

# Omron FINS Ethernet Driver

© 2019 PTC Inc. All Rights Reserved.

# Table of Contents

<b>Omron FINS Ethernet Driver</b> .....	<b>1</b>
<b>Table of Contents</b> .....	<b>3</b>
Omron FINS Ethernet Driver .....	5
<b>Overview</b> .....	<b>6</b>
<b>Setup</b> .....	<b>7</b>
Channel Properties — General .....	7
Channel Properties — Ethernet Communications .....	8
Channel Properties — Write Optimizations .....	8
Channel Properties — Communications Parameters .....	9
Channel Properties — Advanced .....	10
Device Properties — General .....	10
Device Properties — Scan Mode .....	12
Device Properties — Timing .....	12
Device Properties — Auto-Demotion .....	13
Device Properties — FINS Network Configuration .....	14
Device Properties — Run Mode Settings .....	15
Device Properties — Communications Parameters .....	16
Device Properties — Redundancy .....	16
FINS Networks .....	17
Multihoming .....	21
<b>Optimizing Communications</b> .....	<b>23</b>
<b>Data Types Description</b> .....	<b>24</b>
<b>Address Descriptions</b> .....	<b>25</b>
C200H Addressing .....	25
C500 Addressing .....	30
C1000H Addressing .....	34
C2000H Addressing .....	38
CV500 Addressing .....	43
CV1000 Addressing .....	48
CV2000 Addressing .....	53
CVM1-CPU01 Addressing .....	59
CVM1-CPU11 Addressing .....	64
CVM1-CPU21 Addressing .....	69
CS1 Addressing .....	74
CJ1 Addressing .....	81

CJ2 Addressing .....	86
<b>Event Log Messages .....</b>	<b>92</b>
Device responded with error.   Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. ....	92
Device responded with local node error.   Main error = <code>, Sub error = <code>. ....	92
Device responded with destination node error.   Main error = <code>, Sub error = <code>. ....	92
Device responded with a communications error.   Main error = <code>, Sub error = <code>. ....	93
Device cannot process command.   Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. ....	93
Device responded with routing table error.   Main error = <code>, Sub error = <code>. ....	93
Device responded with a command format error.   Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. ....	94
Device responded with a command parameter error.   Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. ....	94
Device responded with read not possible.   Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. ....	94
Device responded with error in unit.   Main error = <code>, Sub error = <code>. ....	95
Device cannot accept command.   Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. ....	95
Access right denied.   Tag address = '<address>', Main error = <code>, Sub error = <code>. ....	95
Device responded with write not possible.   Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. ....	95
Device deactivated due to network settings conflict with another device.   Other device = '<device>'. ....	96
The current project was created with an older version of this driver. Examine device properties to validate network parameters. ....	96
Unable to write to tag. The device is in run mode.   Tag address = '<address>'. ....	97
Device responded with network relay error.   Net address = <address>, Node address = <address>, Main error = <code>, Sub error = <code>, End code = <code>. ....	97
Device responded with fatal CPU unit error.   End code = <code>. ....	97
Device responded with non-fatal CPU unit error.   End code = <code>. ....	97
Failed to set device back to Run Mode following write to tag.   Tag address = '<address>'. ....	98
Main and Sub Error Codes .....	99
Remote Network Error .....	107
<b>Index .....</b>	<b>108</b>

## **Omron FINS Ethernet Driver**

---

Help version 1.075

### **CONTENTS**

#### **Overview**

What is the Omron FINS Ethernet Driver?

#### **Setup**

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

#### **Optimizing Communications**

How do I get the best performance from the driver?

#### **Data Types Description**

What data types does this driver support?

#### **Address Descriptions**

How do I address a data location on an Omron FINS Ethernet device?

#### **Event Log Messages**

What messages does the Omron FINS Ethernet Driver produce?

## Overview

---

The Omron FINS Ethernet Driver provides a reliable way to connect Omron FINS Ethernet controller to client applications; including HMI, SCADA, Historian, MES, ERP, and countless custom applications.

## 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 website.

### Communication Protocol

Omron FINS using UDP/IP.

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

The Omron FINS Ethernet Driver can define up to 512 channels.

### Channel Properties — General

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

Property Groups	<input type="checkbox"/> <b>Identification</b>	
General	Name	
Write Optimizations	Description	
Advanced	Driver	
	<input type="checkbox"/> <b>Diagnostics</b>	
	Diagnostics Capture	Disable

### Identification

**Name:** User-defined identity of this channel. In each server project, each channel name must be unique. Although names can be up to 256 characters, some client applications have a limited display window when browsing the OPC server's tag space. The channel name is part of the OPC browser information. The property is required for creating a channel.

**Note:** For information on reserved characters, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in the server help.

**Description:** User-defined information about this channel.

**Note:** Many of these properties, including Description, have an associated system tag.

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

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

## Diagnostics

**Diagnostics Capture:** When enabled, this option makes the channel's diagnostic information available to OPC applications. Because the server's diagnostic features require a minimal amount of overhead processing, it is recommended that they be utilized when needed and disabled when not. The default is disabled.

● **Note:** This property is not available if the driver does not support diagnostics.

● For more information, refer to "Communication Diagnostics" in the server help.

## Channel Properties — Ethernet Communications

Ethernet Communication can be used to communicate with devices.

Property Groups	Ethernet Settings	
General	Network Adapter	Default
<b>Ethernet Communications</b>		
Write Optimizations		
Advanced		

### Ethernet Settings

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

## Channel Properties — Write Optimizations

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

Property Groups	Write Optimizations	
General	Optimization Method	Write Only Latest Value for All Tags
<b>Write Optimizations</b>	Duty Cycle	10

### Write Optimizations

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

- **Write All Values for All Tags:** This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.

- **Write Only Latest Value for Non-Boolean Tags:** Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.
  - **Note:** This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.
- **Write Only Latest Value for All Tags:** This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write queue before they are sent. This is the default mode.

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

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

## Channel Properties — Communications Parameters

Property Groups	☐ <b>Communication Parameters</b>	
General	Port	9600
Ethernet Communications		
Write Optimizations		
Advanced		
<b>Communication Parameters</b>		

**Port:** Specify the port number to 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.

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

- Channel1.device1 uses port 9601.
- Channel2.device2 uses port 9602.
- CX programmer uses port 9600.

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

● **Important:** To communicate to the same device (or multiple physical devices with the same port number configured in the devices) across multiple 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](#).

• **See Also:** [Optimizing Communication](#)

## Channel Properties — Advanced

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

Property Groups	<input type="checkbox"/> <b>Non-Normalized Float Handling</b>	
General	Floating-Point Values	Replace with Zero
Write Optimizations	<input type="checkbox"/> <b>Inter-Device Delay</b>	
<b>Advanced</b>	Inter-Device Delay (ms)	0

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

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

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

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

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

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

## Device Properties — General

Property Groups	Identification	
General	Name	
Scan Mode	Description	
Timing	Channel Assignment	
Auto-Demotion	Driver	
Redundancy	Model	
	ID Format	Decimal
	ID	2
	Operating Mode	
	Data Collection	Enable
	Simulated	No

## Identification

**Name:** User-defined identity of this device.

**Description:** User-defined information about this device.

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

**Driver:** Selected protocol driver for this device.

**Model:** The specific version of the device. *For a list of models that support the FINS Communications Service, refer to the manufacturer's website.*

**ID:** The 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. All parameters specified (except the ID when a relay device is used) refer to the destination device.

## Operating Mode

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

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

### Notes:

1. This System tag (\_Simulated) is read only and cannot be written to for runtime protection. The System tag allows this property to be monitored from the client.

- In Simulation mode, the item's memory map is based on client update rate(s) (Group Update Rate for OPC clients or Scan Rate for native and DDE interfaces). This means that two clients that reference the same item with different update rates return different data.

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

## Device Properties — Scan Mode

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

Property Groups	<input type="checkbox"/> <b>Scan Mode</b>	
General	Scan Mode	Respect Client-Specified Scan Rate ▾
Scan Mode	Initial Updates from Cache	Disable

**Scan Mode:** Specifies how tags in the device are scanned for updates sent to subscribing clients.

Descriptions of the options are:

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

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

## Device Properties — Timing

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

Property Groups	<input type="checkbox"/> <b>Communication Timeouts</b>	
General	Connect Timeout (s)	3
Scan Mode	Request Timeout (ms)	1000
<b>Timing</b>	Attempts Before Timeout	3
Redundancy	<input type="checkbox"/> <b>Timing</b>	
	Inter-Request Delay (ms)	0

## Communications Timeouts

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

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

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

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

## Timing

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

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

## Device Properties — Auto-Demotion

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

Property Groups	<input type="checkbox"/> <b>Auto-Demotion</b>	
General	Demote on Failure	Enable
Scan Mode	Timeouts to Demote	3
Timing	Demotion Period (ms)	10000
<b>Auto-Demotion</b>	Discard Requests when Demoted	Disable

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

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

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

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

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

## Device Properties — FINS Network Configuration

The FINS Network Configuration group establishes how communication is routed in a FINS network using the Omron FINS Ethernet Driver. The default value for network addresses is zero. This is interpreted by FINS as instruction to use the local network; as such, 0 must not be used when data is routed through gateway PLCs. 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, specify the actual network number (1–127) to avoid routing ambiguities.

Property Groups	<input type="checkbox"/> <b>FINS Network Configuration</b>	
General	Source Network Address	0
Scan Mode	Source Node	87
Timing	Destination Network Address	0
Auto-Demotion	Destination Node	25
<b>FINS Network Configuration</b>	Destination Unit	0
Run Mode Settings		
Communication Parameters		
Redundancy		

**Source Network Address:** Specifies the address number of the source network (the computer running the driver). The Source Network Address is sometimes referred to as SNA in Omron FINS documentation. The valid range is 0 to 127. The default setting is 0.

**Source Node:** Specifies the number of the source node (the computer running the driver). The default setting is the last octet of the network adapter specified in channel properties. This parameter specifies the node number of the source device. The Source Node is sometimes referred to as SA1 in Omron FINS documentation. The valid range is 0 to 254.

**Tips:**

- If the destination PLC is configured to use automatic address generation, this number must be the host number portion of the host computer 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, the table must have an entry for the host computer IP. The node number in table entry must agree with the source node for the driver.
- If a channel is duplicated using the Copy and Paste commands, the new channel retains the original channel's FINS network configuration parameters. Unless the Source Node is changed to a unique value, however, correct data transmission cannot be guaranteed.

**Destination Network Address:** Specifies the address number of the destination device. The Destination Network Address is sometimes referred to as DNA in Omron FINS documentation. The valid range is 0 to 127. The default setting is 0.

**Destination Node:** Specifies the node number of the destination device. The Destination Node is sometimes referred to as DA1 in Omron FINS documentation. The valid range is 0 to 254. The default setting is the last octet specified in the device ID. The correct setting value is obtained from the specific Omron Ethernet module being accessed.

• **Note:** The Source Unit (SA2) is assumed to be 0.

• **Tip:** Many Omron Ethernet modules have rotary switches for setting the Destination Node. Generally there are two node number switches: x16 and x1. The switches can be difficult to read due to their small size and because the arrow can appear to fall between numbers. It may be necessary to try several Destination Node numbers to confirm the correct one. For example; on the x16 switch, the arrow may appear between 9 and A; on the x1 switch, the arrow may appear between 2 and 3. The correct Destination Node is one of the following numbers: 163 (for hex A3), 162 (for hex A2), 147 (for hex 93), or 146 (for hex 92).

**Destination Unit:** Specifies the number of the destination device unit number, sometimes referred to as DA2. The valid range is 0 to 255. The default setting is 0. The correct setting value is obtained from the specific Omron Ethernet module being accessed.

• **Note:** Misconfiguration of the network, in combination with some errors, can cause a Remote Network Error.

• 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 result in communications problems. For example, when using the default address of 0 (local network), the Destination Node and Source Node cannot be the same.

• **See Also:** [FINS Networks](#)

## Device Properties — Run Mode Settings

This group specifies the driver's behavior when the device makes writes to Timer Status and Counter Status while in Run Mode

Property Groups	[-] <b>Run Mode Settings</b>	
General	CS and TS Writes	Fail Write, Log Message
Scan Mode		
Timing		
Auto-Demotion		
FINS Network Configuration		
<b>Run Mode Settings</b>		
Communication Parameters		
Redundancy		

**CS and TS Writes:** Select how the device performs writes in Run mode. The default setting is Fail Write, Log Message.

- **Fail Write, Log Message:** This option posts a message to the Event Log when the Write command fails.
- **Set PLC to Monitor Mode, Perform Write:** This option changes the PLC to Monitor Mode before performing the write.
- **Set PLC to Monitor Mode, Write, Reset to Run:** This option changes the PLC to Monitor Mode before performing the write. Once complete, the PLC is reset to Run mode.

## Device Properties — Communications Parameters

Property Groups	[-] <b>Communication Parameters</b>	
General	Request Size (bytes)	512
Scan Mode		
Timing		
Auto-Demotion		
FINS Network Configuration		
Run Mode Settings		
<b>Communication Parameters</b>		
Redundancy		

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

## Device Properties — Redundancy

Property Groups	[-] <b>Redundancy</b>	
General	Secondary Path	...
Scan Mode	Operating Mode	Switch On Failure
Timing	Monitor Item	
<b>Redundancy</b>	Monitor Interval (s)	300
	Return to Primary ASAP	Yes

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

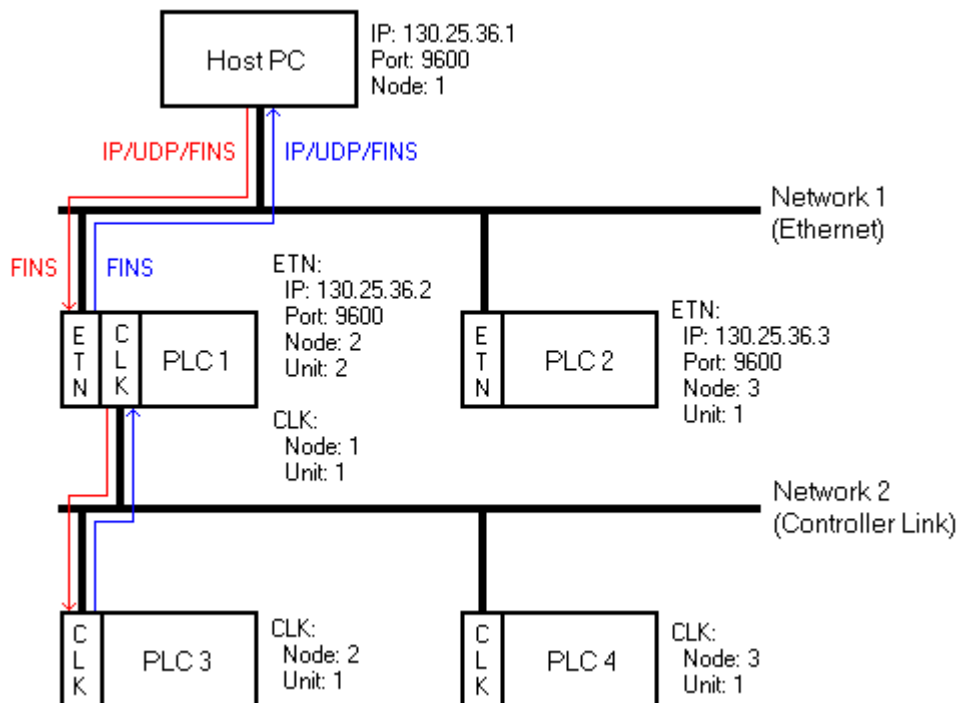
• Consult the website, a sales representative, or the user manual for more information.

## 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 HostLink is not considered a network level.

The diagram below shows a FINS network comprised of interconnected Ethernet and Controller Link networks, and serves 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 always sets DA2 and SA2 to zero; meaning, communication is 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 determines 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,  $130.25.36.2 \text{ AND } 255.255.255.0$  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 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 is taken from the table and used to construct the UDP/IP datagram. If the node number is not found, the destination IP is 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 uses 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 is 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 (SNA):** 1.\*
- **Source Node (SA1):** 1.
- **Destination Network Address (DNA):** 2.\*
- **Destination Node (DA1):** 1.

\*The default value for network addresses is zero. This is interpreted by FINS as instruction 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, specify the actual network number (1–127) to avoid routing ambiguities.

● **Notes:**

1. SA2 and DA2 are automatically set to zero by the driver.
2. Similar device objects would need to be created for PLC 1, 2, and 4.

## 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 can be called because 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, and only 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 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 is actually 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 works fine. Users may encounter the problem when attempting to improve performance by placing the device objects on separate channels. It is important to know that this driver gives 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 needs 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 sends 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.

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 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 is waiting for a response from only one PLC at a time. This is only relevant when multi-level FINS networks are being used.

🔴 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 result in communications problems.

🔵 *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 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 to change (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.

**Notes:**

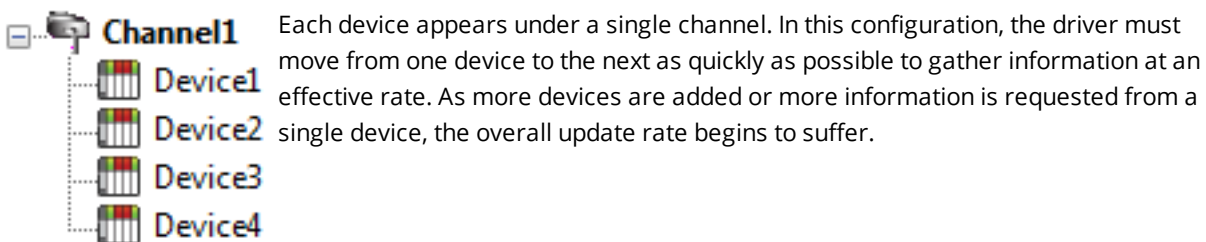
1. Windows NT can add up to five IP addresses for each NIC via the control panel. More IP addresses can be added to the registry manually as necessary. To browse, look under **HEKY\_LOCAL\_MACHINE | SYSTEM | CurrentControlSet | Services**. Then, select the service associated with the adapter card in 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.
2. There is 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 Communications](#)

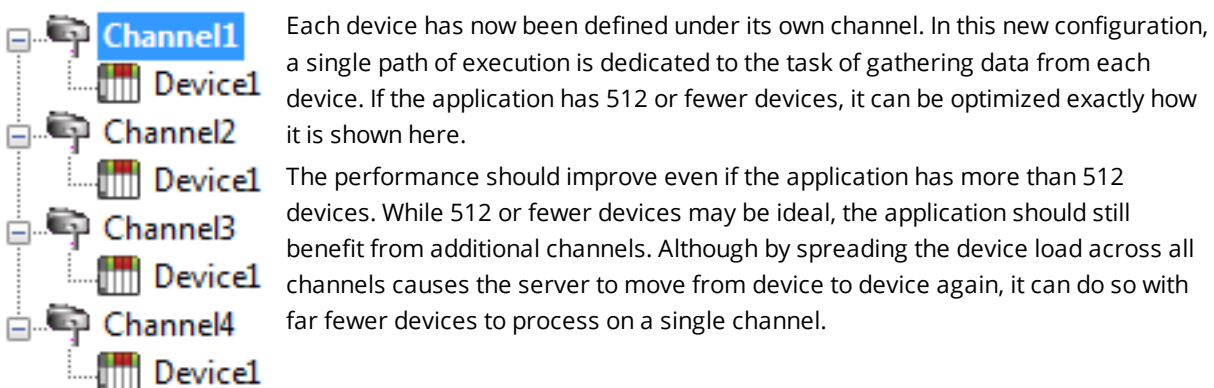
## Optimizing Communications

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. These guidelines that can be used to 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 can be defined under that channel. Each of these devices represents a single Omron FINS Ethernet controller from which data can be collected. While this approach to defining the application provides a high level of performance, it does not take full advantage of the driver or the network. An example of how the application may appear when configured using a single channel is shown below.



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 driver can define up to 512 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.



The Request Size property can affect the 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 larger request size. To refine the performance of this driver, the Request Size may be configured from 32 to 512 bytes for each device.

● If using a multi-levelled FINS network, multiple device objects may need to be created in the server to 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.

## 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	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Auxiliary Relay as String with HiLo Byte Order	AR00.056H-AR27.002H  .l is string length, range 2 to 56 characters	<b>String</b>	Read/Write
Auxiliary Relay as String with LoHi Byte Order	AR00.056L-AR27.002L  .l is string length, range 2 to 56 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Memory	DM0000-DM6655 DM0000-DM6654	<b>Word</b> , Short, BCD Long, DWord, LBCD,	Read/Write



Device Type	Range	Data Type	Access
	DMxxxx.00-DMxxxx.15  • See also <a href="#">String Support</a>	Float <b>Boolean</b>	
Data Memory as String with HiLo Byte Order	DM0000.512H-DM6655.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Data Memory as String with LoHi Byte Order	DM0000.512L-DM6655.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Expansion Data Memory (Current Bank)	EM0000-EM6143 EM0000-EM6142 EMxxxx.00-EMxxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	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 characters	<b>String</b>	Read/Write
Expansion Data Memory (current bank) as String with LoHi Byte Order	EM0000.512L-EM6143.002L  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Expansion Data Memory	EM00:0000-EM07:6143 EM00:0000-EM07:6142 EMx:x.00-EMxx:xxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Expansion Data Memory as String with HiLo Byte	EM00:0000.512H-EM07:6143.002H  .l is string length, range 2 to 512	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
Order	characters		
Expansion Data Memory as String with LoHi Byte Order	EM00:0000.512L-EM07:6143.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Holding Relay	HR00-HR99 HR00-HR98 HRxx.00-HRxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Holding Relay as String with HiLo Byte Order	HR00.200H-HR99.002H .l is string length, range 2 to 200 characters	<b>String</b>	Read/Write
Holding Relay as String with LoHi Byte Order	HR00.200L-HR99.002L .l is string length, range 2 to 200 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Internal Relay	IR000-IR511 IR000-IR510 IRxxx.00-IRxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Internal Relay as a String with HiLo Byte Order	IR000.512H-IR511.002H .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Internal Relay as a String with LoHi Byte Order	IR000.512L-IR511.002L .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Internal Relay as a String	IR000.256D-IR511.001D	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
Using Only the High Order Byte of each Word	.l is string length, range 1 to 256 characters		
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 characters	<b>String</b>	Read/Write
Link Relays	LR00-LR63 LR00-LR62 LRxx.00-LRxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Link Relay as String with HiLo Byte Order	LR00.128H-LR63.002H .l is string length, range 2 to 128 characters	<b>String</b>	Read/Write
Link Relay as String with LoHi Byte Order	LR00.128L-LR63.002L .l is string length, range 2 to 128 characters	<b>String</b>	Read/Write
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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Temporary Relay	TR TR0-TR7	<b>Word</b> , Short <b>Boolean</b>	Read/Write
Timer/Counter	TC000-TC511	<b>BCD</b> , Word, Short	Read/Write
Timer/Counter as String with HiLo Byte Order	TC000.512H-TC511.002H .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Timer/Counter as String with LoHi Byte Order	TC000.512L-TC511.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
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

IROD @ Short = -50

IROD @ Word = 8050 (MSB set)

IRO @ BCD = 8050

IROD @ Short = 50

IROD @ Word = 50

IRO @ 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 contains 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 [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 and String. 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.

● 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. DWords DM0 and DM1 overlap at word DM1. Writing to DM0 also modifies 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  ● See also <a href="#">String Support</a>	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float, <b>Boolean</b>	Read/Write
Data Memory as String with HiLo Byte Order	DM000.512H-DM511.002H  .l is string length, range 2 to 512	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
	characters		
Data Memory as String with LoHi Byte Order	DM000.512L-DM511.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Holding Relay	HR00-HR31 HR00-HR30 HRxx.00-HRxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float, <b>Boolean</b>	Read/Write
Holding Relay as String with HiLo Byte Order	HR00.064H-HR31.002H .l is string length, range 2 to 64 characters	<b>String</b>	Read/Write
Holding Relay as String with LoHi Byte Order	HR00.064L-HR31.002L .l is string length, range 2 to 64 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Internal Relay	IR00-IR63 IR00-IR62 IRxx.00-IRxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float, <b>Boolean</b>	Read/Write
Internal Relay as a String with HiLo Byte Order	IR00.128H-IR63.002H .l is string length, range 2 to 128 characters	<b>String</b>	Read/Write
Internal Relay as a String with LoHi Byte Order	IR00.128L-IR63.002L .l is string length, range 2 to 128 characters	<b>String</b>	Read/Write
Internal Relay as a String	IR00.064D-IR63.001D	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
Using Only the High Order Byte of each Word	.l is string length, range 1 to 64 characters		
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 characters	<b>String</b>	Read/Write
Link Relays	LR00-LR31 LR00-LR30 LRxx.00-LRxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float, <b>Boolean</b>	Read/Write
Link Relays as String with HiLo Byte Order	LR00.064H-LR31.002H .l is string length, range 2 to 64 characters	<b>String</b>	Read/Write
Link Relays as String with LoHi Byte Order	LR00.064L-LR31.002L .l is string length, range 2 to 64 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Temporary Relay	TR TR0-TR7	<b>Word</b> , Short, <b>Boolean</b>	Read/Write
Timer/Counter	TC000-TC127	<b>BCD</b> , Word, Short	Read/Write
Timer/Counter as String with HiLo Byte Order	TC000.256H-TC127.002H .l is string length, range 2 to 256 characters	<b>String</b>	Read/Write
Timer/Counter as String with LoHi Byte Order	TC000.256L-TC127.002L .l is string length, range 2 to 256 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
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

IROD @ Short = -50

IROD @ Word = 8050 (MSB set)

IRO @ BCD = 8050

IROD @ Short = 50

IROD @ Word = 50

IRO @ 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 contains 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 [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 and String. 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.


⚠ 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. DWords DM0 and DM1 overlap at word DM1. Writing to DM0 also modifies 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, 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	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Auxiliary Relay as String with HiLo Byte Order	AR00.056H-AR27.002H .l is string length, range 2 to 56 characters	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
Auxiliary Relay as String with LoHi Byte Order	AR00.056L-AR27.002L .l is string length, range 2 to 56 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Memory	DM0000-DM4095 DM0000-DM4094 DMxxxx.00-DMxxxx.15  See also <a href="#">String Support</a>	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Data Memory as String with HiLo Byte Order	DM0000.512H-DM4095.002H .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Data Memory as String with LoHi Byte Order	DM0000.512L-DM4095.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Holding Relay	HR00-HR99 HR00-HR98 HRxx.00-HRxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Holding Relay as String with HiLo Byte Order	HR00.200H-HR99.002H .l is string length, range 2 to 200 characters	<b>String</b>	Read/Write
Holding Relay as String with LoHi Byte Order	HR00.200L-HR99.002L .l is string length, range 2 to 200 characters	<b>String</b>	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 characters	<b>String</b>	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	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
	to 100 characters		
Internal Relay	IR000-IR255 IR000-IR254 IRxxx.00-IRxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Internal Relay as a String with HiLo Byte Order	IR000.512H-IR255.002H .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Internal Relay as a String with LoHi Byte Order	IR000.512L-IR255.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Link Relays	LR00-LR63 LR00-LR62 LRxx.00-LRxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float, <b>Boolean</b>	Read/Write
Link Relays as String with HiLo Byte Order	LR00.128H-LR63.002H .l is string length, range 2 to 128 characters	<b>String</b>	Read/Write
Link Relays as String with LoHi Byte Order	LR00.128L-LR63.002L .l is string length, range 2 to 128 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Temporary Relay	TR TR0-TR7	<b>Word</b> , Short, <b>Boolean</b>	Read/Write
Timer/Counter	TC000-TC511	<b>BCD</b> , Word, Short	Read/Write
Timer/Counter as String with HiLo Byte Order	TC000.512H-TC511.002H .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Timer/Counter as String with LoHi Byte Order	TC000.512L-TC511.002L .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
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 characters	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 characters	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:

IROD @ Short = -50  
IROD @ Word = 8050 (MSB set)  
IRO @ BCD = 8050

IROD @ Short = 50  
IROD @ Word = 50  
IRO @ 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 contains 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 [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 and String. 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.

🔴 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. DWords DM0 and DM1 overlap at word DM1. Writing to DM0 also modifies 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, 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	<b>Word</b> , Short, BCD, Long, DWord,	Read/Write

Device Type	Range	Data Type	Access
		LBCD, Float, <b>Boolean</b>	
Auxiliary Relay as String with HiLo Byte Order	AR00.056H-AR27.002H  .l is string length, range 2 to 56 characters	<b>String</b>	Read/Write
Auxiliary Relay as String with LoHi Byte Order	AR00.056L-AR27.002L  .l is string length, range 2 to 56 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Memory	DM0000-DM6655 DM0000-DM6654 DMxxxx.00-DMxxxx.15   See also <a href="#">String Support</a>	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float, <b>Boolean</b>	Read/Write
Data Memory as String with HiLo Byte Order	DM0000.512H-DM6655.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Data Memory as String with LoHi Byte Order	DM0000.512L-DM6655.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Holding Relay	HR00-HR99 HR00-HR98 HRxx.00-HRxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float, <b>Boolean</b>	Read/Write
Holding Relay as String with HiLo Byte Order	HR00.200H-HR99.002H  .l is string length, range 2 to 200	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
	characters		
Holding Relay as String with LoHi Byte Order	HR00.200L-HR99.002L .l is string length, range 2 to 200 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Internal Relay	IR000-IR255 IR000-IR254 IRxxx.00-IRxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Internal Relay as a String with HiLo Byte Order	IR000.512H-IR255.002H .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Internal Relay as a String with LoHi Byte Order	IR000.512L-IR255.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Link Relays	LR00-LR63 LR00-LR62 LRxx.00-LRxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float, <b>Boolean</b>	Read/Write
Link Relays as String with HiLo Byte Order	LR00.128H-LR63.002H .l is string length, range 2 to 128 characters	<b>String</b>	Read/Write
Link Relays as String with LoHi Byte Order	LR00.128L-LR63.002L .l is string length, range 2 to 128 characters	<b>String</b>	Read/Write
Link Relays as String	LR00.064D-LR63.001D	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
Using Only the High Order Byte of each Word	.l is string length, range 1 to 64 characters		
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 characters	<b>String</b>	Read/Write
Temporary Relay	TR TR0-TR7	<b>Word, Short, Boolean</b>	Read/Write
Timer/Counter	TC000-TC511	<b>BCD, Word, Short</b>	Read/Write
Timer/Counter as String with HiLo Byte Order	TC000.512H-TC511.002H .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Timer/Counter as String with LoHi Byte Order	TC000.512L-TC511.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Timer/Counter Status	TS000-TS511	<b>Boolean</b>	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**

IROD @ Short = -50  
 IROD @ Word = 8050 (MSB set)  
 IRO @ BCD = 8050

IROD @ Short = 50  
 IROD @ Word = 50  
 IRO @ 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 contains 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 [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 and String. 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.

⚠ 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. DWords DM0 and DM1 overlap at word DM1. Writing to DM0 also modifies 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, 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	<b>Boolean</b>	Read Only
Auxiliary Relay	A000-A255	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read/Write
	A000-A254		Read Only
	A256-A511	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read/Write
	A256-A510		Read Only
Auxiliary Relay as String with HiLo Byte Order	A000.00-A000.15-A255.00- A255.15	<b>Boolean</b>	Read Only
	A256.00-A256.15-A511.00- A511.15	<b>Boolean</b>	Read Only
Auxiliary Relay as String with HiLo Byte Order	A000.512H-A255.002H	<b>String</b>	Read/Write
	A256.512H-A511.002H	<b>String</b>	Read Only
Auxiliary Relay as String with LoHi Byte Order	.l is string length, range 2 to 512 characters		
	A000.512L-A255.002L	<b>String</b>	Read/Write
Auxiliary Relay as String Using Only the High Order Byte of each Word	A256.512L-A511.002L	<b>String</b>	Read Only
	.l is string length, range 2 to 512 characters		
Auxiliary Relay as String Using Only the High Order Byte of each Word	A000.256D-A255.001D	<b>String</b>	Read/Write
	A256.256D-A511.001D	<b>String</b>	Read Only
Auxiliary Relay as String Using Only the Low Order Byte of each Word	.l is string length, range 1 to 256 characters		
	A000.256E-A255.001E	<b>String</b>	Read/Write
Auxiliary Relay as String Using Only the Low Order Byte of each Word	A256.256E-A511.001E	<b>String</b>	Read Only
	.l is string length, range 1 to 256 characters		

Device Type	Range	Data Type	Access
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxxx.00-CIOxxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
CIO Memory as String with HiLo Byte Order	CIO0000.512H-CIO2555.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
CIO Memory as String with LoHi Byte Order	CIO0000.512L-CIO2555.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter	C000-C511	<b>BCD</b> , Word, Short	Read/Write
Counter as String with HiLo Byte Order	C000.512H-C511.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Counter as String with LoHi Byte Order	C000.512L-C511.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter Status	CS000-CS511	<b>Boolean</b>	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
CPU Bus Link as String with HiLo Byte Order	G000.512H-G255.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
CPU Bus Link as String with LoHi Byte Order	G000.512L-G255.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Memory	D0000-D8191 D0000-D8190 Dxxxx.00-Dxxxx.15   See also <a href="#">String Support</a>	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Data Memory as String with HiLo Byte Order	D0000.512H-D8191.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Data Memory as String with LoHi Byte Order	D0000.512L-D8191.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Register	DR0-DR2	<b>Word</b> , Short, BCD*	Read/Write
Index Register	IR0-IR2	<b>Word</b> , Short, BCD*	Read/Write
Step Timer	ST000-ST511	<b>Word</b> , Short, BCD*	Read/Write
Step Timer Status	STS000-STS511	<b>Boolean</b>	Read/Write
Temporary Relay	TR TR0-TR7	<b>Word</b> , Short <b>Boolean</b>	Read/Write
Timer	T000-T511	<b>BCD</b> , Word, Short	Read/Write
Timer as String with HiLo Byte Order	T000.512H-T511.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Timer as String	T000.512L-T511.002L	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
with LoHi Byte Order	.l is string length, range 2 to 512 characters		
Timer as String Using Only the High Order Byte of each Word	T000.256D-T511.001D .l is string length, range 1 to 256 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Timer Status	TS000-TS511	<b>Boolean</b>	Read/Write
Transition Flag	TN000-TN511	<b>Boolean</b>	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 contains 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 [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, Step Timer, and String. 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.

🔴 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. DWords DM0 and DM1 overlap at word DM1. Writing to DM0 also modifies 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, 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	<b>Boolean</b>	Read Only
Auxiliary Relay	A000-A255	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read/Write
	A000-A254		Read Only
	A256-A511	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read/Write
	A256-A510		Read Only
Auxiliary Relay as String with HiLo Byte Order	A000.512H-A255.002H	<b>String</b>	Read/Write
	A256.512H-A511.002H	<b>String</b>	Read Only
.l is string length, range 2 to 512 characters			
Auxiliary Relay as String with LoHi Byte Order	A000.512L-A255.002L	<b>String</b>	Read/Write
	A256.512L-A511.002L	<b>String</b>	Read Only
.l is string length, range 2 to 512 characters			
Auxiliary Relay as String Using Only the High Order Byte of each Word	A000.256D-A255.001D	<b>String</b>	Read/Write
	A256.256D-A511.001D	<b>String</b>	Read Only
.l is string length, range 1 to 256 characters			
Auxiliary Relay as String Using Only the Low Order Byte of each Word	A000.256E-A255.001E	<b>String</b>	Read/Write
	A256.256E-A511.001E	<b>String</b>	Read Only
.l is string length, range 1 to 256 characters			
CIO	CIO0000-CIO2555	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read/Write
	CIO0000-CIO2554		
	CIOxxxx.00-CIOxxxx.15	<b>Boolean</b>	
CIO Memory as String with HiLo Byte Order	CIO0000.512H-CIO2555.002H	<b>String</b>	Read/Write
.l is string length, range 2 to			

Device Type	Range	Data Type	Access
	512 characters		
CIO Memory as String with LoHi Byte Order	CIO0000.512L-CIO2555.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter	C0000-C1023	<b>BCD, Word, Short</b>	Read/Write
Counter as String with HiLo Byte Order	C0000.512H-C1023.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Counter as String with LoHi Byte Order	C0000.512L-C1023.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter Status	CS0000-CS1023	<b>Boolean</b>	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	<b>Word, Short, BCD Long, DWord, LBCD, Float Boolean</b>	Read/Write
CPU Bus Link as String with HiLo Byte Order	G000.512H-G255.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
CPU Bus Link as String with LoHi Byte Order	G000.512L-G255.002L  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
CPU Bus Link as String	G000.256D-G255.001D	<b>String</b>	Read/Write



Device Type	Range	Data Type	Access
Using Only the High Order Byte of each Word	.l is string length, range 1 to 256 characters		
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 characters	<b>String</b>	Read/Write
Data Memory	D00000-D24575 D00000-D24574 Dxxxxx.00-Dxxxxx.15  • See also <a href="#">String Support</a>	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Data Memory as String with HiLo Byte Order	D00000.512H-D24575.002H .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Data Memory as String with LoHi Byte Order	D00000.512L-D24575.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Register	DR0-DR2	<b>Word</b> , Short, BCD*	Read/Write
Expansion Data Memory (current bank)	E00000-E32765 E00000-E32764 Exxxxx.00-Exxxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
	256 characters		
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 characters	<b>String</b>	Read/Write
Expansion Data Memory	E00:00000-E07:32765 E00:00000-E07:32764 Ex:x.00-Exx:xxxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Index Register	IR0-IR2	<b>Word</b> , Short, BCD*	Read/Write
Step Timer	ST0000-ST1023	<b>Word</b> , Short, BCD*	Read/Write
Step Timer Status	STS0000-STS1023	<b>Boolean</b>	Read/Write
Temporary Relay	TR TR0-TR7	<b>Word</b> , Short <b>Boolean</b>	Read/Write
Timer	T0000-T1023	<b>BCD</b> , Word, Short	Read/Write
Timer as String with HiLo Byte Order	T0000.512H-T1023.002H .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Timer as String with LoHi Byte Order	T0000.512L-T1023.002L .l is string length, range 2 to	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
	512 characters		
Timer as String Using Only the High Order Byte of each Word	T0000.256D-T1023.001D .l is string length, range 1 to 256 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Timer Status	TS0000-TS1023	<b>Boolean</b>	Read/Write
Transition Flag	TN0000-TN1023	<b>Boolean</b>	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

IROD @ Short = -50

IROD @ Word = 8050 (MSB set)

IRO @ BCD = 8050

IROD @ Short = 50

IROD @ Word = 50

IRO @ 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 contains 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 [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, Step Timer, and String. 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.

🔴 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. DWords D0 and D1 overlap at word D1. Writing to D0 also modifies 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, 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	<b>Boolean</b>	Read Only
Auxiliary Relay	A000-A255	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read/Write
	A000-A254		Read Only
	A256-A511	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read/Write
	A256-A510		Read Only
Auxiliary Relay as String with HiLo Byte Order	A000.512H-A255.002H	<b>String</b>	Read/Write
	A256.512H-A511.002H	<b>String</b>	Read Only
Auxiliary Relay as String with LoHi Byte Order	A000.512L-A255.002L	<b>String</b>	Read/Write
	A256.512L-A511.002L	<b>String</b>	Read Only
Auxiliary Relay as String Using Only the High Order Byte of each Word	A000.256D-A255.001D	<b>String</b>	Read/Write
	A256.256D-A511.001D	<b>String</b>	Read Only
Auxiliary Relay as String Using Only the Low Order Byte of each Word	A000.256E-A255.001E	<b>String</b>	Read/Write
	A256.256E-A511.001E	<b>String</b>	Read Only
CIO	CIO0000-CIO2555	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read/Write
	CIO0000-CIO2554		
	CIOxxxx.00-CIOxxxx.15	<b>Boolean</b>	
CIO Memory as String with HiLo Byte Order	CIO0000.512H-CIO2555.002H	<b>String</b>	Read/Write
CIO Memory as String with LoHi Byte Order	CIO0000.512L-CIO2555.002L	<b>String</b>	Read/Write

<b>Device Type</b>	<b>Range</b>	<b>Data Type</b>	<b>Access</b>
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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter	C0000-C1023	<b>BCD</b> , Word, Short	Read/Write
Counter as String with HiLo Byte Order	C0000.512H-C1023.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Counter as String with LoHi Byte Order	C0000.512L-C1023.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter Status	CS0000-CS1023	<b>Boolean</b>	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
CPU Bus Link as String with HiLo Byte Order	G000.512H-G255.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
CPU Bus Link as String with LoHi Byte Order	G000.512L-G255.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
Data Memory	D00000-D24575 D00000-D24574 Dxxxx.00-Dxxxx.15  • See also <a href="#">String Support</a>	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Data Memory as String with HiLo Byte Order	D00000.512H-D24575.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Data Memory as String with LoHi Byte Order	D00000.512L-D24575.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Register	DR0-DR2	<b>Word</b> , Short, BCD*	Read/Write
Expansion Data Memory (current bank)	E00000-E32765 E00000-E32764 Exxxxx.00-Exxxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Expansion Data Memory	E00:00000-E07:32765 E00:00000-E07:32764 Ex:x.00-Exx:xxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read/Write

Device Type	Range	Data Type	Access
		<b>Boolean</b>	
Expansion Data Memory as String with HiLo Byte Order	E00:00000.512H - E07:32765.002H  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Index Register	IR0-IR2	<b>Word, Short, BCD*</b>	Read/Write
Step Timer	ST0000-ST1023	<b>Word, Short, BCD*</b>	Read/Write
Step Timer Status	STS0000-ST51023	<b>Boolean</b>	Read/Write
Temporary Relay	TR TR0-TR7	<b>Word, Short Boolean</b>	Read/Write
Timer	T0000-T1023	<b>BCD, Word, Short</b>	Read/Write
Timer as String with HiLo Byte Order	T0000.512H-T1023.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Timer as String with LoHi Byte Order	T0000.512L-T1023.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write



Device Type	Range	Data Type	Access
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

IROD @ Short = -50

IROD @ Word = 8050 (MSB set)

IRO @ BCD = 8050

IROD @ Short = 50

IROD @ Word = 50

IRO @ 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 contains 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 [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, Step Timer, and String. 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.

● 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. DWords D0 and D1 overlap at word D1. Writing to D0 also modifies 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, 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	<b>Word</b> , Short, BCD	Read/Write
	A000-A254	Long, DWord, LBCD, Float	
	A256-A511	<b>Word</b> , Short, BCD	Read Only
	A256-A510		

Device Type	Range	Data Type	Access
	A000.00-A000.15-A255.00-A255.15  A256.00-A256.15-A511.00-A511.15	Long, DWord, LBCD, Float  <b>Boolean</b>  <b>Boolean</b>	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 characters	<b>String</b> <b>String</b>	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 characters	<b>String</b> <b>String</b>	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 characters	<b>String</b> <b>String</b>	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 characters	<b>String</b> <b>String</b>	Read/Write  Read Only
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxxx.00-CIOxxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
CIO Memory as String with HiLo Byte Order	CIO0000.512H-CIO2555.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
CIO Memory as String with LoHi Byte Order	CIO0000.512L-CIO2555.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter	C000-C511	<b>BCD</b> , Word, Short	Read/Write

Device Type	Range	Data Type	Access
Counter as String with HiLo Byte Order	C000.512H-C511.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Counter as String with LoHi Byte Order	C000.512L-C511.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter Status	CS000-CS511	<b>Boolean</b>	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
CPU Bus Link as String with HiLo Byte Order	G000.512H-G255.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
CPU Bus Link as String with LoHi Byte Order	G000.512L-G255.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Memory	D0000-D8191 D0000-D8190 Dxxxx.00-Dxxxx.15   See also <a href="#">String Support</a>	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Data Memory as String with HiLo Byte Order	D0000.512H-D8191.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
Data Memory as String with LoHi Byte Order	D0000.512L- D8191.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Register	DR0-DR2	<b>Word</b> , Short, BCD*	Read/Write
Index Register	IR0-IR2	<b>Word</b> , Short, BCD*	Read/Write
Temporary Relay	TR TR0-TR7	<b>Word</b> , Short <b>Boolean</b>	Read/Write
Timer	T000-T511	<b>BCD</b> , Word, Short	Read/Write
Timer as String with HiLo Byte Order	T000.512H-T511.002H .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Timer as String with LoHi Byte Order	T000.512L-T511.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Timer Status	TS000-TS511	<b>Boolean</b>	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

IROD @ Short = -50  
IROD @ Word = 8050 (MSB set)  
IRO @ BCD = 8050

IROD @ Short = 50  
IROD @ Word = 50  
IRO @ 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 contains 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 [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 String. 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.

● 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. DWords DM0 and DM1 overlap at word DM1. Writing to DM0 also modifies 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, 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	<b>Word</b> , Short, BCD	Read/Write
	A000-A254	Long, DWord, LBCD, Float	
	A256-A511	<b>Word</b> , Short, BCD	Read Only
	A256-A510	Long, DWord, LBCD, Float	
Auxiliary Relay as String with HiLo Byte Order	A000.00-A000.15-A255.00-A255.15	<b>Boolean</b>	Read/Write
	A256.00-A256.15-A511.00-A511.15	<b>Boolean</b>	Read Only
Auxiliary Relay as String with LoHi Byte Order	A000.512H-A255.002H	<b>String</b>	Read/Write
	A256.512H-A511.002H	<b>String</b>	Read Only
	.l is string length, range 2 to 512 characters		
Auxiliary Relay as String with LoHi Byte Order	A000.512L-A255.002L	<b>String</b>	Read/Write
	A256.512L-A511.002L	<b>String</b>	

Device Type	Range	Data Type	Access
	.l is string length, range 2 to 512 characters		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 characters	<b>String</b> <b>String</b>	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 characters	<b>String</b> <b>String</b>	Read/Write  Read Only
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxxx.00-CIOxxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
CIO Memory as String with HiLo Byte Order	CIO0000.512H-CIO2555.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
CIO Memory as String with LoHi Byte Order	CIO0000.512L-CIO2555.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter	C0000-C1023	<b>BCD</b> , Word, Short	Read/Write
Counter as String with HiLo Byte Order	C0000.512H-C1023.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Counter as String with LoHi Byte Order	C0000.512L-C1023.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write



Device Type	Range	Data Type	Access
Counter as String Using Only the Low Order Byte of each Word	C0000.256E-C1023.001E  .l is string length, range 1 to 256 characters	<b>String</b>	Read/Write
Counter Status	CS0000-CS1023	<b>Boolean</b>	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
CPU Bus Link as String with HiLo Byte Order	G000.512H-G255.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
CPU Bus Link as String with LoHi Byte Order	G000.512L-G255.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Memory	D00000-D24575 D00000-D24574 Dxxxxx.00-Dxxxxx.15   See also <a href="#">String Support</a>	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Data Memory as String with HiLo Byte Order	D00000.512H-D24575.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Data Memory as String with LoHi Byte Order	D00000.512L-D24575.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
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 characters	String	Read/Write
Timer as String with LoHi Byte Order	T0000.512L-T1023.002L  .l is string length, range 2 to 512 characters	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 characters	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 characters	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

IROD @ Short = -50  
 IROD @ Word = 8050 (MSB set)  
 IRO @ BCD = 8050

IROD @ Short = 50  
 IROD @ Word = 50  
 IRO @ 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 contains 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 [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 String. 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.

● 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. DWords DM0 and DM1 overlap at word DM1. Writing to DM0 also modifies 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, 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	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float	Read/Write
	A256-A511 A256-A510	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float	Read Only
	A000.00-A000.15-A255.00- A255.15	<b>Boolean</b>	Read/Write
	A256.00-A256.15-A511.00- A511.15	<b>Boolean</b>	Read Only
Auxiliary Relay as String with HiLo Byte Order	A000.512H-A255.002H A256.512H-A511.002H	<b>String</b> <b>String</b>	Read/Write Read Only
	.l is string length, range 2 to 512 characters		
Auxiliary Relay as String with LoHi Byte Order	A000.512L-A255.002L A256.512L-A511.002L	<b>String</b> <b>String</b>	Read/Write Read Only
	.l is string length, range 2 to 512 characters		
Auxiliary Relay as String Using Only the High Order Byte of each Word	A000.256D-A255.001D A256.256D-A511.001D	<b>String</b> <b>String</b>	Read/Write Read Only
	.l is string length, range 1 to 256 characters		
Auxiliary Relay as String Using Only the Low Order Byte of each Word	A000.256E-A255.001E A256.256E-A511.001E	<b>String</b> <b>String</b>	Read/Write Read Only
	.l is string length, range 1 to 256 characters		
CIO	CIO0000-CIO2555 CIO0000-CIO2554	<b>Word</b> , Short, BCD,	Read/Write

Device Type	Range	Data Type	Access
	CIOxxxx.00-CIOxxxx.15	Long, DWord, LBCD, Float <b>Boolean</b>	
CIO Memory as String with HiLo Byte Order	CIO0000.512H-CIO2555.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
CIO Memory as String with LoHi Byte Order	CIO0000.512L-CIO2555.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter	C0000-C1023	<b>BCD, Word, Short</b>	Read/Write
Counter as String with HiLo Byte Order	C0000.512H-C1023.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Counter as String with LoHi Byte Order	C0000.512L-C1023.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter Status	CS0000-CS1023	<b>Boolean</b>	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	<b>Word, Short, BCD,</b> Long, DWord, LBCD, Float, <b>Boolean</b>	Read/Write
CPU Bus Link as String with HiLo Byte Order	G000.512H-G255.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
CPU Bus Link as String	G000.512L-G255.002L	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
with LoHi Byte Order	.l is string length, range 2 to 512 characters		
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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Memory	D00000-D24575 D00000-D24574 Dxxxxx.00-Dxxxxx.15  • See also <a href="#">String Support</a>	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Data Memory as String with HiLo Byte Order	D00000.512H-D24575.002H .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Data Memory as String with LoHi Byte Order	D00000.512L-D24575.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Register	DR0-DR2	<b>Word</b> , Short, BCD*	Read/Write
Expansion Data Memory (current bank)	E00000-E32765 E00000-E32764 Exxxxx.00-Exxxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float <b>Boolean</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Expansion Data Memory	E00:00000-E07:32765 E00:00000-E07:32764 Ex:x.00-Exx:xxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float <b>Boolean</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Index Register	IR0-IR2	<b>Word</b> , Short, BCD*	Read/Write
Temporary Relay	TR TR0-TR7	<b>Word</b> , Short, <b>Boolean</b>	Read/Write
Timer	T0000-T1023	<b>BCD</b> , Word, Short	Read/Write
Timer as String with HiLo Byte Order	T0000.512H-T1023.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Timer as String with LoHi Byte Order	T0000.512L-T1023.002L  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Timer as String	T0000.256D-T1023.001D	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
Using Only the High Order Byte of each Word	.l is string length, range 1 to 256 characters		
Timer as String Using Only the Low Order Byte of each Word	T0000.256E-T1023.001E .l is string length, range 1 to 256 characters	<b>String</b>	Read/Write
Timer Status	TS0000-TS1023	<b>Boolean</b>	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

IROD @ Short = -50

IROD @ Word = 8050 (MSB set)

IRO @ BCD = 8050

IROD @ Short = 50

IROD @ Word = 50

IRO @ 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 contains 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 [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 String. 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.

⚠ 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. DWords DM0 and DM1 overlap at word DM1. Writing to DM0 also modifies 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, 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	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float	Read Only
	A448-A959 A448-A958	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float	Read/Write
	A000.00-A000.15-A447.00- A447.15	<b>Boolean</b>	Read Only
	A448.00-A448.15-A959.00- A959.15	<b>Boolean</b>	Read/Write
Auxiliary Relay as String with HiLo Byte Order	A000.512H-A447.002H A448.512H-A959.002H	<b>String</b> <b>String</b>	Read Only Read/Write
	.l is string length, range 2 to 512 characters		
Auxiliary Relay as String with LoHi Byte Order	A000.512L-A447.002L A448.512L-A959.002L	<b>String</b> <b>String</b>	Read Only Read/Write
	.l is string length, range 2 to 512 characters		
Auxiliary Relay as String using only the High Order Byte of each Word	A000.256D-A447.001D A448.256D-A959.001D	<b>String</b> <b>String</b>	Read Only Read/Write
	.l is string length, range 1 to 256 characters		
Auxiliary Relay as String using only the Low Order Byte of each Word	A000.256E-A447.001E A448.256E-A959.001E	<b>String</b> <b>String</b>	Read Only Read/Write
	.l is string length, range 1 to 256 characters		
CIO	CIO0000-CIO6143 CIO0000-CIO6142 CIOxxxx.00-CIOxxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float, <b>Boolean</b>	Read/Write
	CIO memory as String with HiLo Byte Order	CIO0000.512H- CIO6143.002H	<b>String</b>
CIO memory as String with LoHi Byte Order	CIO0000.512L- CIO6143.002L	<b>String</b>	Read/Write
	.l is string length, range 2 to 512 characters		
CIO memory as String Using Only the High Order Byte of each Word	CIO0000.256D- CIO6143.001D	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
	.l is string length, range 1 to 256 characters		
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 characters	<b>String</b>	Read/Write
Counter	C0000-C4095	<b>BCD, Word, Short</b>	Read/Write
Counter as String with HiLo Byte Order	C0000.512H-C4095.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Counter as String with LoHi Byte Order	C0000.512L-C4095.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter Status	CS0000-CS4095	<b>Boolean</b>	Read/Write*
Data Memory	D00000-D32767 D00000-D32766 Dxxxxx.00-Dxxxxx.15   See also <a href="#">String Support</a>	<b>Word, Short, BCD, Long, DWord, LBCD, Float, Boolean</b>	Read/Write
Data Memory as String with HiLo Byte Order	D00000.512H-D32767.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Data Memory as String with LoHi Byte Order	D00000.512L-D32767.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Memory as String Using Only the Low Order	D00000.256E-D32767.001E	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
Byte of each Word	.l is string length, range 1 to 256 characters		
Data Register	DR00-DR15 DR00-DR14	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float	Read/Write*
Expansion Data Memory (current bank)	E00000-E32767 E00000-E32766 Exxxx.00-Exxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float, <b>Boolean</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Expansion Data Memory	E00:00000-E12:32767 E00:00000-E12:32766 Ex:x.00-Exx:xxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float <b>Boolean</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Expansion Data Memory as String Using Only the Low	E00:00000.256E-E12:32767.001E	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
Order Byte of each Word	.l is string length, range 1 to 256 characters		
Holding Relay	H0000-H1535 H0000-H1534 Hxxxx.00-Hxxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float, <b>Boolean</b>	Read/Write
Holding Relay as String with HiLo Byte Order	H0000.512H-H1535.002H .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Holding Relay as String with LoHi Byte Order	H0000.512L-H1535.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Index Register	IR00-IR15	<b>DWord</b> , Long, LBCD, Float	Read/Write*
Task Flag	TK00-TK31	<b>Boolean</b>	Read Only
Timer	T0000-T4095	<b>BCD</b> , Word, Short	Read/Write
Timer as String with HiLo Byte Order	T0000.512H-T4095.002H .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Timer as String with LoHi Byte Order	T0000.512L-T4095.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Timer Status	TS0000-TS4095	<b>Boolean</b>	Read/Write*
Working Relay	W000-W511 W000-W510 Wxxx.00-Wxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float, <b>Boolean</b>	Read/Write

Device Type	Range	Data Type	Access
Working Relay as String with HiLo Byte Order	W000.512H-W511.002H .l is string length, range 2 to 512 characters	String	Read/Write
Working Relay as String with LoHi Byte Order	W000.512L-W511.002L .l is string length, range 2 to 512 characters	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 characters	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 characters	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 also modifies 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 contains 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 [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 and String. 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 does not change. The following occurs:

- If the device is in Run Mode and a write is attempted to a CS or TS register, an "unable to write" error message is returned.
- If the device is in Run Mode and a write is attempted to a DR or IR register, the write succeeds. Although no error message is returned, the value in the device does 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	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read Only
	A448-A959 A448-A958	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read/Write
	A000.00-A000.15-A447.00- A447.15	<b>Boolean</b>	Read Only
	A448.00-A448.15-A959.00- A959.15	<b>Boolean</b>	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 characters	<b>String</b> <b>String</b>	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 characters	<b>String</b> <b>String</b>	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 characters	<b>String</b> <b>String</b>	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	<b>String</b> <b>String</b>	Read Only  Read/Write



Device Type	Range	Data Type	Access
	.l is string length, range 1 to 256 characters		
CIO	CIO0000-CIO6143 CIO0000-CIO6142 CIOxxxx.00-CIOxxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
CIO Memory as String with HiLo Byte Order	CIO0000.512H-CIO6143.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
CIO Memory as String with LoHi Byte Order	CIO0000.512L-CIO6143.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter	C0000-C4095	<b>BCD</b> , Word, Short	Read/Write
Counter Status	CS0000-CS4095	<b>Boolean</b>	Read/Write*
Data Memory	D00000-D32767 D00000-D32766 Dxxxxx.00-Dxxxxx.15   See also <a href="#">String Support</a>	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Data Memory as String with HiLo Byte Order	D00000.512H-D32767.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
Data Memory as String with LoHi Byte Order	D00000.512L-D32767.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
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 characters	<b>String</b>	Read/Write
Data Register	DR00-DR15 DR00-DR14	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read/Write*
Expansion Data Memory (current bank)	E00000-E32767 E00000-E32766 Exxxx.00-Exxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Expansion Data Memory	E00:00000-E12:32767 E00:00000-E12:32766 Ex:x.00-Ex:xxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
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 characters	<b>String</b>	Read/Write
Holding Relay	H0000-H1535 H0000-H1534 Hxxxx.00-Hxxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Index Register	IR00-IR15	<b>DWord</b> , Long, LBCD, Float	Read/Write*
Task Flag	TK00-TK31	<b>Boolean</b>	Read Only
Timer	T0000-T4095	<b>BCD</b> , Word, Short	Read/Write
Timer Status	TS0000-TS4095	<b>Boolean</b>	Read/Write*
Working Relay	W000-W511 W000-W510 Wxxx.00-Wxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float <b>Boolean</b>	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 also modifies 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 contains 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 [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 and String. 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 does not change. The following occurs:


- If the device is in Run Mode and a write is attempted to a CS or TS register, an "unable to write" error message is returned.
- If the device is in Run Mode and a write is attempted to a DR or IR register, the write succeeds. Although no error message is returned, the value in the device does 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	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read Only
	A448-A1471 A448-A1470	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read/Write
	A10000-A11535 A10000-A11534	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read Only
	A000.00-A000.15-A447.00- A447.15	<b>Boolean</b>	Read Only
	A448.00-A448.15- A1471.00-A1471.15	<b>Boolean</b>	Read/Write
	A10000.00-A10000.15- A11535.00-A11535.15	<b>Boolean</b>	Read Only
Auxiliary Relay as String with HiLo Byte Order	A000.512H-A447.002H A448.512H-A1471.002H A10000.512H- A11535.002H	<b>String</b> <b>String</b> <b>String</b>	Read Only Read/Write Read Only
	.l is string length, range 2 to 512 characters		
Auxiliary Relay as String with LoHi Byte Order	A000.512L-A447.002L A448.512L-A1471.002L A10000.512L-A11535.002L	<b>String</b> <b>String</b> <b>String</b>	Read Only Read/Write Read Only
	.l is string length, range 2 to 512 characters		

Device Type	Range	Data Type	Access
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 characters	<b>String</b> <b>String</b> <b>String</b>	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 characters	<b>String</b> <b>String</b> <b>String</b>	Read Only  Read/Write  Read Only
CIO	CIO0000-CIO6143 CIO0000-CIO6142 CIOxxxx.00-CIOxxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
CIO Memory as String with HiLo Byte Order	CIO0000.512H- CIO6143.002H  .l is string length, range 2 to 512 characters	<b>String</b>	Read/Write
CIO Memory as String with LoHi Byte Order	CIO0000.512L- CIO6143.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Counter	C0000-C4095	<b>BCD</b> , Word, Short	Read/Write
Counter Status	CS0000-CS4095	<b>Boolean</b>	Read/Write*
Data Memory	D00000-D32767 D00000-D32766 Dxxxxx.00-Dxxxxx.15   See also <a href="#">String Support</a>	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	Read/Write
Data Memory as String with HiLo Byte Order	D00000.512H- D32767.002H  .l is string length, range 2	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
	to 512 characters		
Data Memory as String with LoHi Byte Order	D00000.512L-D32767.002L .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Data Register	DR00-DR15 DR00-DR14	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float	Read/Write*
Expansion Data Memory (current bank)	E00000-E32767 E00000-E32766 Exxxxx.00-Exxxxx.15	<b>Word</b> , Short, BCD Long, DWord, LBCD, Float <b>Boolean</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Expansion Data Memory	E00:00000-E24:32767 E00:00000-E24:32766 Ex:x.00-Exx:xxxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float <b>Boolean</b>	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 characters	<b>String</b>	Read/Write

Device Type	Range	Data Type	Access
Expansion Data Memory as String with LoHi Byte Order	E00:00000.512L- E24:32767.002L  .l is string length, range 2 to 512 characters	<b>String</b>	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 characters	<b>String</b>	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 characters	<b>String</b>	Read/Write
Holding Relay	H0000-H1535 H0000-H1534 Hxxxx.00-Hxxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float  <b>Boolean</b>	Read/Write
Index Register	IR00-IR15	<b>DWord</b> , Long, LBCD, Float	Read/Write*
Task Flag	TK00-TK127	<b>Boolean</b>	Read Only
Timer	T0000-T4095	<b>BCD</b> , Word, Short	Read/Write
Timer Status	TS0000-TS4095	<b>Boolean</b>	Read/Write*
Working Relay	W000-W511 W000-W510 Wxxx.00-Wxxx.15	<b>Word</b> , Short, BCD, Long, DWord, LBCD, Float  <b>Boolean</b>	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 also modifies 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 contains 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 [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 and String. 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.

## Event Log Messages

The following information concerns messages posted to the Event Log pane in the main user interface. Consult the server help on filtering and sorting the Event Log detail view. Server help contains many common messages, so should also be searched. Generally, the type of message (informational, warning) and troubleshooting information is provided whenever possible.

---

**Device responded with error. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes indicate possible causes.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

**See Also:**

Main and Sub Error Codes

---

**Device responded with local node error. | Main error = <code>, Sub error = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes indicate possible causes.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

**See Also:**

Main and Sub Error Codes

---

**Device responded with destination node error. | Main error = <code>, Sub error = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes indicate possible causes.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

---

**See Also:**

Main and Sub Error Codes

**Device responded with a communications error. | Main error = <code>, Sub error = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes indicate possible causes.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

**See Also:**

Main and Sub Error Codes

**Device cannot process command. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes indicate possible causes.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

**See Also:**

Main and Sub Error Codes

**Device responded with routing table error. | Main error = <code>, Sub error = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes indicate possible causes.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

**See Also:**

Main and Sub Error Codes

**Device responded with a command format error. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes indicate possible causes.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

**See Also:**

Main and Sub Error Codes

**Device responded with a command parameter error. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes indicate possible causes.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

**See Also:**

Main and Sub Error Codes

**Device responded with read not possible. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes indicate possible causes.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

**See Also:**

Main and Sub Error Codes

---

**Device responded with error in unit. | Main error = <code>, Sub error = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes indicate possible causes.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

**See Also:**

Main and Sub Error Codes

---

**Device cannot accept command. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes indicate possible causes.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

**See Also:**

Main and Sub Error Codes

---

**Access right denied. | Tag address = '<address>', Main error = <code>, Sub error = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes indicate possible causes.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

**See Also:**

Main and Sub Error Codes

---

**Device responded with write not possible. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.**

---

**Error Type:**

Warning

---

**Possible Cause:**

The error codes indicate possible causes.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

**See Also:**

Main and Sub Error Codes

---

**Device deactivated due to network settings conflict with another device. | Other device = '<device>'.**

---

**Error Type:**

Warning

**Possible Cause:**

The network parameters for the device were changed during run time and conflict with another device. Communication with this device is not possible until the conflict is resolved.

**Possible Solution:**

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

**See Also:**

FINS Networks

---

**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.

**Possible Solution:**

Examine the properties of each device. Make any change a device properties to trigger validation. Once all the devices and properties are validated, save the project.

**Note:**

This solution does not have to be repeated; new devices that are added to the project are validated automatically.

**See Also:**

FINS Networks

---

**Unable to write to tag. The device is in run mode. | Tag address = '<address>'.**

---

**Error Type:**

Warning

**Possible Cause:**

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

**Possible Solution:**

Set the device to Programming Mode.

---

**Device responded with network relay error. | Net address = <address>, Node address = <address>, Main error = <code>, Sub error = <code>, End code = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes provide information about the end code word bits 6, 7, and 15.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description corrections and solutions.

**Note:**

A network relay error in combination with other conditions and configuration can cause a remote network error.

**See Also:**

FINS Network Configuration

---

**Device responded with fatal CPU unit error. | End code = <code>.**

---

**Error Type:**

Warning

**Possible Cause:**

The error codes provide information about the end code word bits 6, 7, and 15.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description corrections and solutions.

**See Also:**

Main and Sub Error Codes

---

**Device responded with non-fatal CPU unit error. | End code = <code>.**

---

**Error Type:**



Warning

**Possible Cause:**

The error codes provide information about the end code word bits 6, 7, and 15.

**Possible Solution:**

Refer to Main and Sub Error Codes for a description corrections and solutions.

**See Also:**

Main and Sub Error Codes

**Failed to set device back to Run Mode following write to tag. | Tag address = '<address>'.**

---

**Error Type:**

Warning

## 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.

Main Code	Sub Code	Check Point	Probable Cause	Correction
	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: Communications 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	Routing table	There is an error in the	Set the routing tables correctly.

Main Code	Sub Code	Check Point	Probable Cause	Correction
	error		routing tables.	
	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.

Main Code	Sub Code	Check Point	Probable Cause	Correction
	06: Program missing	Program number in command data	FFFF Hex was not specified.	Specify FFFF Hex.
	09: Relational error	Command data	A large-small 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	The file does not exist at the specified file	Check the path and file name and correct them.

Main Code	Sub Code	Check Point	Probable Cause	Correction
		device	device.	
	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.	Verify that the rotary switch labeled "UNIT No." is set for the unit number.  Verify the two rotary switches labeled "NODE No." are set for Node No, typically the same as

Main Code	Sub Code	Check Point	Probable Cause	Correction
				the right-most octet of the device's IP address.
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	CPU bus line	An error occurred in	Check units, boards, and cables

Main Code	Sub Code	Check Point	Probable Cause	Correction
	error		data transfer between the CPU and a CPU Bus unit.	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	The node receiving the command is already	Wait for current processing to end or force the end of a

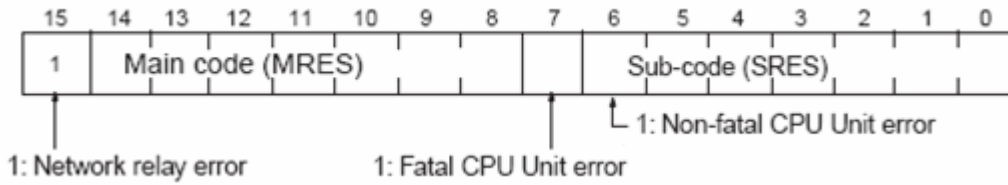


Main Code	Sub Code	Check Point	Probable Cause	Correction
		being executed	processing 5 commands.	current process and then execute the command.
	05: Service already executing		The service is being executed.	Wait for the service to end or force the end of the service and then execute the command.
	06: 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 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.	

\* A destination node error in combination with other conditions and configuration can cause a Remote Network Error.

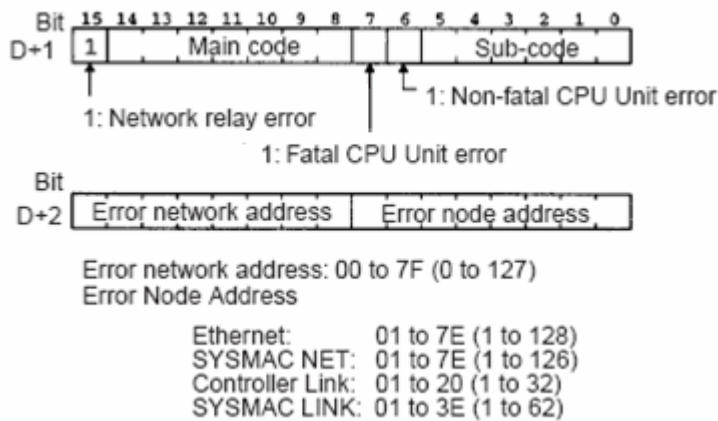
## Flags

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



## Remote Network Error

When using gateway devices; if there is a break in communications on the Remote FINS network, it is advantageous to set the device into error state (particularly if using Media Level Redundancy).

The following conditions and configuration, when they occur, are considered a remote network error and the device enters error state.

1. Network Relay Error bit in end code word is set.
2. Main code (MRES) in end code word equals destination node error (02).
3. The device FINS network configuration source network address number or destination network address number parameters have a value of 1 to 127, which indicates a remote network address.

### See Also:

[Main and Sub Error Codes](#)

[FINS Network Configuration](#)

# Index

## A

Access right denied. | Tag address = '<address>', Main error = <code>, Sub error = <code>. 95  
Address Descriptions 25  
Address specifications 25  
Attempts Before Timeout 13  
Auto-Demotion 13  
Automatic Address Generation 18

## B

BCD 24  
Boolean 24

## C

C1000H Addressing 34  
C2000H Addressing 38  
C200H Addressing 25  
C500 Addressing 30  
Channel Assignment 11  
Check Point 99  
CJ1 Addressing 81  
CJ2 Addressing 86  
Combined Address Conversion 18  
Communications Parameters 9, 16  
Communications Timeouts 12-13  
Connect Timeout 13  
CS1 Addressing 74  
CV1000 Addressing 48  
CV2000 Addressing 53  
CV500 Addressing 43  
CVM1-CPU01 Addressing 59  
CVM1-CPU11 Addressing 64  
CVM1-CPU21 Addressing 69

**D**

Data Collection 11

Data Types Description 24

Demote on Failure 14

Demotion Period 14

Destination 15

Device cannot accept command. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. 95

Device cannot process command. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. 93

Device deactivated due to network settings conflict with another device. | Other device = '<device>'. 96

Device responded with a command format error. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. 94

Device responded with a command parameter error. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. 94

Device responded with a communications error. | Main error = <code>, Sub error = <code>. 93

Device responded with destination node error. | Main error = <code>, Sub error = <code>. 92

Device responded with error in unit. | Main error = <code>, Sub error = <code>. 95

Device responded with error. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. 92

Device responded with fatal CPU unit error. | End code = <code>. 97

Device responded with local node error. | Main error = <code>, Sub error = <code>. 92

Device responded with network relay error. | Net address = <address>, Node address = <address>, Main error = <code>, Sub error = <code>, End code = <code>. 97

Device responded with non-fatal CPU unit error. | End code = <code>. 97

Device responded with read not possible. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. 94

Device responded with routing table error. | Main error = <code>, Sub error = <code>. 93

Device responded with write not possible. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. 95

Discard Requests when Demoted 14

Do Not Scan, Demand Poll Only 12

Driver 11

Duplicate Devices 20

DWord 24

**E**

Ethernet Communications 18

Event Log Messages 92

## **F**

Failed to set device back to Run Mode following write to tag. | Tag address = '<address>'. 98

FINS Messages 17

FINS Network Configuration 14

FINS Networks 17

Float 24

## **G**

General 10

## **I**

ID 11

Initial Updates from Cache 12

Inter-Request Delay 13

IP Address Table 18

## **L**

LBCD 24

Long 24

## **M**

Main and Sub Error Codes 99

Main Code 99

Model 11

Multihoming 21

## **N**

Network Adapter Cards 21

NIC 21

**O**

Optimizing Communications 23

Overview 6

**R**

Redundancy 16

Remote Network Error. 107

Request Size 16

Request Timeout 13

Respect Tag-Specified Scan Rate 12

Routing Tables 19

Run Mode Settings 15

**S**

Scan Mode 12

Setup 7

Short 24

Simulated 11

Source 14

String 24

Sub Code 99

Supported Devices 7

**T**

The current project was created with an older version of this driver. Examine device properties to validate network parameters. 96

Timeouts to Demote 14

**U**

Unable to write to tag. The device is in run mode. | Tag address = '<address>'. 97

**W**

Word 24