

Fanuc Focas HSSB Driver

© 2016 PTC Inc. All Rights Reserved.

Table of Contents

Fanuc Focas HSSB Driver	1
Table of Contents	2
Fanuc Focas HSSB Driver	4
Overview	4
External Dependencies	4
Install Focas Library	5
Setup	6
Channel Properties - General	6
Channel Properties - Write Optimizations	7
Channel Properties - Advanced	8
Device Properties - General	9
Device Properties - Scan Mode	10
Device Properties - Timing	11
Device Properties - Auto-Demotion	12
Device Properties - Communications Parameters	13
Optimizing Communications	14
Data Types Description	15
Address Descriptions	16
Series 15i	16
Series 16i	18
Series 18i	20
Series 21i	22
Power Mate i	24
Open	26
Tool Offset	28
Workpiece Zero Offset	29
Error Descriptions	31
Address Validation	31
Address <address> is out of range for the specified device or register.	31
Array size is out of range for address <address>.	31
Array support is not available for the specified address: <address>.	31
Data Type <type> is not valid for device address <address>.	32
Device address <address> contains a syntax error.	32
Device address <address> is read only.	32
Missing address.	32
Device Status Messages.	33

Device <device name> is not responding.	33
Unable to write to <address> on device <device name>.	33
General Driver Messages	33
Could not acquire library handle for device <channel.device>. FWLIB error: <code>.	34
Could not read one or more vacant macros in range starting at <address> on device <device>. ...	34
Could not set request timeout for device <channel.device>. FWLIB error: <code>.	34
Device ID <node> is too large for device <channel.device>. The maximum allowed is <max. node>.	35
Failed to read maximum Node ID for device <channel.device>. FWLIB error: <code>.	35
Read error occurred for address starting at <address> on device <channel.device>. FWLIB error: <code>.	36
Unable to start the Fanuc Focas Data Window Library services.	36
Write error occurred for address <address> on device <channel.device>. FWLIB error: <code>. ...	36
Focas1 Data Window Library Codes	38
Index	40

Fanuc Focas HSSB Driver

Help version 1.040

CONTENTS

[Overview](#)

What is the Fanuc Focas HSSB Driver?

[Device Setup](#)

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

[Optimizing Communications](#)

How do I get the best performance from the Fanuc Focas HSSB Driver?

[Data Types Description](#)

What data types does this driver support?

[Address Descriptions](#)

How do I address a data location on a Fanuc Focas1/Focas2 device?

[Error Descriptions](#)

What error messages does the Fanuc Focas HSSB Driver produce?

Overview

The Fanuc Focas HSSB Driver provides a reliable way to connect Fanuc Focas High-Speed Serial Bus (HSSB) controllers to OPC Client applications, including HMI, SCADA, Historian, MES, ERP, and countless custom applications. It is intended for use with Fanuc Focas1 Programmable Logic Controllers.

Note: For more information on the additional hardware that is required for use with this driver, refer to [External Dependencies](#).

External Dependencies

This driver has external dependencies. For this driver to communicate with the hardware, FANUC CNC Focas1/Ethernet Library (part number A02B-0207-K732) or FANUC Focas2 Library (part number A02B-0207-K737) must be [installed](#) on the system. Although the library does not need to be installed to create a server project, the project will not run without it.

Note: The Focas2 Library combines both Ethernet and HSSB capabilities and can be purchased from the FANUC distributor or by calling 1-888-326-8287. Choose CNC, PARTS to place the order, then request the part number.

Important: An HSSB interface card must be installed in the host computer and connected to the controller with the appropriate fiber optic cable.

Install Focas Library

This driver requires the Focas library to communicate with the hardware (FANUC CNC Focas1/Ethernet Library (part number A02B-0207-K732) or FANUC Focas2 Library (part number A02B-0207-K737). Follow these steps to install the library:

1. Obtain the library from the distributor (typically Fwlib*.zip).
2. Move or paste the Fwlib*.zip file to the Windows/System32 folder.
3. Unzip / extract the contents of the Fwlib*.zip in the Windows/System32 folder.
4. Reboot the computer.
5. Run the OPC server and configure a Focas1 project.

See Also: [External Dependencies](#)

Setup

Supported Devices

This driver can communicate with controllers that are compatible with the Focas1 or Focas2 CNC/PMC data window control libraries. This includes, but is not limited to, the following:

Series 0i
 Series 15
 Series 15i
 Series 16
 Series 16i
 Series 18
 Series 18i
 Series 21
 Series 21i
 Series 30i
 Series 31i
 Series 32i
 Power Mate i
 Open Addressing

Device ID

This property specifies the controller's HSSB node number. Up to 8 devices may be defined on a given channel. The valid range is 0 to 65535. The default setting is 0.

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	[-] Identification	
General	Name	
Ethernet Communications	Description	
Write Optimizations	Driver	
Advanced	[-] Diagnostics	
	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.

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

Description: User-defined information about this channel.

• Many of these properties, including Description, have an associated system tag.

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

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

Diagnostics

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

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

Not all drivers support diagnostics. To determine whether diagnostics are available for a particular driver, open the driver information and locate the "Supports device level diagnostics" statement.

Channel Properties - Write Optimizations

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

Property Groups	<input type="checkbox"/> Write Optimizations	
General	Optimization Method	Write Only Latest Value for All Tags
Ethernet Communications	Duty Cycle	10
Write Optimizations		

Write Optimizations

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

- **Write All Values for All Tags:** This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.
- **Write Only Latest Value for Non-Boolean Tags:** Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the

server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.

- **Note:** This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.
- **Write Only Latest Value for All Tags:** This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write queue before they are sent. This is the default mode.

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

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

Channel Properties - Advanced

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

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

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

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

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

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

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

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

Device Properties - General

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

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

Identification

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

● **Note:** Although descriptive names are generally a good idea, some OPC client applications may have a limited display window when browsing the OPC server's tag space. The device name and channel name become part of the browse tree information as well. Within an OPC client, the combination of channel name and device name would appear as "ChannelName.DeviceName".

● For more information, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in server help.

Description: User-defined information about this device.

● Many of these properties, including Description, have an associated system tag.

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

Driver: Selected protocol driver for this device.

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

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

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

changed to suit the needs of the application or the characteristics of the selected communications driver. The ID format can be Decimal, Octal, and Hexadecimal.

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

Operating Mode

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

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

● Notes:

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

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

Device Properties - Scan Mode

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

Property Groups	☐ Scan Mode	
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 subscribed clients. Descriptions of the options are:

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

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

Device Properties - Timing

The device Communications Timeouts 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. Communications Timeouts properties are specific to each configured device.

Property Groups	<input checked="" type="checkbox"/> Communication Timeouts	
General	Connect Timeout (s)	3
Scan Mode	Request Timeout (ms)	5000
Ethernet Encapsulation	Retry Attempts	3
Timing	<input checked="" type="checkbox"/> Timing	
Auto-Demotion	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.

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

Timing

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

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

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	Auto-Demotion	
General	Demote on Failure	Enable
Scan Mode	Timeouts to Demote	3
Timing	Demotion Period (ms)	10000
Auto-Demotion	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 - Communications Parameters

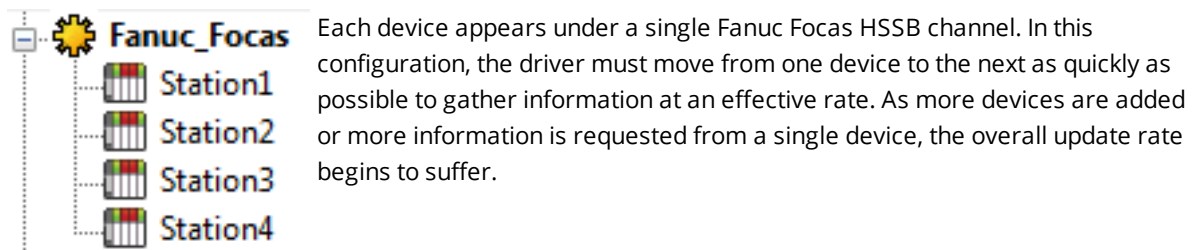
Property Groups	[-] Communications Parameters	
General	Maximum Request Size (bytes)	256
Scan Mode		
Communications Parameters		

Maximum Request Size: Specify the number of bytes that may be requested from a device at one time. To refine the driver's performance, configure the request size to one of the following: 8, 16, 32, 64, 128, 256, or 512. The default value is 256 bytes.

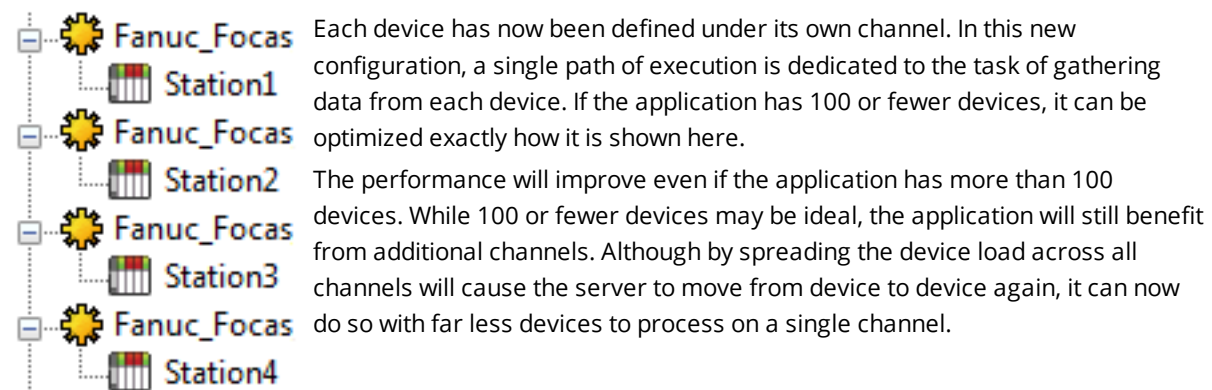
Optimizing Communications

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

Our server refers to communications protocols like Fanuc Focas HSSB as a channel. Each channel defined in the application represents a separate path of execution in the server. Once a channel has been defined, a series of devices must then be defined under that channel. Each of these devices represents a single Fanuc Focas controller from which data will be collected. While this approach to defining the application will provide a high level of performance, it won't take full advantage of the Fanuc Focas HSSB Driver or the network. An example of how the application may appear when configured using a single channel is shown below.



If the Fanuc Focas HSSB 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 100 channels. Using multiple channels distributes the data collection workload by simultaneously issuing multiple requests to the network. An example of how the same application may appear when configured using multiple channels to improve performance is shown below.



Request Size can also affect driver performance. Request size refers to the number of bytes that may be requested from a device at one time, and is available on every defined device. To refine this driver's performance, configure the request size to one of the following settings: 8, 16, 32, 64, 128, 256, or 512 bytes. Depending on the model of Fanuc Focas1/Focas2 device being used, the setting chosen can dramatically affect the application. The default value of 256 bytes is recommended. If the application consists of large requests for consecutively ordered data, users can try increasing the request size setting for the device.

For more information, refer to [Setup](#).

Data Types Description

Data Type	Description
Boolean	Single bit
Byte	Unsigned 8-bit value bit 0 is the low bit bit 7 is the high bit
Word	Unsigned 16-bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16-bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32-bit value bit 0 is the low bit bit 31 is the high bit
Long	Signed 32-bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
Float	32-bit floating point value
String	Null terminated ASCII string

Address Descriptions

Address specifications may vary depending on the model in use. Select a link from the following list to obtain specific address information for the model of interest.

Note: If the model of interest is listed as supported but is not selectable, use the Open model.

[Series 15i](#)

[Series 16i](#)

[Series 18i](#)

[Series 21i](#)

[Power Mate i](#)

[Open](#)

Series 15i

The following addresses are supported for this model. Not all address ranges may be valid for the particular device being used. For more information, refer to the specific device's documentation. Click on the following links to jump to the specific section.

[CNC Data](#)

[Arrays](#)

[Strings](#)

PMC Data

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

Address Type	Range	Data Type	Access
A (Message demand)	A00000-A00124 A00000-A00123 A00000-A00121 Axxxxx.0-Axxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
C (Counter)	C00000-C00199 C00000-C00198 C00000-C00196 Cxxxxx.0-Cxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
D (Data table)	D00000-D09999 D00000-D09998 D00000-D09996 Dxxxxx.0-Dxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
F (Signal to CNC->PMC)	F00000-F00511 F00000-F00510 F00000-F00508 Fxxxxx.0-Fxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
G (Signal to PMC->CNC)	G00000-G00511 G00000-G00510 G00000-G00508 Gxxxxx.0-Gxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
K (Keep relay)	K00000-K00909	Byte , Char	Read/Write

Address Type	Range	Data Type	Access
	K00000-K00908 K00000-K00906 Kxxxxx.0-Kxxxxx.7	Word, Short DWord, Long, Float Boolean	
R (Internal relay)	R00000-R09199 R00000-R09198 R00000-R09196 Rxxxxx.0-Rxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
T (Changeable timer)	T00000-T00299 T00000-T00298 T00000-T00296 Txxxxx.0-Txxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
X (Signal to machine->PMC)	X00000-X00127 X00000-X00126 X00000-X00124 Xxxxxx.0-Xxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
Y (Signal to PMC->machine)	Y00000-Y00127 Y00000-Y00126 Y00000-Y00124 Yxxxxx.0-Yxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
Custom Macro Value (common range)	#0100-#0999	Float	Read/Write
Custom Macro Value (local range)	#0001-#0033	Float	Read Only
Custom Macro Value (system range)	#1000-#9999	Float	Read/Write

CNC Data

Tool Offset

Workpiece Zero Offset

Arrays

Arrays are supported for all PMC addresses, except for Custom Macros in the system range and where Boolean or string data types are used. Tool Offset data cannot be addressed as an array. The syntax for declaring an array is as follows:

Mxxxx[cols] with assumed row count of 1.

Mxxxx[rows][cols] where M is the address type and xxxxx is the byte offset of the first element in the array.

Note: For all arrays, the total number of bytes being requested cannot exceed the specified request size.

Strings

All address types can be read and written to as ASCII strings. Each byte of memory will contain one ASCII character. The length of strings can range from 1 to 120 and is entered in place of the bit number. An additional character "M" is appended to the address to distinguish string addresses from bit addresses.

Example

To address a string of length 100 characters starting at D00200, enter D00200.100 M.

Note: Use caution when modifying Word, Short, DWord, Long, and Float types. Since all addresses start at a byte offset within the device, it is possible for the memory associated with tags to overlap. For example, word tags D00000 and D00001 overlap at byte 1. Writing to D00000 will also modify the value held in

D00001. It is recommended that these memory types be used such that each value to be read and written to by the driver occupy a unique range of memory in the device. For example, users might map 3 Word values to bytes D00000-D00001, D00002-D00003, and D00004-D00005. Tags to access these values would then have addresses D00000, D00002, and D00004 respectively, and a data type of Word.

Series 16i

The following addresses are supported for this model. Not all address ranges may be valid for the particular device being used. For more information, refer to the specific device's documentation. Click on the following links to jump to the specific section.

[CNC Data](#)

[Arrays](#)

[Strings](#)

PMC Data

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

Address Type	Range	Data Type	Access
A (Message demand)	A00000-A00124 A00000-A00123 A00000-A00121 Axxxx.0-Axxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
C (Counter)	C00000-C00199 C00000-C00198 C00000-C00196 Cxxxx.0-Cxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
D (Data table)	D00000-D09999 D00000-D09998 D00000-D09996 Dxxxx.0-Dxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
E (Extended relay)	E00000-E07999 E00000-E07998 E00000-E07996 Exxxxx.0-Exxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
F (Signal to CNC->PMC)	F00000-F02511 F00000-F02510 F00000-F02508 Fxxxx.0-Fxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
G (Signal to PMC->CNC)	G00000-G02511 G00000-G02510 G00000-G02508 Gxxxx.0-Gxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
K (Keep relay)	K00000-K00909 K00000-K00908 K00000-K00906 Kxxxx.0-Kxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
M (Input signal from other devices)	M00000-M00511 M00000-M00510 M00000-M00508	Byte , Char Word, Short DWord, Long, Float	Read Only

Address Type	Range	Data Type	Access
	Mxxxxx.0-Mxxxxx.7	Boolean	
N (Output signal from other devices)	N00000-N00511 N00000-N00510 N00000-N00508 Nxxxxx.0-Nxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
R (Internal relay)	R00000-R09119 R00000-R09118 R00000-R09116 Rxxxxx.0-Rxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
T (Changeable timer)	T00000-T00299 T00000-T00298 T00000-T00296 Txxxxx.0-Txxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
X (Signal to machine->PMC)	X00000-X00127 X00000-X00126 X00000-X00124 Xxxxxx.0-Xxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
Y (Signal to PMC->machine)	Y00000-Y00127 Y00000-Y00126 Y00000-Y00124 Yxxxxx.0-Yxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
Custom Macro Value (common range)	#0100-#0999	Float	Read/Write
Custom Macro Value (local range)	#0001-#0033	Float	Read Only
Custom Macro Value (system range)	#1000-#9999	Float	Read/Write

CNC Data

[Tool Offset](#)

[Workpiece Zero Offset](#)

Arrays

Arrays are supported for all PMC addresses, except for Custom Macros in the system range and where Boolean or string data types are used. Tool Offset data cannot be addressed as an array. The syntax for declaring an array is as follows:

Mxxxxx[cols] with assumed row count of 1.

Mxxxxx[rows][cols] where M is the address type and xxxxx is the byte offset of the first element in the array.

Note: For all arrays, the total number of bytes being requested cannot exceed the specified request size.

Strings

All address types can be read and written to as ASCII strings. Each byte of memory will contain one ASCII character. The length of strings can range from 1 to 120 and is entered in place of the bit number. An additional character "M" is appended to the address to distinguish string addresses from bit addresses.

Example

To address a string of length 100 characters starting at D00200, enter D00200.100 M.

Note: Use caution when modifying Word, Short, DWord, Long, and Float types. Since all addresses start at a byte offset within the device, it is possible for the memory associated with tags to overlap. For example, word tags D00000 and D00001 overlap at byte 1. Writing to D00000 will also modify the value held in D00001. It is recommended that these memory types be used such that each value to be read and written to by the driver occupy a unique range of memory in the device. For example, users might map 3 Word values to bytes D00000-D00001, D00002-D00003, and D00004-D00005. Tags to access these values would then have addresses D00000, D00002, and D00004 respectively, and a data type of Word.

Series 18i

The following addresses are supported for this model. Not all address ranges may be valid for the particular device being used. For more information, refer to the specific device's documentation. Click on the following links to jump to the specific section.

[CNC Data](#)

[Arrays](#)

[Strings](#)

PMC Data

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

Address Type	Range	Data Type	Access
A (Message demand)	A00000-A00124 A00000-A00123 A00000-A00121 Axxxxx.0-Axxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
C (Counter)	C00000-C00199 C00000-C00198 C00000-C00196 Cxxxxx.0-Cxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
D (Data table)	D00000-D09999 D00000-D09998 D00000-D09996 Dxxxxx.0-Dxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
E (Extended relay)	E00000-E07999 E00000-E07998 E00000-E07996 Exxxxx.0-Exxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
F (Signal to CNC->PMC)	F00000-F02511 F00000-F02510 F00000-F02508 Fxxxxx.0-Fxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
G (Signal to PMC->CNC)	G00000-G02511 G00000-G02510 G00000-G02508 Gxxxxx.0-Gxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
K (Keep relay)	K00000-K00909 K00000-K00908 K00000-K00906 Kxxxxx.0-Kxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write

Address Type	Range	Data Type	Access
M (Input signal from other devices)	M00000-M00511 M00000-M00510 M00000-M00508 Mxxxxx.0-Mxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
N (Output signal from other devices)	N00000-N00511 N00000-N00510 N00000-N00508 Nxxxxx.0-Nxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
R (Internal relay)	R00000-R09119 R00000-R09118 R00000-R09116 Rxxxxx.0-Rxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
T (Changeable timer)	T00000-T00299 T00000-T00298 T00000-T00296 Txxxxx.0-Txxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
X (Signal to machine->PMC)	X00000-X00127 X00000-X00126 X00000-X00124 Xxxxxx.0-Xxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
Y (Signal to PMC->machine)	Y00000-Y00127 Y00000-Y00126 Y00000-Y00124 Yxxxxx.0-Yxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
Custom Macro Value (common range)	#0100-#0999	Float	Read/Write
Custom Macro Value (local range)	#0001-#0033	Float	Read Only
Custom Macro Value (system range)	#1000-#9999	Float	Read/Write

CNC Data

[Tool Offset](#)

[Workpiece Zero Offset](#)

Arrays

Arrays are supported for all PMC addresses, except for Custom Macros in the system range and where Boolean or string data types are used. Tool Offset data cannot be addressed as an array. The syntax for declaring an array is as follows:

Mxxxxx[cols] with assumed row count of 1.

Mxxxxx[rows][cols] where M is the address type and xxxxx is the byte offset of the first element in the array.

Note: For all arrays, the total number of bytes being requested cannot exceed the specified request size.

Strings

All address types can be read and written to as ASCII strings. Each byte of memory will contain one ASCII character. The length of strings can range from 1 to 120 and is entered in place of the bit number. An additional character "M" is appended to the address to distinguish string addresses from bit addresses.

Example

To address a string of length 100 characters starting at D00200, enter D00200.100 M.

Note: Use caution when modifying Word, Short, DWord, Long, and Float types. Since all addresses start at a byte offset within the device, it is possible for the memory associated with tags to overlap. For example, word tags D00000 and D00001 overlap at byte 1. Writing to D00000 will also modify the value held in D00001. It is recommended that these memory types be used such that each value to be read and written to by the driver occupy a unique range of memory in the device. For example, users might map 3 Word values to bytes D00000-D00001, D00002-D00003, and D00004-D00005. Tags to access these values would then have addresses D00000, D00002, and D00004 respectively, and a data type of Word.

Series 21i

The following addresses are supported for this model. Not all address ranges may be valid for the particular device being used. For more information, refer to the specific device's documentation. Click on the following links to jump to the specific section.

[CNC Data](#)

[Arrays](#)

[Strings](#)

PMC Data

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

Address Type	Range	Data Type	Access
A (Message demand)	A00000-A00124 A00000-A00123 A00000-A00121 Axxxxx.0-Axxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
C (Counter)	C00000-C00199 C00000-C00198 C00000-C00196 Cxxxxx.0-Cxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
D (Data table)	D00000-D09999 D00000-D09998 D00000-D09996 Dxxxxx.0-Dxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
E (Extended relay)	E00000-E07999 E00000-E07998 E00000-E07996 Exxxxx.0-Exxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
F (Signal to CNC->PMC)	F00000-F02511 F00000-F02510 F00000-F02508 Fxxxxx.0-Fxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
G (Signal to PMC->CNC)	G00000-G02511 G00000-G02510 G00000-G02508 Gxxxxx.0-Gxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
K (Keep relay)	K00000-K00909 K00000-K00908	Byte , Char Word, Short	Read/Write

Address Type	Range	Data Type	Access
	K00000-K00906 Kxxxxx.0-Kxxxxx.7	DWord, Long, Float Boolean	
M (Input signal from other devices)	M00000-M00511 M00000-M00510 M00000-M00508 Mxxxxx.0-Mxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
N (Output signal from other devices)	N00000-N00511 N00000-N00510 N00000-N00508 Nxxxxx.0-Nxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
R (Internal relay)	R00000-R09119 R00000-R09118 R00000-R09116 Rxxxxx.0-Rxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
T (Changeable timer)	T00000-T00299 T00000-T00298 T00000-T00296 Txxxxx.0-Txxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
X (Signal to machine->PMC)	X00000-X00127 X00000-X00126 X00000-X00124 Xxxxxx.0-Xxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
Y (Signal to PMC->machine)	Y00000-Y00127 Y00000-Y00126 Y00000-Y00124 Yxxxxx.0-Yxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
Custom Macro Value (common range)	#0100-#0999	Float	Read/Write
Custom Macro Value (local range)	#0001-#0033	Float	Read Only
Custom Macro Value (system range)	#1000-#9999	Float	Read/Write

CNC Data

[Tool Offset](#)

[Workpiece Zero Offset](#)

Arrays

Arrays are supported for all PMC addresses, except for Custom Macros in the system range and where Boolean or string data types are used. Tool Offset data cannot be addressed as an array. The syntax for declaring an array is as follows:

Mxxxxx[cols] with assumed row count of 1.

Mxxxxx[rows][cols] where M is the address type and xxxxx is the byte offset of the first element in the array.

Note: For all arrays, the total number of bytes being requested cannot exceed the specified request size.

Strings

All address types can be read and written to as ASCII strings. Each byte of memory will contain one ASCII character. The length of strings can range from 1 to 120 and is entered in place of the bit number. An additional character "M" is appended to the address to distinguish string addresses from bit addresses.

Example

To address a string of length 100 characters starting at D00200, enter D00200.100 M.

Note: Use caution when modifying Word, Short, DWord, Long, and Float types. Since all addresses start at a byte offset within the device, it is possible for the memory associated with tags to overlap. For example, word tags D00000 and D00001 overlap at byte 1. Writing to D00000 will also modify the value held in D00001. It is recommended that these memory types be used such that each value to be read and written to by the driver occupy a unique range of memory in the device. For example, users might map 3 Word values to bytes D00000-D00001, D00002-D00003, and D00004-D00005. Tags to access these values would then have addresses D00000, D00002, and D00004 respectively, and a data type of Word.

Power Mate i

The following addresses are supported for this model. Not all address ranges may be valid for the particular device being used. For more information, refer to the specific device's documentation. Click on the following links to jump to the specific section.

[CNC Data](#)[Arrays](#)[Strings](#)**PMC Data**

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

Address Type	Range	Data Type	Access
A (Message demand)	A00000-A00124 A00000-A00123 A00000-A00121 Axxxxx.0-Axxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
C (Counter)	C00000-C00199 C00000-C00198 C00000-C00196 Cxxxxx.0-Cxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
D (Data table)	D00000-D09999 D00000-D09998 D00000-D09996 Dxxxxx.0-Dxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
E (Extended relay)	E00000-E07999 E00000-E07998 E00000-E07996 Exxxxx.0-Exxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
F (Signal to CNC->PMC)	F00000-F02511 F00000-F02510 F00000-F02508 Fxxxxx.0-Fxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
G (Signal to PMC->CNC)	G00000-G02511 G00000-G02510 G00000-G02508 Gxxxxx.0-Gxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write

Address Type	Range	Data Type	Access
K (Keep relay)	K00000-K00909 K00000-K00908 K00000-K00906 Kxxxxx.0-Kxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
M (Input signal from other devices)	M00000-M00511 M00000-M00510 M00000-M00508 Mxxxxx.0-Mxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
N (Output signal from other devices)	N00000-N00511 N00000-N00510 N00000-N00508 Nxxxxx.0-Nxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
R (Internal relay)	R00000-R09119 R00000-R09118 R00000-R09116 Rxxxxx.0-Rxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
T (Changeable timer)	T00000-T00299 T00000-T00298 T00000-T00296 Txxxxx.0-Txxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
X (Signal to machine->PMC)	X00000-X00127 X00000-X00126 X00000-X00124 Xxxxxx.0-Xxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
Y (Signal to PMC->machine)	Y00000-Y00127 Y00000-Y00126 Y00000-Y00124 Yxxxxx.0-Yxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
Custom Macro Value (common range)	#0100-#0999	Float	Read/Write
Custom Macro Value (local range)	#0001-#0033	Float	Read Only
Custom Macro Value (system range)	#1000-#9999	Float	Read/Write

CNC Data

[Tool Offset](#)

[Workpiece Zero Offset](#)

Arrays

Arrays are supported for all PMC addresses, except for Custom Macros in the system range and where Boolean or string data types are used. Tool Offset data cannot be addressed as an array. The syntax for declaring an array is as follows:

Mxxxx[cols] with assumed row count of 1.

Mxxxx[rows][cols] where M is the address type and xxxxx is the byte offset of the first element in the array.

Note: For all arrays, the total number of bytes being requested cannot exceed the specified request size.

Strings

All address types can be read and written to as ASCII strings. Each byte of memory will contain one ASCII character. The length of strings can range from 1 to 120 and is entered in place of the bit number. An additional character "M" is appended to the address to distinguish string addresses from bit addresses.

Example

To address a string of length 100 characters starting at D00200, enter D00200.100 M.

Note: Use caution when modifying Word, Short, DWord, Long, and Float types. Since all addresses start at a byte offset within the device, it is possible for the memory associated with tags to overlap. For example, word tags D00000 and D00001 overlap at byte 1. Writing to D00000 will also modify the value held in D00001. It is recommended that these memory types be used such that each value to be read and written to by the driver occupy a unique range of memory in the device. For example, users might map 3 Word values to bytes D00000-D00001, D00002-D00003, and D00004-D00005. Tags to access these values would then have addresses D00000, D00002, and D00004 respectively, and a data type of Word.

Open

The following addresses are supported for this model. Not all address ranges may be valid for the particular device being used. For more information, refer to the specific device's documentation. Click on the following links to jump to the specific section.

[CNC Data](#)

[Arrays](#)

[Strings](#)

PMC Data

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

Address Type	Range	Data Type	Access
A (Message demand)	A00000-A32767 A00000-A32766 A00000-A32764 Axxxxx.0-Axxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
C (Counter)	C00000-C32767 C00000-C32766 C00000-C32764 Cxxxxx.0-Cxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
D (Data table)	D00000-D32767 D00000-D32766 D00000-D32764 Dxxxxx.0-Dxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
E (Extended relay)	E00000-E32767 E00000-E32766 E00000-E32764 Exxxxx.0-Exxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
F (Signal to CNC->PMC)	F00000-F32767 F00000-F32766 F00000-F32764 Fxxxxx.0-Fxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
G (Signal to PMC->CNC)	G00000-G32767	Byte , Char	Read/Write

Address Type	Range	Data Type	Access
	G00000-G32766 G00000-G32764 Gxxxxx.0-Gxxxxx.7	Word, Short DWord, Long, Float Boolean	
K (Keep relay)	K00000-K32767 K00000-K32766 K00000-K32764 Kxxxxx.0-Kxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
M (Input signal from other devices)	M00000-M32767 M00000-M32766 M00000-M32764 Mxxxxx.0-Mxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
N (Output signal from other devices)	N00000-N32767 N00000-N32766 N00000-N32764 Nxxxxx.0-Nxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
R (Internal relay)	R00000-R32767 R00000-R32766 R00000-R32764 Rxxxxx.0-Rxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
T (Changeable timer)	T00000-T32767 T00000-T32766 T00000-T32764 Txxxxx.0-Txxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
X (Signal to machine->PMC)	X00000-X32767 X00000-X32766 X00000-X32764 Xxxxxx.0-Xxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read Only
Y (Signal to PMC->machine)	Y00000-Y32767 Y00000-Y32766 Y00000-Y32764 Yxxxxx.0-Yxxxxx.7	Byte , Char Word, Short DWord, Long, Float Boolean	Read/Write
Custom Macro Value (common range)	#0100-#0999	Float	Read/Write
Custom Macro Value (local range)	#0001-#0033	Float	Read Only
Custom Macro Value (system range)	#1000-#9999	Float	Read/Write

CNC Data

Tool Offset

Workpiece Zero Offset

Arrays

Arrays are supported for all PMC addresses, except for Custom Macros in the system range and where Boolean or string data types are used. Tool Offset data cannot be addressed as an array. The syntax for declaring an array is as follows:

Mxxxxx[cols] with assumed row count of 1.

Mxxxxx[rows][cols] where M is the address type and xxxxx is the byte offset of the first element in the array.

Note: For all arrays, the total number of bytes being requested cannot exceed the specified request size.

Strings

All address types can be read and written to as ASCII strings. Each byte of memory will contain one ASCII character. The length of strings can range from 1 to 120 and is entered in place of the bit number. An additional character "M" is appended to the address to distinguish string addresses from bit addresses.

Example

To address a string of length 100 characters, starting at D00200, enter D00200.100 M.

Note: Use caution when modifying Word, Short, DWord, Long, and Float types. Since all addresses start at a byte offset within the device, it is possible for the memory associated with tags to overlap. For example, word tags D00000 and D00001 overlap at byte 1. Writing to D00000 will also modify the value held in D00001. It is recommended that these memory types be used such that each value to be read and written to by the driver occupy a unique range of memory in the device. For example, users might map 3 Word values to bytes D00000-D00001, D00002-D00003, and D00004-D00005. Tags to access these values would then have addresses D00000, D00002, and D00004 respectively, and a data type of Word.

Tool Offset

CNC Data

Address Type	Range	Data Type	Access
Tool Offset	TOFS:nn:o nn = Tool Number (01-64) o = Offset Type (0-9, see note below)	Long, DWord	Read/Write

Tool Offset Types

The tool offset type's meaning depends on the hardware. The following tables summarize the various offset types.

	Cutter Radius	Tool Length
Wear	0	2
Geometry	1	3

Lathe Series (T series)

	X-Axis	Z-Axis	Nose R	Imaginary Tool Nose	Y-Axis
Wear	0	2	4	6	8
Geometry	1	3	5	7	9

Tool Offset Values

Series 15, 150i

6007#0 (OFE)	6004#0 (OFD)	6002#1 (OFC)	6002#0 (OFA)	Linear axis mm input [mm]	Linear axis inch input [inch]	Rotation axis [deg]
0	0	0	1	0.01	0.001	0.01
0	0	0	0	0.001	0.0001	0.001
0	0	1	0	0.0001	0.00001	0.0001
0	1	0	0	0.00001	0.000001	0.00001
1	0	0	0	0.000001	0.0000001	0.000001

Series 16/18/21, 160/180/210, 160i/180i/210i, 0i, Power Mate, Open

	1004#1 (ISC)	1004#0 (ISA)	Linear axis mm input [mm]	Linear axis inch input [inch]	Rotation axis [deg]
IS-A*	0	1	0.01	0.001	0.01
IS-B	0	0	0.001	0.0001	0.001
IS-C**	1	0	0.0001	0.00001	0.0001

*IS-A is effective for Power Mate i-H.

**IS-C is effective for Power Mate i-D.

Workpiece Zero Offset

Not all addresses are valid for all device models.

CNC Data

Address Type	Range	Data Type	Access
Workpiece Zero Offset	ZOFS:aa:ooo aa = axis (01-32) ooo = offset (000-306)	Long, DWord	Read/Write

Workpiece Zero Offset Values**Series 150**

	1009#1 (ISE)	1004#5 (ISD)	1004#1 (ISF)	1004#0 (ISR)	Linear axis mm input [mm]	Linear axis inch input [inch]	Rotation axis [deg]
IS-A	0	0	0	1	0.01	0.001	0.01
IS-B	0	0	0	0	0.001	0.0001	0.001
IS-C	0	0	1	0	0.0001	0.00001	0.0001
IS-D	0	1	0	0	0.00001	0.000001	0.00001
IS-E	1	0	0	0	0.000001	0.0000001	0.000001

Series 15, 150i

	1012#3 (ISE)	1012#2 (ISD)	1012#1 (ISC)	1012#0 (ISA)	Linear axis mm input [mm]	Linear axis inch input [inch]	Rotation axis [deg]
IS-A	0	0	0	1	0.01	0.001	0.01
IS-B	0	0	0	0	0.001	0.0001	0.001
IS-C	0	0	1	0	0.0001	0.00001	0.0001
IS-D	0	1	0	0	0.00001	0.000001	0.00001

	1012#3 (ISE)	1012#2 (ISD)	1012#1 (ISC)	1012#0 (ISA)	Linear axis mm input [mm]	Linear axis inch input [inch]	Rotation axis [deg]
IS-E	1	0	0	0	0.000001	0.0000001	0.000001

Series 16/18/21, 160/180/210, 160i/180i/210i, 0i, Power Mate, Open

	1004#1 (ISC)	1004#0 (ISA)	Linear axis mm input [mm]	Linear axis inch input [inch]	Rotation axis [deg]
IS-A	0	1	0.01	0.001	0.01
IS-B	0	0	0.001	0.0001	0.001
IS-C	1	0	0.0001	0.00001	0.0001

Series 300i

	1013#3 (ISE)	1013#2 (ISD)	1013#1 (ISC)	1013#0 (ISA)	Linear axis mm input [mm]	Linear axis inch input [inch]	Rotation axis [deg]
IS-A	0	0	0	1	0.01	0.001	0.01
IS-B	0	0	0	0	0.001	0.0001	0.001
IS-C	0	0	1	0	0.0001	0.00001	0.0001
IS-D	0	1	0	0	0.00001	0.000001	0.00001
IS-E	1	0	0	0	0.000001	0.0000001	0.000001

Error Descriptions

The following categories of messages may be generated. Click on a link for a list of messages.

[Address Validation](#)

[Device Status Messages](#)

[General Driver Messages](#)

[Focas1 Data Window Library Codes](#)

Address Validation

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

[Address <address> is out of range for the specified device or register.](#)

[Array size is out of range for address <address>.](#)

[Array support is not available for the specified address: <address>.](#)

[Data type <type> is not valid for device address <address>.](#)

[Device address <address> contains a syntax error.](#)

[Device address <address> is read only.](#)

[Missing address.](#)

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

Error Type:

Warning

Possible Cause:

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

Solution:

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

Array size is out of range for address <address>.

Error Type:

Warning

Possible Cause:

A tag address that has been specified statically is requesting an array size that is too large for the address type or block size of the driver.

Solution:

Re-enter the address in the client application to specify a smaller value for the array or a different starting point.

Array support is not available for the specified address: <address>.

Error Type:

Warning

Possible Cause:

A tag address that has been specified statically contains an array reference for an address type that doesn't support arrays.

Solution:

Re-enter the address in the client application to remove the array reference or correct the address type.

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

Error Type:

Warning

Possible Cause:

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

Solution:

Modify the requested data type in the client application.

Device address <address> contains a syntax error.

Error Type:

Warning

Possible Cause:

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

Solution:

Re-enter the address in the client application.

Device address <address> is read only.

Error Type:

Warning

Possible Cause:

A tag address that has been specified statically has a requested access mode that is not compatible with what the device supports for that address.

Solution:

Change the access mode in the client application.

Missing address.

Error Type:

Warning

Possible Cause:

A tag address that has been specified statically has no length.

Solution:

Re-enter the address in the client application.

Device Status Messages.

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

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

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

Device <device name> is not responding.

Error Type:

Serious

Possible Cause:

1. The connection between the device and the host PC is broken.
2. The IP address assigned to the device is incorrect.
3. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

Solution:

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

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

Error Type:

Serious

Possible Cause:

1. The connection between the device and the host PC is broken.
2. The named device may have been assigned an incorrect IP address.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify that the IP address given to the named device matches that of the actual device.

General Driver Messages

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

[Could not acquire library handle for device <channel.device>. FWLIB error: <code>.](#)

[Could not set request timeout for device <channel.device>. FWLIB error: <code>.](#)

Could not read one or more vacant macros in range starting at <address> on device <device>.
Device ID <node> is too large for device <channel.device>. The maximum allowed is <max. node>.
Failed to read maximum Node ID for device <channel.device>. FWLIB error: <code>.
Read error occurred for address starting at <address> on device <channel.device>. FWLIB error: <code>.
Unable to start the Fanuc Focas Data Window Library services.
Write error occurred for address <address> on device <channel.device>. FWLIB error: <code>.

Could not acquire library handle for device <channel.device>. FWLIB error: <code>.

Error Type:

Warning

Possible Cause:

1. Call to Focas1 Data Window Library to connect to device failed.
2. Invalid device IP or port number.
3. The device may not be running.
4. The device may be busy processing other requests.
5. There may be a cabling problem.

Solution:

The error code provided by the library should help diagnose the problem. If the problem is transient, the driver should be able to connect on a subsequent retry.

See Also:

[Focas1 Data Window Library Error Codes](#)

Could not read one or more vacant macros in range starting at <address> on device <device>.

Error Type:

Warning

Possible Cause:

The macro number is not configured in the device.

Solution:

Check the tag address and device configuration.

Could not set request timeout for device <channel.device>. FWLIB error: <code>.

Error Type:

Warning

Possible Cause:

1. Call to Focas1 Data Window Library to set request timeout failed.
2. Invalid timeout.
3. The device may be busy processing other requests.
4. There may be a cabling problem.

Solution:

The error code provided by the library should help diagnose the problem. If the problem is transient, the driver should be able to set the timeout on a subsequent retry.

See Also:

[Focas1 Data Window Library Error Codes](#)

Device ID <node> is too large for device <channel.device>. The maximum allowed is <max. node>.

Error Type:

Serious

Possible Cause:

The node number configured as the Device ID is greater than the maximum node supported by the controller.

Solution:

Set the Device ID to a compatible node number.

Failed to read maximum Node ID for device <channel.device>. FWLIB error: <code>.

Error Type:

Serious

Possible Cause:

1. There is something wrong with the connection.
2. An incorrect version of the Focas library is installed.
3. The HSSB interface card and/or the drivers that are required for execution are not installed.

Solution:

1. Check the connection between the device and the host computer.
2. Ensure that "Focas1 for HSSB" or "Focas2 (Combined Ethernet and HSSB)" library software is installed on the host computer.
3. Install an HSSB interface card in the host computer, and use the appropriate fiber optic cable to connect it to the controller.

See Also:

[Focas1 Data Window Library Error Codes](#)

Read error occurred for address starting at <address> on device <channel.device>. FWLIB error: <code>.

Error Type:

Warning

Possible Cause:

1. Call to Focas1 Data Window Library to read data failed.
2. Invalid PMC type.
3. Invalid addresses.
4. Invalid request size.
5. The device may be busy processing other requests.
6. There may be a cabling problem.

Solution:

The error code provided by the library should help diagnose the problem. If the problem is transient, the driver should be able to read the data on a subsequent retry.

See Also:

[Focas1 Data Window Library Error Codes](#)

Unable to start the Fanuc Focas Data Window Library services.

Error Type:

Fatal

Possible Cause:

The driver was unable to load the Fanuc Focas1 Data Window Library.

Solution:

Make sure the library is installed on the computer. Contact the GE distributor for this software.

Write error occurred for address <address> on device <channel.device>. FWLIB error: <code>.

Error Type:

Warning

Possible Cause:

1. Call to Focas1 Data Window Library to write data failed.
2. Invalid PMC type.
3. Invalid address.

4. Invalid request size.
5. The device may be busy processing other requests.
6. There may be a cabling problem.

Solution:

The error code provided by the library should help diagnose the problem. If the problem is transient, the driver should be able to write the data on a subsequent retry.

See Also:

[Focas1 Data Window Library Error Codes](#)

Focas1 Data Window Library Codes

This driver uses the Fanuc Focas1 Data Window Library software to communicate with devices on the network. When the library cannot complete a request made by this driver, it will return an error code describing the reason. These error codes are included in the relevant driver error messages. This table is provided to aid in diagnosing the hardware or software configuration problem causing these errors.

Note: For more information, refer to [Device Setup](#).

Error Code	Error Type	Description
-15	DLL	There is no DLL file for CNC series.
-11	Bus	A bus error of the CNC system occurred. Contact the service section (or the section in charge).
-10	System	A system error of the CNC system occurred. Contact the service section (or the section in charge).
-9	Communication	Investigate the serial line or I/F board.
-8	Handle	Invalid connection handle.
-7	Version	The CNC/PMC version does not match that of the library. Replace the library or the CNC/PMC control software.
-6	Unexpected	An unanticipated error occurred.
-5	System	A system error of CNC occurred. Contact the service section (or the section in charge).
-4	Parity	A hardware error occurred. Contact the service section.
-3	Install	The drivers required for execution are not installed.
-2	Reset	The RESET or STOP button was pressed.
-1	Busy	The CNC was busy processing another request. This commonly occurs during slave device connect attempts. The driver will retry until a connection is made.
0	Normal	Function was completed without error.
1 (CNC)	Function	Function was not executed or is not available. This can occur if the Unsolicited Message Server goes down while the driver is using it. The driver will attempt to restart the message server.
1 (PMC)	No PMC	The PMC does not exist.
2	Length	Invalid data block length.
3 (CNC)	Number	Invalid data number.
3 (PMC)	Range	Invalid address range.
4 (CNC)	Attribute	Invalid data attribute. This could result from a bad address type or range for data Read/Write.
4 (PMC)	Type	Invalid address type.
5	Data	Invalid data.
6	No Option	Invalid CNC option.
7	Protection	Write operation is prohibited.
8	Overflow	CNC tape memory is overflowed.

Error Code	Error Type	Description
9	Parameter	CNC parameter is set incorrectly.
10	Buffer	The buffer is empty or full. This can occur if there are more slave devices than the Unsolicited Message Server is configured to handle.
11	Path	Invalid path number.
12	Mode	Invalid CNC mode.
13	Reject	CNC rejected request. This can occur if an attempt is made to start multiple unsolicited messaging sessions with the same device.
14	Data Server	Data server error occurred.
15	Alarm	Function cannot be executed due to an alarm in CNC.
16	Stop	CNC status is stop or emergency.
17	Password	Data is protected by the CNC data protection function.

Index

A

Address <address> is out of range for the specified device or register. 31

Address Descriptions 16

Address Validation 31

Advanced Channel Properties 8

Array size is out of range for address <address>. 31

Array support is not available for the specified address:<address>. 31

B

Boolean 15

C

Channel Assignment 9

Channel Properties - General 6

Channel Properties - Write Optimizations 7

Communications Parameters 13

Communications Timeouts 11-12

Connect Timeout 11

Could not acquire library handle for device <channel.device>. FWLIB error <code>. 34

Could not read one or more vacant macros in range starting at <address> on device <device>. 34

Could not set request timeout for device <channel.device>. FWLIB error <code>. 34

D

Data Collection 10

Data Type <type> is not valid for device address <address>. 32

Data Types Description 15

Demote on Failure 12

Demotion Period 12

Description 9

Device <device name> is not responding. 33

Device address <address> contains a syntax error. 32

Device address <address> is read only. 32

Device ID 6
Device ID <node> is too large for device <channel.device>. The maximum allowed is <max node>. 35
Device Properties - Auto-Demotion 12
Device Properties - General 9
Device Status Messages. 33
Diagnostics 7
Discard Requests when Demoted 12
Do Not Scan, Demand Poll Only 11
Driver 7, 9
Driver Error Messages 33
Duty Cycle 8
DWord 15

E

Error Descriptions 31
External Dependencies 4

F

Failed to read maximum Node ID for device <channel.device>. FWLIB error: <code>. 35
Float 15
Focas1 Data Window Library Error Codes 38

I

ID 9
IEEE-754 floating point 8
Initial Updates from Cache 11
Install Focas Library 5
Inter-Request Delay 12

L

Long 15

M

Missing address. 32

Model 9

N

Name 9

Non-Normalized Float Handling 8

O

Open 26

Optimization Method 7

Optimizing Fanuc Focas HSSB Communications 14

Overview 4

P

Power Mate i 24

R

Read error occurred for address starting at <address> on device <channel.device>. FWLIB error <code>. 36

Request All Data at Scan Rate 11

Request Data No Faster than Scan Rate 11

Request Timeout 11

Respect Client-Specified Scan Rate 11

Respect Tag-Specified Scan Rate 11

Retry Attempts 12

S

Scan Mode 10

Series 15i 16

Series 16i 18

Series 18i 20

Series 21i 22

Setup 6

Short 15

Simulated 10

T

Timeouts to Demote 12

Tool Offset Tags 28

U

Unable to start the Fanuc Focas Data Window Library services. 36

Unable to write tag <address> on device <device name>. 33

W

Word 15

Workpiece Zero Offset Tags 29

Write All Values for All Tags 7

Write error occurred for address <address> on device <channel.device>. FWLIB error <code>. 36

Write Only Latest Value for All Tags 8

Write Only Latest Value for Non-Boolean Tags 7

Write Optimizations 7