boolean** **Boolean**	Read/Write Read Only Read/Write Read Only
Auxiliary Relay as string with HiLo byte order	A000.512H-A255.002H A256.512H-A511.002H .l is string length, range 2 to 512 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string with LoHi byte order	A000.512L-A255.002L A256.512L-A511.002L .l is string length, range 2 to 512 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string Using Only the High Order byte of each word	A000.256D-A255.001D A256.256D-A511.001D .l is string length, range 1 to 256 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.256E-A255.001E A256.256E-A511.001E .l is string length, range 1 to 256 chars	**String** **String**	Read/Write Read Only
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxxx.00-CIOxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.512H-CIO2555.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.512L-CIO2555.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.256D-CIO2555.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.256E-CIO2555.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter	C0000-C1023	**BCD**, Word, Short	Read/Write
Counter as string with HiLo byte order	C0000.512H-C1023.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Counter as string	C0000.512L-C1023.002L	**String**	Read/Write

with LoHi byte order	.l is string length, range 2 to 512 chars		
Counter as string Using Only the High Order byte of each word	C0000.256D-C1023.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter as string Using Only the Low Order byte of each word	C0000.256E-C1023.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter Status	CS0000-CS1023	**Boolean**	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CPU Bus Link as string with HiLo byte order	G000.512H-G255.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CPU Bus Link as string with LoHi byte order	G000.512L-G255.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CPU Bus Link as string Using Only the High Order byte of each word	G000.256D-G255.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CPU Bus Link as string Using Only the Low Order byte of each word	G000.256E-G255.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory	D00000-D24575 D00000-D24574 Dxxxxx.00-Dxxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Data Memory as string with HiLo byte order	D00000.512H-D24575.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string with LoHi byte order	D00000.512L-D24575.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string Using Only the High Order byte of each word	D00000.256D-D24575.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D00000.256E-D24575.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Register	DR0-DR2	**Word**, Short, BCD*	Read/Write
Expansion Data Memory (current bank)	E00000-E32765 E00000-E32764 Exxxxx.00-Exxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.512H-E32765.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory	E00000.512L-E32765.002L	**String**	Read/Write

(current bank) as string with LoHi byte order	.l is string length, range 2 to 512 chars		
Expansion Data Memory (current bank) as string Using Only the High Order byte of each word	E00000.256D-E32765.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string Using Only the Low Order byte of each word	E00000.256E-E32765.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory	E00:00000-E07:32765 E00:00000-E07:32764 Ex:x.00-Exx:xxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory as string with HiLo byte order	E00:00000.512H-E07:32765.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory as string with LoHi byte order	E00:00000.512L-E07:32765.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory as string Using Only the High Order byte of each word	E00:00000.256D-E07:32765.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory as string Using Only the Low Order byte of each word	E00:00000.256E-E07:32765.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Index Register	IR0-IR2	**Word**, Short, BCD*	Read/Write
Temporary Relay	TR TR0-TR7	**Word**, Short **Boolean**	Read/Write
Timer	T0000-T1023	**BCD**, Word, Short	Read/Write
Timer as string with HiLo byte order	T0000.512H-T1023.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer as string with LoHi byte order	T0000.512L-T1023.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer as string Using Only the High Order byte of each word	T0000.256D-T1023.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer as string Using Only the Low Order byte of each word	T0000.256E-T1023.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer Status	TS0000-TS1023	**Boolean**	Read/Write

*Arrays are not supported.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range is as follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range is as follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example
IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CVM1-CPU21 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) 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 512 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 512. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register, the length of the string can be from 1 to 256 characters and is entered in place of a bit number. The string length times 2 cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 256. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address. For more information, refer to **Device Setup**.

Examples
1. To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H

2. To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter:
D01100.078L

3. To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter:
D02000.055D

4. To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter:
D02200.037E

Array Support

Arrays are supported for all data types except Boolean, Data Register and Index Register. There are two methods of addressing an array. Examples are given using data memory locations.

Dxxxx [rows] [cols]
*Dxxxx [cols]**

*This method assumes that "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes (2 for Word, Short and BCD; 4 for DWord, Long, LBCD and Float) cannot exceed the request size that has been assigned to the device. For example, a 10 X 10 array of words results in an array size of 200 bytes, which would require a request size of at least 256.

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD and Float). Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Thus, writing to DM0 will also modify the value held in DM1. It is recommended that users utilize these data types so that overlapping does not occur. For example, when using DWords, users may want to use DM0, DM2, DM4 and so on to prevent overlapping Words.

CS1 Addressing

The default data types for dynamically defined tags are shown in **bold**. For more information, refer below the table for notes and restrictions on **BCD Support**, **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Auxiliary Relay	A000-A447 A000-A446 A448-A959 A448-A958 A000.00-A000.15-A447.00-A447.15 A448.00-A448.15-A959.00-A959.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Word**, Short, BCD Long, DWord, LBCD, Float **Boolean** **Boolean**	Read Only Read/Write Read Only Read/Write
Auxiliary Relay as string with HiLo byte order	A000.512H-A447.002H A448.512H-A959.002H .l is string length, range 2 to 512 chars	**String** **String**	Read Only Read/Write
Auxiliary Relay as string with LoHi byte order	A000.512L-A447.002L A448.512L-A959.002L .l is string length, range 2 to 512 chars	**String** **String**	Read Only Read/Write
Auxiliary Relay as string Using Only the High Order byte of each word	A000.256D-A447.001D A448.256D-A959.001D .l is string length, range 1 to 256 chars	**String** **String**	Read Only Read/Write
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.256E-A447.001E A448.256E-A959.001E .l is string length, range 1 to 256 chars	**String** **String**	Read Only Read/Write
CIO	CIO0000-CIO6143 CIO0000-CIO6142 CIOxxxx.00-CIOxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.512H-CIO6143.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.512L-CIO6143.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.256D-CIO6143.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CIO Memory as string	CIO0000.256E-CIO6143.001E	**String**	Read/Write

Using Only the Low Order byte of each word	.l is string length, range 1 to 256 chars		
Counter	C0000-C4095	**BCD**, Word, Short	Read/Write
Counter as string with HiLo byte order	C0000.512H-C4095.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Counter as string with LoHi byte order	C0000.512L-C4095.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Counter as string Using Only the High Order byte of each word	C0000.256D-C4095.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter as string Using Only the Low Order byte of each word	C0000.256E-C4095.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter Status	CS0000-CS4095	**Boolean**	Read/Write*
Data Memory	D00000-D32767 D00000-D32766 Dxxxxx.00-Dxxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Data Memory as string with HiLo byte order	D00000.512H-D32767.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string with LoHi byte order	D00000.512L-D32767.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string Using Only the High Order byte of each word	D00000.256D-D32767.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D00000.256E-D32767.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Register	DR00-DR15 DR00-DR14	**Word**, Short, BCD Long, DWord, LBCD, Float	Read/Write*
Expansion Data Memory (current bank)	E00000-E32767 E00000-E32766 Exxxxx.00-Exxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.512H-E32767.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.512L-E32767.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string Using Only the High Order byte of each word	E00000.256D-E32767.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string	E00000.256E-E32767.001E	**String**	Read/Write

Using Only the Low Order byte of each word	.l is string length, range 1 to 256 chars		
Expansion Data Memory	E00:00000-E12:32767 E00:00000-E12:32766 Ex:x.00-Exx:xxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory as string with HiLo byte order	E00:00000.512H-E12:32767.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory as string with LoHi byte order	E00:00000.512L-E12:32767.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory as string Using Only the High Order byte of each word	E00:00000.256D-E12:32767.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory as string Using Only the Low Order byte of each word	E00:00000.256E-E12:32767.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Holding Relay	H0000-H1535 H0000-H1534 Hxxxx.00-Hxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Holding Relay as string with HiLo byte order	H0000.512H-H1535.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Holding Relay as string with LoHi byte order	H0000.512L-H1535.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Holding Relay as string Using Only the High Order byte of each word	H0000.256D-H1535.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Holding Relay as string Using Only the Low Order byte of each word	H0000.256E-H1535.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Index Register	IR00-IR15	**DWord**, Long, LBCD, Float	Read/Write*
Task Flag	TK00-TK31	**Boolean**	Read Only
Timer	T0000-T4095	**BCD**, Word, Short	Read/Write
Timer as string with HiLo byte order	T0000.512H-T4095.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer as string with LoHi byte order	T0000.512L-T4095.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Timer as string Using Only the High Order byte of each word	T0000.256D-T4095.001D .l is string length, range 1 to 256 chars	**String**	Read/Write

Timer as string Using Only the Low Order byte of each word	T0000.256E-T4095.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Timer Status	TS0000-TS4095	**Boolean**	Read/Write*
Working Relay	W000-W511 W000-W510 Wxxx.00-Wxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Working Relay as string with HiLo byte order	W000.512H-W511.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Working Relay as string with LoHi byte order	W000.512L-W511.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Working Relay as string Using Only the High Order byte of each word	W000.256D-W511.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Working Relay as string Using Only the Low Order byte of each word	W000.256E-W511.001E .l is string length, range 1 to 256 chars	**String**	Read/Write

*Use caution when modifying 32-bit values (DWord, Long, LBCD and Float). Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1 and writing to D0 will also modify the value held in D1. It is recommended that users utilize these data types so that overlapping does not occur. When using DWords, users may want to use D0, D2, D4 and so on to prevent overlapping Words. The exception to this is IR tags, which are native 32-bit values with MSB to LSB byte ordering for the CS1-series PLCs.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range is as follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range is as follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example
D0D @ Short = -50
D0D @ Word = 8050 (MSB set)
D0 @ BCD = 8050

D0D @ Short = 50
D0D @ Word = 50
D0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CS1 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) 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 512 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 512. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register, the length of the string can be from 1 to 256 characters and is entered in place of a bit number. The string length times 2 cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 256. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address. For more information, refer to **Device Setup**.

Examples
1. To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H

2. To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter:
D01100.078L

3. To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter:
D02000.055D

4. To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter:
D02200.037E

Array Support

Arrays are supported for all data types except Boolean. There are two methods of addressing an array. Examples are given using data memory locations.

Dxxxx [rows] [cols]
*Dxxxx [cols]**

*This method assumes that "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes (2 for Word, Short and BCD; 4 for DWord, Long, LBCD and Float) cannot exceed the request size that has been assigned to the device. For example, a 10 X 10 array of words results in an array size of 200 bytes, which would require a request size of at least 256.

Writing to CS, TS, DR and IR Registers

CS and TS registers can be written to only when the device is in Monitor Mode or Programming Mode. DR and IR registers can be written to only when the device is in Programming Mode. CS, TS, DR and IR registers cannot be written to in Run Mode. If the device is in Run Mode and a write is attempted to one of these registers, the value in the device will not change. The following will occur:

- If the device is in Run Mode and a write is attempted to a CS or TS register, an "unable to write" error message will be returned.
- If the device is in Run Mode and a write is attempted to a DR or IR register, the write will succeed. Although no error message will be returned, the value in the device will not change because the device is in Run Mode.

CJ1 Addressing

The default data types for dynamically defined tags are shown in **bold**. For more information, refer below the table for notes and restrictions on **BCD Support**, **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Auxiliary Relay	A000-A447 A000-A446	**Word**, Short, BCD Long, DWord, LBCD, Float	Read Only

	A448-A959 A448-A958 A000.00-A000.15-A447.00-A447.15 A448.00-A448.15-A959.00-A959.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean** **Boolean**	Read/Write Read Only Read/Write
Auxiliary Relay as string with HiLo byte order	A000.512H-A447.002H A448.512H-A959.002H .l is string length, range 2 to 512 chars	**String** **String**	Read Only Read/Write
Auxiliary Relay as string with LoHi byte order	A000.512L-A447.002L A448.512L-A959.002L .l is string length, range 2 to 512 chars	**String** **String**	Read Only Read/Write
Auxiliary Relay as string Using Only the High Order byte of each word	A000.256D-A447.001D A448.256D-A959.001D .l is string length, range 1 to 256 chars	**String** **String**	Read Only Read/Write
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.256E-A447.001E A448.256E-A959.001E .l is string length, range 1 to 256 chars	**String** **String**	Read Only Read/Write
CIO	CIO0000-CIO6143 CIO0000-CIO6142 CIOxxxx.00-CIOxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.512H-CIO6143.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.512L-CIO6143.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.256D-CIO6143.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.256E-CIO6143.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter	C0000-C4095	**BCD**, Word, Short	Read/Write
Counter Status	CS0000-CS4095	**Boolean**	Read/Write*
Data Memory	D00000-D32767 D00000-D32766 Dxxxxx.00-Dxxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Data Memory as string with HiLo byte order	D00000.512H-D32767.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string with LoHi byte order	D00000.512L-D32767.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string	D00000.256D-D32767.001D	**String**	Read/Write

Using Only the High Order byte of each word	.l is string length, range 1 to 256 chars		
Data Memory as string Using Only the Low Order byte of each word	D00000.256E-D32767.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Register	DR00-DR15 DR00-DR14	**Word**, Short, BCD Long, DWord, LBCD, Float	Read/Write*
Expansion Data Memory (current bank)	E00000-E32767 E00000-E32766 Exxxxx.00-Exxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.512H-E32767.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.512L-E32767.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string Using Only the High Order byte of each word	E00000.256D-E32767.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string Using Only the Low Order byte of each word	E00000.256E-E32767.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory	E00:00000-E12:32767 E00:00000-E12:32766 Ex:x.00-Exx:xxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory as string with HiLo byte order	E00:00000.512H-E12:32767.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory as string with LoHi byte order	E00:00000.512L-E12:32767.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory as string Using Only the High Order byte of each word	E00:00000.256D-E12:32767.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory as string Using Only the Low Order byte of each word	E00:00000.256E-E12:32767.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Holding Relay	H0000-H1535 H0000-H1534 Hxxxx.00-Hxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Index Register	IR00-IR15	**DWord**, Long, LBCD, Float	Read/Write*
Task Flag	TK00-TK31	**Boolean**	Read Only
Timer	T0000-T4095	**BCD**, Word, Short	Read/Write

Timer Status	TS0000-TS4095	**Boolean**	Read/Write*
Working Relay	W000-W511 W000-W510 Wxxx.00-Wxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write

*Use caution when modifying 32-bit values (DWord, Long, LBCD and Float). Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1 and writing to D0 will also modify the value held in D1. It is recommended that users utilize these data types so that overlapping does not occur. When using DWords, users may want to use D0, D2, D4 and so on to prevent overlapping Words. The exception to this is IR tags, which are native 32-bit values with MSB to LSB byte ordering for the CJ1-series PLCs.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range is as follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range is as follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example
D0D @ Short = -50
D0D @ Word = 8050 (MSB set)
D0 @ BCD = 8050

D0D @ Short = 50
D0D @ Word = 50
D0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CJ1 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) 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 512 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 512. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register, the length of the string can be from 1 to 256 characters and is entered in place of a bit number. The string length times 2 cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 256. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address. For more information, refer to **Device Setup**.

Examples
1. To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H

2. To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter: D01100.078L

3. To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter: D02000.055D

4. To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter: D02200.037E

Array Support

Arrays are supported for all data types except Boolean. There are two methods of addressing an array. Examples are given using data memory locations.

Dxxxx [rows] [cols]
*Dxxxx [cols]**

*This method assumes that "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes (2 for Word, Short and BCD; 4 for DWord, Long, LBCD and Float) cannot exceed the request size that has been assigned to the device. For example, a 10 X 10 array of words results in an array size of 200 bytes, which would require a request size of at least 256.

Writing to CS, TS, DR and IR Registers

CS and TS registers can be written to only when the device is in Monitor Mode or Programming Mode. DR and IR registers can be written to only when the device is in Programming Mode. CS, TS, DR and IR registers cannot be written to in Run Mode. If the device is in Run Mode and a write is attempted to one of these registers, the value in the device will not change. The following will occur:

- If the device is in Run Mode and a write is attempted to a CS or TS register, an "unable to write" error message will be returned.
- If the device is in Run Mode and a write is attempted to a DR or IR register, the write will succeed. Although no error message will be returned, the value in the device will not change because the device is in Run Mode.

CJ2 Addressing

The default data types for dynamically defined tags are shown in **bold**. For more information, refer below the table for notes and restrictions on **BCD Support**, **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Auxiliary Relay	A000-A447 A000-A446 A448-A1471 A448-A1470 A10000-A11535 A10000-A11534 A000.00-A000.15-A447.00-A447.15 A448.00-A448.15-A1471.00-A1471.15 A10000.00-A10000.15-A11535.00-A11535.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Word**, Short, BCD Long, DWord, LBCD, Float **Word**, Short, BCD Long, DWord, LBCD, Float **Boolean** **Boolean** **Boolean**	Read Only Read/Write Read Only Read Only Read/Write Read Only
Auxiliary Relay as string with HiLo byte order	A000.512H-A447.002H A448.512H-A1471.002H A10000.512H-A11535.002H .l is string length, range 2 to 512 chars	**String** **String** **String**	Read Only Read/Write Read Only
Auxiliary Relay as string	A000.512L-A447.002L	**String**	Read Only

with LoHi byte order	A448.512L-A1471.002L A10000.512L-A11535.002L .l is string length, range 2 to 512 chars	**String** **String**	Read/Write Read Only
Auxiliary Relay as string Using Only the High Order byte of each word	A000.256D-A447.001D A448.256D-A1471.001D A10000.256D-A11535.001D .l is string length, range 1 to 256 chars	**String** **String** **String**	Read Only Read/Write Read Only
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.256E-A447.001E A448.256E-A1471.001E A10000.256E-A11535.001E .l is string length, range 1 to 256 chars	**String** **String** **String**	Read Only Read/Write Read Only
CIO	CIO0000-CIO6143 CIO0000-CIO6142 CIOxxxx.00-CIOxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.512H-CIO6143.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.512L-CIO6143.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.256D-CIO6143.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.256E-CIO6143.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Counter	C0000-C4095	**BCD**, Word, Short	Read/Write
Counter Status	CS0000-CS4095	**Boolean**	Read/Write*
Data Memory	D00000-D32767 D00000-D32766 Dxxxxx.00-Dxxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Data Memory as string with HiLo byte order	D00000.512H-D32767.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string with LoHi byte order	D00000.512L-D32767.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Data Memory as string Using Only the High Order byte of each word	D00000.256D-D32767.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D00000.256E-D32767.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Data Register	DR00-DR15 DR00-DR14	**Word**, Short, BCD Long, DWord, LBCD, Float	Read/Write*

Expansion Data Memory (current bank)	E00000-E32767 E00000-E32766 Exxxxx.00-Exxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.512H-E32767.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.512L-E32767.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string Using Only the High Order byte of each word	E00000.256D-E32767.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory (current bank) as string Using Only the Low Order byte of each word	E00000.256E-E32767.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory	E00:00000-E24:32767 E00:00000-E24:32766 Ex:x.00-Exx:xxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Expansion Data Memory as string with HiLo byte order	E00:00000.512H-E24:32767.002H .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory as string with LoHi byte order	E00:00000.512L-E24:32767.002L .l is string length, range 2 to 512 chars	**String**	Read/Write
Expansion Data Memory as string Using Only the High Order byte of each word	E00:00000.256D-E24:32767.001D .l is string length, range 1 to 256 chars	**String**	Read/Write
Expansion Data Memory as string Using Only the Low Order byte of each word	E00:00000.256E-E24:32767.001E .l is string length, range 1 to 256 chars	**String**	Read/Write
Holding Relay	H0000-H1535 H0000-H1534 Hxxxx.00-Hxxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write
Index Register	IR00-IR15	**DWord**, Long, LBCD, Float	Read/Write*
Task Flag	TK00-TK127	**Boolean**	Read Only
Timer	T0000-T4095	**BCD**, Word, Short	Read/Write
Timer Status	TS0000-TS4095	**Boolean**	Read/Write*
Working Relay	W000-W511 W000-W510 Wxxx.00-Wxxx.15	**Word**, Short, BCD Long, DWord, LBCD, Float **Boolean**	Read/Write

*Use caution when modifying 32-bit values (DWord, Long, LBCD and Float). Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1 and writing to D0 will also modify the value held in D1. It is recommended that users utilize these data types so that overlapping does not occur. When using DWords, users may want to use D0, D2, D4 and so on to prevent overlapping Words. The exception to this is IR tags, which are native 32-bit values with MSB to LSB byte ordering for the CJ2-series PLCs.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range is as follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range is as follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example
D0D @ Short = -50
D0D @ Word = 8050 (MSB set)
D0 @ BCD = 8050

D0D @ Short = 50
D0D @ Word = 50
D0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CJ2 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) 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 512 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 512. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register, the length of the string can be from 1 to 256 characters and is entered in place of a bit number. The string length times 2 cannot exceed the request size that has been assigned to the device up to a request size of 512. For request size values greater than 512, string length cannot exceed 256. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address. For more information, refer to **Device Setup**.

Examples
1. To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H

2. To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter:
D01100.078L

3. To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter:
D02000.055D

4. To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter:
D02200.037E

Array Support

Arrays are supported for all data types except Boolean. There are two methods of addressing an array. Examples are given using data memory locations.

Dxxxx [rows] [cols]
*Dxxxx [cols]**

*This method assumes that "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes (2 for Word, Short and BCD; 4 for DWord, Long, LBCD and Float) cannot exceed the request size that has been assigned to the device. For example, a 10 X 10 array of words results in an array size of 200 bytes, which would require a request size of at least 256.

Writing to CS, TS, DR and IR Registers

CS and TS registers can be written to only when the device is in Monitor Mode or Programming Mode. DR and IR registers can be written to only when the device is in Programming Mode. CS, TS, DR and IR registers cannot be written to in Run Mode. If the device is in Run Mode and a write is attempted to one of these registers, the value in the device will not change. The following will occur:

- If the device is in Run Mode and a write is attempted to a CS or TS register, an "unable to write" error message will be returned.
- If the device is in Run Mode and a write is attempted to a DR or IR register, the write will succeed. Although no error message will be returned, the value in the device will not change because the device is in Run Mode.

Error Descriptions

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

Address Validation

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

Omron FINS Ethernet Device Specific Messages

Device '<device name>' access right denied (Tag '<address>'). [Main, Sub: '<main code, sub code>']
Device '<device name>' cannot accept command (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']
Device '<device name>' cannot process command (Tag '<address>', Size '<bytes>') [Main, Sub: '<main code, sub code>']
Device '<device name>' deactivated due to network settings conflict with device `<device name>'
Device '<device name>' is not responding
Device '<device name>' responded with a command format error (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']
Device '<device name>' responded with a command parameter error (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']
Device '<device name>' responded with a communications error. [Main, Sub: '<main code, sub code>']
Device '<device name>' responded with destination node error. [Main, Sub: '<main code, sub code>']
Device '<device name>' responded with error '<error code>' (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']
Device '<device name>' responded with error in unit. [Main, Sub: '<main code, sub code>']
Device '<device name>' responded with Fatal CPU Unit error. [End Code: '<end code>']
Device '<device name>' responded with local node error. [Main, Sub: '<main code, sub code>']
Device '<device name>' responded with Network relay error. [Net Addr, Node Addr: '<network address, node address>'], [Main, Sub: '<main code, sub code>'], [End Code: '<end code>']
Device '<device name>' responded with Non-fatal CPU Unit error. [End Code: '<end code>']
Device '<device name>' responded with read not possible (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']
Device '<device name>' responded with routing table error. [Main, Sub: '<main code, sub code>']
Device '<device name>' responded with write not possible (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']
The current project was created with an older version of this driver. Examine device properties to validate network parameters
Unable to bind to adapter: '<adapter>'. Connect failed
Unable to write to '<address>' on device '<device name>'
Unable to write to register <register address> for device <device name>. The device is in run mode
Winsock initialization failed (OS Error = n)
Winsock V1.1 or higher must be installed to use the Omron FINS Ethernet device driver

See Also: Main and Sub Error Codes

Address Validation

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

Address Validation

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

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

Error Type:
Warning

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

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

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

Error Type:
Warning

Possible Cause:
A tag address that has been specified 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:
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.

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.

Omron FINS Ethernet Device Specific Messages

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

Omron FINS Ethernet Device Specific Messages

Device '<device name>' access right denied (Tag '<address>'). [Main, Sub: '<main code, sub code>']

Error Type:
Warning

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' cannot accept command (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:
Warning

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' cannot process command (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:
Warning

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' deactivated due to network settings conflict with device '<device name>'

Error Type:
Serious

Possible Cause:
The network parameters for the stated device were changed during run time and were found to be in conflict with another device. Communication with this device will not be possible until the conflict is resolved.

Solution:
Change one or more of the following: channel network adapter, port number or Device ID.

Note:
For more information, refer to the "Duplicate Devices" section of **FINS Networks**.

Device '<device name>' is not responding

Error Type:
Serious

Possible Cause:
1. The connection between the device and the Host PC is broken.
2. The IP address assigned to the device is incorrect.
3. The named device may have been assigned an incorrect Network ID.
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 PLC device.
2. Verify that the IP address given to the named device matches that of the actual device.
3. Verify that the Network ID given to the named device matches that of the actual device.
4. Increase the Request Timeout setting so that the entire response can be handled.

Device '<device name>' responded with a command format error (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:
Warning

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with a command parameter error (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:
Warning

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with a communications error. [Main, Sub: '<main code, sub code>']

Error Type:
Serious

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with destination node error. [Main, Sub: '<main code, sub code>']

Error Type:
Serious

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with error '<error code>' (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:
Warning

Possible Cause:

Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with error in unit. [Main, Sub: '<main code, sub code>']

Error Type:
Serious

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with Fatal CPU Unit error. [End Code: '<end code>']

Error Type:
Warning

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the end code word bits 6, 7, and 15. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the end code word bits 6, 7, and 15. The description includes corrections.

Device '<device name>' responded with local node error. [Main, Sub: '<main code, sub code>']

Error Type:
Serious

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with Network relay error. [Net Addr, Node Addr: '<network address, node address>'], [Main, Sub: '<main code, sub code>'], [End Code: '<end code>']

Error Type:
Warning

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the end code word bits 6, 7, and 15. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the end code word bits 6, 7, and 15. The description includes corrections.

Device '<device name>' responded with Non-fatal CPU Unit error. [End Code: '<end code>']

Error Type:
Warning

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the end code word bits 6, 7, and 15. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the end code word bits 6, 7, and 15. The description includes corrections.

Device '<device name>' responded with read not possible (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:
Warning

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with routing table error. [Main, Sub: '<main code, sub code>']

Error Type:
Serious

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with write not possible (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:
Warning

Possible Cause:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes possible causes.

Solution:
Refer to **Main and Sub Error Codes** for a description of the Main and Sub Codes. The description includes corrections.

The current project was created with an older version of this driver. Examine device properties to validate network parameters

Error Type:
Warning

Possible Cause:

The current project was created using a version of this driver that checked for network parameter conflicts during runtime. This version only performs runtime checks on parameters that can be changed with system tags. All other parameters are now checked when Device Properties are edited.

Solution:
Examine the properties of each device. Change the value of any property and then change it back to its original value in order to alter the property sheet. Then, click "Accept" or "OK" to validate all of the network parameters associated with the device. Once the devices are validated, save the project.

Note:
This solution will not have to be repeated. Any new devices that are added to the project will be validated automatically.

See Also:
FINS Networks

Unable to bind to adapter: '<adapter>'. Connect failed

Error Type:
Warning

Possible Cause:
Two or more devices in different channels are using the same source IP address and port number.

Solution:
In order to communicate to the same device across multiple OPC server channels, users must do one of the following:

1. Use a unique port number for each channel.
2. Use the same port number but configure each channel to use a unique network adapter and IP.

See Also:
Channel Setup
Multihoming

Unable to write to '<address>' on device '<device name>'

Error Type:
Serious

Possible Cause:
1. The connection between the device and the Host PC is broken.
2. The named device may have been assigned an incorrect IP address.
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 the IP address given to the named device matches that of the actual device.
3. Verify the Network ID given to the named device matches that of the actual device.

Unable to write to register <register address> for device <device name>. The device is in run mode

Error Type:
Warning

Possible Cause:
The device is in Run Mode. CS and TS registers can only be written to when the device is in Programming Mode.

Solution:
Set the device to Programming Mode.

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 Omron FINS 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.

Main and Sub Error Codes

Main Code	Sub-Code	Check Point	Probable Cause	Correction
00: Normal completion	00: Normal completion			
	01: Service canceled		Service was canceled.	Check the capacity of the destination area in the third node.
		Data link status	Service was canceled.	Check the status of the data link.
01: Local node error	01: Local node not in network	Network status of local node	Local node is not participating in the network.	Connect the node to the network.
	02: Token timeout	Maximum node address	Token does not arrive.	Set the local node to within the maximum node address.
	03: Retries failed		Send was not possible during the specified number of retries.	Execute a communications test between the nodes and re-examine the system environment if it fails.
	04: Too many send frames	Number of enabled send frames	Cannot send because maximum number of event frames exceeded.	Check event execution on the network and reduce the number of events per cycle. Increase the maximum number of event frames.
	05: Node address range error	Node address	Node address setting error occurred.	Check the settings of the rotary switches to be sure that the address is within range and that each address is set only once in the same network.
	06: Node address duplication	Node addresses	The same node address has been set twice in the same network.	Change the address of one of the nodes with the same address.

02: Destination node error	01: Destination node not in network	INS indicator on Unit	The destination node is not in the network.	Add the destination node to the network.
	02: Unit missing	Instruction control data	There is no Unit with the specified unit address.	Check the destination unit address.
	03: Third node missing	Instruction control data	The third node does not exist.	Check the unit address of the third node. Check the node address of the third node in the send data for CMND(490).
		Command data	Broadcasting was specified.	Specify only one node for the third node.
	04: Destination node busy		The destination node is busy.	Increase the number of retries or review the system so that the destination node does not receive so many messages.
	05: Response timeout		The message was destroyed by noise.	Increase the number of retries or test communications between nodes to see if there is too much noise.
		Instruction control data	The response monitor time is too short.	Increase the length of the response monitor time.
		Error history	The send / receive frame was discarded.	Take appropriate measures based on the error history.
03: Controller error	01: Commun-ications controller error	Unit / Board indicators	An error occurred in the communications controller.	Take appropriate measures based on the operation manuals for the related Units / Boards.
	02: CPU Unit error	CPU Unit indicators at destination node	A CPU error occurred in the destination CPU Unit.	Clear the error from the CPU Unit based on its operation manuals.
	03: Controller error	Board indicators	A response was not returned because an error occurred in the Board.	Check network communications status and restart the Board. If the problem persists, replace the Board.
	04: Unit number error	Unit number	The unit number was set incorrectly.	Set the rotary switches correctly, being sure the unit numbers are within range and that each number is used only once.
04: Service unsupported	01: Undefined command	Command code	The Unit / Board does not support the specified command code.	Check the command code.
	02: Not supported by model / version	Unit model and version	The command cannot be executed because the model or version is incorrect.	Check the model number and version.
05: Routing table error	01: Destination address setting error	Routing table	The destination network or node address is not set in the routing tables.	Register the destination network and node in the routing tables.
	02: No routing tables	Routing table	Relaying is not possible because there are no routing tables.	Set routing tables in the source node, designation node and relay nodes.
	03: Routing table error	Routing table	There is an error in the routing tables.	Set the routing tables correctly.
	04: Too many relays	Network configuration	An attempt was made to send to a network that was over 3 networks away	Reconstruct the networks or change the routing tables so that commands are sent within a range of 3 networks or less.

10: Command format error	01: Command too long	Command data	The command is longer than the maximum permissible length.	Check the command format and correct the command data.
	02: Command too short	Command data	The command is shorter than the minimum permissible length.	Check the command format and correct the command data.
	03: Elements / data don't match	Command data	The designated number of elements differs from the number of write data items.	Check the number of elements and set data for each element.
	04: Command format error	Command data	An incorrect format was used.	Check the command format and correct the command data.
	05: Header error	Routing table	Either the relay table in the local node or the local network table in the relay node is incorrect.	Set the routing tables correctly.
11: Parameter error	01: Area classification missing	Memory area code in command data	The specified word does not exist in the memory area or there is no EM Area.	Check the memory areas and parameter codes in the command and correct the command data.
	02: Access size error	Access size specification in command data	The access size specification is incorrect or an odd word address is specified.	Check the memory areas and access size and correct the access size.
	03: Address range error	Starting address in command data	The start address in command process is beyond the accessible area.	Check the area being processed and set the correct range.
	04: Address range exceeded	Starting address and number of elements in command data	The end address in command process is beyond the accessible area.	Check the area being processed and set the correct range.
		Data link tables	The total number of words is beyond the limit.	Correct the data link tables.
	06: Program missing	Program number in command data	FFFF Hex was not specified.	Specify FFFF Hex.
	09: Relational error	Command data	A largesmall relationship in the elements in the command data is incorrect.	Check the command data and correct the relationship between the elements.
		Data link table	A node not set in the common link parameters is set as a refresh parameter.	Correct the data link tables.
	0A: Duplicate data access	I/O access in CPU Unit	Differential monitoring was specified during data tracing or data tracing was specified during differential monitoring.	Abort the current process or wait until it ends before executing the command.
		Data link tables	The same node address is specified more than once.	Correct the data link tables.
	0B: Response too long	Number of elements in command data	The response format is longer than the maximum permissible length.	Check the command format and correct the number of elements.
	0C: Parameter error	Parameters in command data	There is an error in one of the parameter settings.	Check the command data and correct the parameters.
		Data link table file	There is an error in the file.	Check the contents of the file.

20: Read not possible	02: Protected		The program area is protected.	Release protection from a Programming Device and then execute the command.
	03: Table missing	Table	A table has not been registered.	Register a table.
			There is an error in the table.	Correct the table.
	04: Data missing		The search data does not exist.	
	05: Program missing	Program number in command data	A non-existing program number has been specified.	Check the program numbers and specify a valid one.
	06: File missing	File name and file device	The file does not exist at the specified file device.	Check the path and file name and correct them.
	07: Data mismatch	Contents of memory being compared	A data being compared is not the same.	Check memory contents and use the correct data.
			A file read operation failed.	Check the contents of the file.
21: Write not possible	01: Read Only		The specified area is Read Only.	If the area is protected using a switch setting, release protection and then execute the command. If the area is permanently Read Only, the command cannot be executed.
	02: Protected Cannot write data link table		The program area is protected.	Release protection from a Programming Device and then execute the command.
		PLC Setup	Writing is not possible because automatic data link table generation has been specified.	Change the PLC Setup so that the data link tables can be manually written.
	03: Cannot register	Number of files in file device	The file cannot be created because the limit has been exceeded.	Delete any unnecessary files or create more file memory.
		Number of files open	The maximum number of files has already been opened for the system limit.	Close one or more files and then execute the command.
	05: Program missing	Program number in command data	A non-existing program number has been specified.	Check the program numbers and specify a valid one.
	06: File missing	File name	The file does not exist at the specified file device.	Correct the file name and then execute the command.
	07: File name already exists	File name	A file with the same name already exists in the specified file device.	Change the name of the file being written and then execute the command.
	08: Cannot change	Contents of memory being changed	The change cannot be made because doing so would create a problem.	

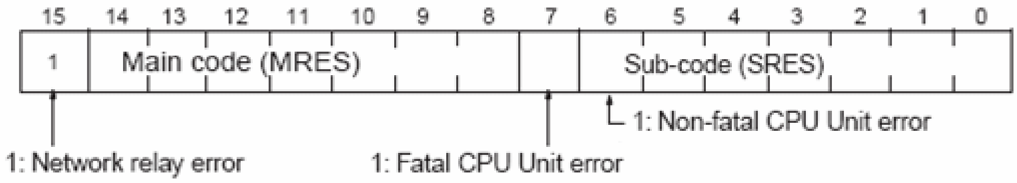
22: Not executable in current mode	01: Not possible during execution		The mode is incorrect.	Check the mode.
		Data link status	The data link is operating.	Check the status of the data links.
	02: Not possible while running		The mode is incorrect.	Check the mode.
		Data link status	The data links are active.	Check the status of the data links.
	03: Wrong PLC mode		The PLC is in PROGRAM mode.	Check the modes of the PLC and computer.
	04: Wrong PLC mode		The PLC is in DEBUG mode.	Check the modes of the PLC and computer.
	05: Wrong PLC mode		The PLC is in MONITOR mode.	Check the modes of the PLC and computer.
	06: Wrong PLC mode		The PLC is in RUN mode.	Check the modes of the PLC and computer.
	07: Specified node not polling node		The specified node is not the polling node.	Check node functioning as the polling node for the network.
	08: Step cannot be executed		The mode is incorrect.	Check step status.
23: No such device	01: File device missing	Unit configuration	The specified memory does not exist as a file device.	Mount memory or format EM as file memory.
	02: Memory missing		There is no file memory.	Check the file memory to see if it is mounted.
	03: Clock missing		There is no clock.	Check the model.
24: Cannot start / stop	01: Table missing	Data link tables	The data link tables have not been registered or they contain an error.	Set the data link tables.

25: Unit error	02: Memory error	Contents of memory being processed	The contents of memory contains an error.	Transfer the correct contents to memory.
	03: I/O setting error	I/O Unit configuration	The registered I/O tables do not agree with the actual I/O configuration.	Correct the I/O tables or the I/O configuration.
	04: Too many I/O points	Number of I/O in registered I/O tables	There are too many I/O points and remote I/O points registered.	Change the registered I/O table so that it is within the limit.
	05: CPU bus error	CPU bus line	An error occurred in data transfer between the CPU and a CPU Bus Unit.	Check Units, Boards and cables to be sure they are connected correctly and then execute the ERROR CLEAR command.
	06: I/O duplication	Rack numbers, Unit numbers and I/O addresses in PLC Setup	The same number / address was set more than once.	Check the PLC Setup and correct the numbers / addresses so that each is used only once.
	07: I/O bus error	I/O bus line	An error occurred in data transfer between the CPU and an I/O Unit.	Check Units, Boards and cables to be sure they are connected correctly and then execute the ERROR CLEAR command.
	09: SYSMAC BUS / 2 error	SYSMAC BUS / 2 transmission path	An error occurred in data transfer on the SYSMAC BUS / 2 line.	Check Units, Boards and cables to be sure they are connected correctly and then execute the ERROR CLEAR command.
	0A: CPU Bus Unit error	CPU Bus Unit transmission path	An error occurred in data transfer for a CPU Bus Unit.	Check Units, Boards and cables to be sure they are connected correctly and then execute the ERROR CLEAR command.
	0D: SYSMAC BUS No. duplication	Word settings	The same word is allocated more than once.	Check the I/O tables and correct the allocations.
	0F: Memory error	Status of memory being processed	A memory error has occurred in internal memory, a memory card, or EM file memory.	For internal memory, write the correct data and then execute the command. For a memory card or EM file memory, the file data has been destroyed. Execute the FILE MEMORY FORMAT command. If the problem persists, replace the memory.
	10: SYSMAC BUS terminator missing		Terminators have not been set.	Set the terminators correctly.

26: Command error	01: No protection	Command protection for program area	The specified area is not protected.	An attempt was made to clear protection on an area that is not protected, i.e., there is no reason to clear protection.
	02: Incorrect password		An incorrect password has been specified.	Specify the correct password.
	04: Protected		The specified area is protected.	Clear protection from a Programming Device and then execute the command.
		Number of commands being executed	The node receiving the command is already processing 5 commands.	Wait for current processing to end or force the end of a current process and then execute the command.
	05: Service already executing		The service is being executed.	Wait for the service to end or force the end of the service and then execute the command.
	06: Service stopped		The service is not being executed.	If necessary, start the service.
	07: No execution right	LNK indicator on Unit / Board	The right to execute the service has not been obtained.	The local node is not in the data link. Execute the command from a node that is participating in the data link.
			A response was not returned because a buffer error occurred.	Restart the Board. If the problem persists, replace the Board.
	08: Settings not complete	Settings required before execution	The settings required before executing the service have not been made.	Make the required settings.
	09: Necessary items not set	Command data	The required elements have not been set in the command data.	Check the command format and set the required elements in the command data.
	0A: Number already defined	Action numbers and transition numbers of program in program area	The specified action / transition number has already been registered in a previous program.	Check the action / transition numbers to ones that are not being used and then execute the command.
	0B: Error will not clear	Cause of error being cleared	The cause of the error has not been removed.	Remove the cause of the error and then execute ERROR CLEAR.
30: Access right error	01: No access right		The access right is held by another device. (Online editing is being executed from another node or ACCESS RIGHT ACQUIRE or ACCESS RIGHT FORCE ACQUIRE has been executed by another node.)	Wait until the access right is released and then execute the command. ACCESS RIGHT ACQUIRE or ACCESS RIGHT FORCE ACQUIRE can be executed to obtain the access right, but this may adversely affect processing by the node that previously held the access right.
40: Abort	01: Service aborted		Service was aborted with ABORT command.	

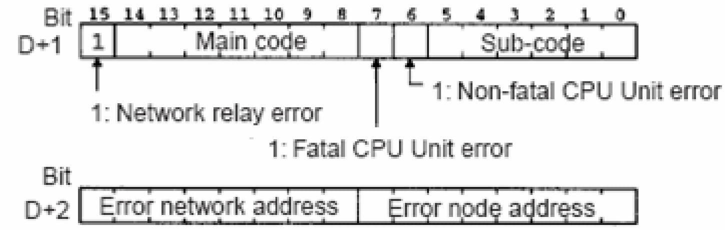
Flags

There are specific flags in the end code word (bits 6, 7, and 15) that may also be ON. The following dialog displays the contents of the end code word.



If bit 6 or 7 is ON, an error has occurred in the destination CPU Unit. If this occurs, refer to the operation manuals for the CPU Unit where the error occurred and then remove the cause of the error.

If bit 15 is ON, an error has occurred during a network relay operation. The end code contains an additional two-byte (one word) network relay error code that can be used to determine the location of the relay error. The following diagram displays the response data of a network relay error. This information can be used to determine the node where the error occurred so that appropriate measures may be taken.



Error network address: 00 to 7F (0 to 127)
Error Node Address

Ethernet:	01 to 7E (1 to 128)
SYSMAC NET:	01 to 7E (1 to 126)
Controller Link:	01 to 20 (1 to 32)
SYSMAC LINK:	01 to 3E (1 to 62)

Index