Fanuc Focas HSSB Driver

© 2025 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
Setup 5
Channel Properties – General
Tag Counts 6
Channel Properties – Write Optimizations
Channel Properties – Advanced
Device Properties – General
Operating Mode
Tag Counts
Device Properties – Scan Mode10
Device Properties – Timing
Device Properties – Auto-Demotion
Device Properties – Communications Parameters
Optimizing Communications
Data Types Description 14
Address Descriptions 15
Series 15i
Series 16i
Series 18i
Series 21i
Power Mate i 22
-
Open
Tool Offset
Workpiece Zero Offset
Event Log Messages 29
Unable to start the GE Focas Data Window Library services. 29
Could not acquire library handle for device. FWLIB error = <error></error>
Could not set request timeout for device. FWLIB error = <error></error>
Read error occurred for address on device. Start address >', FWLIB error = <error></error>
Exception occurred while processing write request on device. Address = 'address >', "Well end = 'end' = '
Write error occurred for address on device. Address = ' <address '<address="<address = " <address="<address = " <address<="" =="" td=""></address>
Device ID is too large for device. Specified ID = <id>, Maximum allowed ID = <max id=""></max></id>
Failed to read maximum node id for device. FWLIB error = <error>. 31</error>
Failed to read maximum node id for device. FWLIB error = <error>. </error>

Fanuc Focas HSSB Driver

Help version 1.047

CONTENTS

Overview What is the Fanuc Focas HSSB Driver?

<u>Setup</u>

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?

Event Log

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.

For more information on the additional hardware that is required for use with this driver, refer to <u>External</u> 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 <u>installed</u> 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:

• Tip: Some system configurations require Administrative privileges to complete these steps.

- 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: <u>External Dependencies</u>

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 16i Series 16i Series 18i Series 21 Series 21i Series 20i Series 30i Series 31i Series 32i Power Mate i Open Addressing

Channel and Device Limits

The maximum number of channels supported by this driver is 100. The maximum number of devices supported by this driver is 20 per channel. Device ID specifies the controller's HSSB node number. The valid range is 0 to 65535. The default setting is 0.

Channel Properties – General

This server supports the use of multiple simultaneous 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	
h	Description	
Write Optimizations	Driver	
Advanced	Diagnostics	
	Diagnostics Capture	Disable
	Tag Counts	
	Static Tags	10

Identification

Name: Specify the 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.

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

Description: Specify user-defined information about this channel.

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

Driver: Specify the protocol / driver for this channel. Specify 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. Changes to the properties should not be made once a large client application has been developed. Utilize proper user role and privilege management to prevent operators from changing properties or accessing 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.

Tag Counts

Static Tags: Provides the total number of defined static tags at this level (device or channel). This information can be helpful in troubleshooting and load balancing.

Channel Properties – Write Optimizations

The server must ensure that the data written from the client application gets to the device on time. Given this goal, the server provides optimization properties to meet specific needs or improve application responsiveness.

Property Groups	Write Optimizations	
General	Optimization Method	Write Only Latest Value for All Tags
	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	Non-Normalized Float Handling	
General	Floating-Point Values	Replace with Zero
Write Optimizations	Inter-Device Delay	
Advanced	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:

7

- 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	
General	Description	
	Channel Assignment	
	Driver	
	Model	
	ID Format	Decimal
	ID	2

Identification

Name: Specify 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: Specify the user-defined information about this device.

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

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

Driver: Selected protocol driver for this device.

Model: Specify the type of device that is associated with this ID. The contents of the drop-down menu depend 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 documentation.*

ID: Specify 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 format is set by the driver by default. Options include 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.

Operating Mode

Property Groups	Identification	
General	Operating Mode	
	Data Collection	Enable
	Simulated	No

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: Place the device into or out of 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 data is not saved if the server removes the item (such as when the server is reinitialized). The default is No.

Notes:

- 1. Updates are not applied until clients disconnect and reconnect.
- 2. The 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.
- 4. When a device is simulated, updates may not appear faster than one (1) second in the client.

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

Tag Counts

Property Groups	Identification	
General	Operating Mode	
	Tag Counts	
	Static Tags	130

Static Tags: Provides the total number of defined static tags at this level (device or channel). This information can be helpful in troubleshooting and load balancing.

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
Scall Moue		

Scan Mode: Specify 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 OPC 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	Communication Timeouts	
General	Connect Timeout (s)	3
Scan Mode	Request Timeout (ms)	1000
Timina	Attempts Before Timeout	3
Thing		

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: Specify 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 9999999 milliseconds (167 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: Specify 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: Specify 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.

Property Groups	Timing	
General	Inter-Request Delay (ms)	0
Scan Mode		
Timing		

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 reattempts 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.

11 .

Device Properties – Communications Parameters

Communications Parameters		
Maximum Request Size (bytes)	256	
ers		
	Maximum Request Size (bytes)	Maximum Request Size (bytes) 256

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.



Channel1 Each device appears under a single Fanuc Focas HSSB channel. In this configuration, the driver must move from one device to the next as quickly as possible to gather information at an effective rate. As more devices are added or more information is requested from a single device, the overall update rate begins to suffer.

If the 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.



Channel1 Each device can be defined under its own channel. In this configuration, a single path of execution is dedicated to the task of gathering data from each device. If the application has fewer devices, it can be optimized exactly how it is shown here.

Channel2
 Channel2
 Device1
 Channel3
 The performance will improve even if the application has more than 100 devices. While fewer devices may be ideal, the application will still benefit from additional channels. Although by spreading the device load across all channels will cause the server to move from device to device again, it can now do so with far less devices to process on a single channel.

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 <u>Setup</u>.

13.

Data Types Description

Data Type	Description
Boolean	Single bit
Byte	Unsigned 8-bit value
Dyte	bit 0 is the low bit
	bit 7 is the high bit
	Unsigned 16-bit value
Word	bit 0 is the low bit
	bit 15 is the high bit
	Signed 16-bit value
Short	bit 0 is the low bit
Short	bit 14 is the high bit
	bit 15 is the sign bit
	Unsigned 32-bit value
DWord	bit 0 is the low bit
	bit 31 is the high bit
	Signed 32-bit value
Long	bit 0 is the low bit
Long	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 a particular device. For more information, refer to the specific device's documentation. Click the following links to jump to a specific section.

CNC Data

<u>Arrays</u> Strings

PMC Data

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

Address Type	Range	Data Type	Access
	A00000-A00124	Byte, Char	
	A00000-A00123	Word, Short	Read/Write
A (Message demand)	A00000-A00121	DWord, Long, Float	Redu/White
	Axxxxx.0-Axxxxx.7	Boolean	
	C00000-C00199	Byte, Char	
C (Counter)	C00000-C00198	Word, Short	Read/Write
	C00000-C00196	DWord, Long, Float	Redu/White
	Cxxxxx.0-Cxxxxx.7	Boolean	
	D00000-D09999	Byte, Char	
D (Data table)	D00000-D09998	Word, Short	Read/Write
	D00000-D09996	DWord, Long, Float	Redu/White
	Dxxxxx.0-Dxxxxx.7	Boolean	
	F00000-F00511	Byte, Char	
F (Signal to CNC->PMC)	F00000-F00510	Word, Short	Read Only
	F00000-F00508	DWord, Long, Float	Reau Only
	Fxxxxx.0-Fxxxxx.7	Boolean	
	G00000-G00511	Byte, Char	
G (Signal to PMC->CNC)	G00000-G00510	Word, Short	Read/Write
	G00000-G00508	DWord, Long, Float	Reau/White
	Gxxxxx.0-Gxxxxx.7	Boolean	
	K00000-K00909	Byte, Char	
K (Keep relay)	K00000-K00908	Word, Short	Read/Write
	K00000-K00906	DWord, Long, Float	Redu/White
	Kxxxxx.0-Kxxxxx.7	Boolean	
P (Internal relay)	R00000-R09199	Byte, Char	Pood/Write
R (Internal relay)	R00000-R09198	Word, Short	Read/Write

Address Type	Range	Data Type	Access
	R00000-R09196	DWord, Long, Float	
	Rxxxxx.0-Rxxxxx.7	Boolean	
	T00000-T00299	Byte, Char	
T (Changeable timer)	T00000-T00298	Word, Short	Read/Write
	T00000-T00296	DWord, Long, Float	nedu/ White
	Txxxxx.0-Txxxxx.7	Boolean	
	X00000-X00127	Byte, Char	
X (Signal to machine-	X00000-X00126	Word, Short	Dood Only
>PMC)	X00000-X00124	DWord, Long, Float	Read Only
	Xxxxxx.0-Xxxxxx.7	Boolean	
	Y00000-Y00127	Byte, Char	
Y (Signal to PMC->ma-	Y00000-Y00126	Word, Short	Read/Write
chine)	Y00000-Y00124	DWord, Long, Float	Redu/ White
	Yxxxxx.0-Yxxxxx.7	Boolean	
Custom Macro Value (com- mon range)	#0100-#0999	Float	Read/Write
Custom Macro Value (local range)	#0001-#0033	Float	Read Only
Custom Macro Value (sys- tem 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.

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 could 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 a particular device. For more information, refer to the specific device's documentation. Click the following links to jump to a specific

section.

CNC Data

<u>Arrays</u> Strings

PMC Data

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

Address Type	Range	Data Type	Access
	A00000-A00124	Byte, Char	
A (Message demand)	A00000-A00123	Word, Short	Read/Write
A (Message demand)	A00000-A00121	DWord, Long, Float	Reau/White
	Axxxxx.0-Axxxxx.7	Boolean	
	C00000-C00199	Byte, Char	
C (Counter)	C00000-C00198	Word, Short	Deed/M/vite
C (Counter)	C00000-C00196	DWord, Long, Float	Read/Write
	Cxxxxx.0-Cxxxxx.7	Boolean	
	D00000-D09999	Byte, Char	
	D00000-D09998	Word, Short	
D (Data table)	D00000-D09996	DWord, Long, Float	Read/Write
	Dxxxxx.0-Dxxxxx.7	Boolean	
	E00000-E07999	Byte, Char	
	E00000-E07998	Word, Short	
E (Extended relay)	E00000-E07996	DWord, Long, Float	Read/Write
	Exxxxx.0-Exxxxx.7	Boolean	
	F00000-F02511	Byte, Char	
	F00000-F02510	Word, Short	
F (Signal to CNC->PMC)	F00000-F02508	DWord, Long, Float	Read Only
	Fxxxxx.0-Fxxxxx.7	Boolean	
	G00000-G02511	Byte, Char	-
	G00000-G02510	Word, Short	
G (Signal to PMC->CNC)	G00000-G02508	DWord, Long, Float	Read/Write
	Gxxxxx.0-Gxxxxx.7	Boolean	
	K00000-K00909	Byte, Char	-
	K00000-K00908	Word, Short	
K (Keep relay)	K00000-K00906	DWord, Long, Float	Read/Write
	Kxxxxx.0-Kxxxxx.7	Boolean	
	M00000-M00511	Byte, Char	-
	M00000-M00510	Word, Short	
M (Input signal from other devices)	M00000-M00508	DWord, Long, Float	Read Only
	Mxxxxx.0-Mxxxxx.7	Boolean	
	N00000-N00511	Byte, Char	
	N00000-N00510	Word, Short	
N (Output signal from other devices)	N00000-N00508	DWord, Long, Float	Read/Write
	Nxxxxx.0-Nxxxxx.7	Boolean	
	R00000-R09119	Byte, Char	
	R00000-R09118	ord, Short	
R (Internal relay)	R00000-R09116	DWord, Long, Float	Read/Write
	Rxxxxx.0-Rxxxxx.7	Boolean	

Address Type	Range	Data Type	Access
	T00000-T00298	Word, Short	
	T00000-T00296	DWord, Long, Float	
	Txxxxx.0-Txxxxx.7	Boolean	
	X00000-X00127	Byte, Char	
X (Signal to machine->PMC)	X00000-X00126	Word, Short	Read Only
	X00000-X00124	DWord, Long, Float	Read Only
	Xxxxxx.0-Xxxxxx.7	Boolean	
	Y00000-Y00127	Byte, Char	
Y (Signal to PMC->machine)	Y00000-Y00126	Word, Short	Read/Write
	Y00000-Y00124	DWord, Long, Float	Reau/write
	Yxxxxx.0-Yxxxxx.7	Boolean	
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

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 could 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 a particular device. For more information, refer to the specific device's documentation. Click the following links to jump to a 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
	A00000-A00124	Byte, Char	
A (Message demand)	A00000-A00123	Word, Short	Deed/M/vite
	A00000-A00121	DWord, Long, Float	Read/Write
	Axxxxx.0-Axxxxx.7	Boolean	
	C00000-C00199	Byte, Char	
0 (0	C00000-C00198	Word, Short	
C (Counter)	C00000-C00196	DWord, Long, Float	Read/Write
	Cxxxxx.0-Cxxxxx.7	Boolean	
	D00000-D09999	Byte, Char	
	D00000-D09998	Word, Short	
D (Data table)	D00000-D09996	DWord, Long, Float	Read/Write
	Dxxxxx.0-Dxxxxx.7	Boolean	
	E00000-E07999	Byte, Char	
	E00000-E07998	Word, Short	
E (Extended relay)	E00000-E07996	DWord, Long, Float	Read/Write
	Exxxxx.0-Exxxxx.7	Boolean	
	F00000-F02511	Byte, Char	
	F00000-F02510	Word, Short	
F (Signal to CNC->PMC)	F00000-F02508	DWord, Long, Float	Read Only
	Fxxxxx.0-Fxxxxx.7	Boolean	
	G00000-G02511	Byte, Char	
	G00000-G02510	Word, Short	
G (Signal to PMC->CNC)	G00000-G02508	DWord, Long, Float	Read/Write
	Gxxxxx.0-Gxxxxx.7	Boolean	
	K00000-K00909	Byte, Char	
	K00000-K00908	Word, Short	
K (Keep relay)	K00000-K00906	DWord, Long, Float	Read/Write
	Kxxxxx.0-Kxxxxx.7	Boolean	
	M00000-M00511	Byte, Char	
M (Input signal from other	M00000-M00510	Word, Short	
M (Input signal from other devices)	M00000-M00508	DWord, Long, Float	Read Only
	MXXXXX.0-MXXXXX.7	Boolean	
	N00000-N00511	Byte, Char	
N (Output signal fram	N00000-N00510	Word, Short	
N (Output signal from other devices)	N00000-N00508	DWord, Long, Float	Read/Write
	Nxxxxx.0-Nxxxxx.7	Boolean	
	R00000-R09119		
	R00000-R09119 R00000-R09118	Byte, Char Word, Short	
R (Internal relay)	R00000-R09118	,	Read/Write
	RXXXXX.0-RXXXXX.7	DWord, Long, Float Boolean	
	T00000-T00299	Byte, Char	
T (Changeable timer)	T00000-T00298	Word, Short	Read/Write
	T00000-T00296	DWord, Long, Float	
	Txxxxx.0-Txxxxx.7	Boolean	
X (Signal to machine-	X00000-X00127	Byte, Char	Developed
>PMC)	X00000-X00126	Word, Short	Read Only
,	X00000-X00124	DWord, Long, Float	

19 _____

Address Type	Range	Data Type	Access
	Xxxxxx.0-Xxxxxx.7	Boolean	
	Y00000-Y00127	Byte , Char	
Y (Signal to PMC->ma-	Y00000-Y00126	Word, Short	Read/Write
chine)	Y00000-Y00124	DWord, Long, Float	Redu/ White
	Yxxxxx.0-Yxxxxx.7	Boolean	
Custom Macro Value (com- mon range)	#0100-#0999	Float	Read/Write
Custom Macro Value (local range)	#0001-#0033	Float	Read Only
Custom Macro Value (sys- tem 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 could 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 a particular device. For more information, refer to the specific device's documentation. Click the following links to jump to a 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
	A00000-A00124	Byte, Char	
A (Message demand)	A00000-A00123	Word, Short	
	A00000-A00121	DWord, Long, Float	Read/Write
	Axxxxx.0-Axxxxx.7	Boolean	
	C00000-C00199	Byte, Char	
	C00000-C00198	Word, Short	
C (Counter)	C00000-C00196	DWord, Long, Float	Read/Write
	Cxxxxx.0-Cxxxxx.7	Boolean	
	D00000-D09999	Byte, Char	
	D00000-D09998	Word, Short	
D (Data table)	D00000-D09996	DWord, Long, Float	Read/Write
	Dxxxxx.0-Dxxxxx.7	Boolean	
	E00000-E07999	Byte, Char	
F (Future de durales s)	E00000-E07998	Word, Short	Deed/M/#te
E (Extended relay)	E00000-E07996	DWord, Long, Float	Read/Write
	Exxxxx.0-Exxxxx.7	Boolean	
	F00000-F02511	Byte, Char	
Γ (Generate ONO > DMO)	F00000-F02510	Word, Short	Deed Only
F (Signal to CNC->PMC)	F00000-F02508	DWord, Long, Float	Read Only
	Fxxxxx.0-Fxxxxx.7	Boolean	
	G00000-G02511	Byte, Char	
G (Signal to PMC->CNC)	G00000-G02510	Word, Short	Read/Write
	G00000-G02508	DWord, Long, Float	Redu/White
	Gxxxxx.0-Gxxxxx.7	Boolean	
	K00000-K00909	Byte, Char	
K (Keep relay)	K00000-K00908	Word, Short	Read/Write
	K00000-K00906	DWord, Long, Float	i teau/ white
	Kxxxxx.0-Kxxxxx.7	Boolean	
	M00000-M00511	Byte, Char	
M (Input signal from other	M00000-M00510	Word, Short	Read Only
devices)	M00000-M00508	DWord, Long, Float	rtead Only
	Mxxxxx.0-Mxxxxx.7	Boolean	
	N00000-N00511	Byte, Char	
N (Output signal from	N00000-N00510	Word, Short	Read/Write
other devices)	N00000-N00508	DWord, Long, Float	
	Nxxxxx.0-Nxxxxx.7	Boolean	
	R00000-R09119	Byte, Char	
R (Internal relay)	R00000-R09118	Word, Short	Read/Write
	R00000-R09116	DWord, Long, Float	
	Rxxxxx.0-Rxxxxx.7	Boolean	
	T00000-T00299	Byte , Char	
T (Changeable timer)	T00000-T00298	Word, Short	Read/Write
(changeable amor)	T00000-T00296	DWord, Long, Float	
	Txxxxx.0-Txxxxx.7	Boolean	
	X00000-X00127	Byte , Char	
X (Signal to machine-	X00000-X00126	Word, Short	Read Only
>PMC)	X00000-X00124	DWord, Long, Float	
	Xxxxxx.0-Xxxxxx.7	Boolean	

Address Type	Range	Data Type	Access
	Y00000-Y00127	Byte, Char	
Y (Signal to PMC->ma-	Y00000-Y00126	Word, Short	Read/Write
chine)	Y00000-Y00124	DWord, Long, Float	Redu/ White
	Yxxxxx.0-Yxxxxx.7	Boolean	
Custom Macro Value (com- mon range)	#0100-#0999	Float	Read/Write
Custom Macro Value (local range)	#0001-#0033	Float	Read Only
Custom Macro Value (sys- tem range)	#1000-#9999	Float	Read/Write

<u>Tool Offset</u> 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 could 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 a particular device. For more information, refer to the specific device's documentation. Click the following links to jump to a specific section.

CNC Data

<u>Arrays</u> 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	Byte , Char	Read/Write

Address Type	Range	Data Type	Access
	A00000-A00123	Word, Short	
	A00000-A00121	DWord, Long, Float	
	Axxxxx.0-Axxxxx.7	Boolean	
	C00000-C00199	Byte, Char	
$O(O_{1})$	C00000-C00198	Word, Short	
C (Counter)	C00000-C00196	DWord, Long, Float	Read/Write
	Cxxxxx.0-Cxxxxx.7	Boolean	
	D00000-D09999	Byte, Char	
	D00000-D09998	Word, Short	
D (Data table)	D00000-D09996	DWord, Long, Float	Read/Write
	Dxxxxx.0-Dxxxxx.7	Boolean	
	E00000-E07999	Byte, Char	
	E00000-E07998	Word, Short	
E (Extended relay)	E00000-E07996	DWord, Long, Float	Read/Write
	Exxxxx.0-Exxxxx.7	Boolean	
	F00000-F02511	Byte, Char	
	F00000-F02510	Word, Short	
F (Signal to CNC->PMC)	F00000-F02508	DWord, Long, Float	Read Only
	Fxxxxx.0-Fxxxxx.7	Boolean	
	G00000-G02511	Byte, Char	
	G00000-G02510	Word, Short	
G (Signal to PMC->CNC)	G00000-G02508	DWord, Long, Float	Read/Write
	Gxxxxx.0-Gxxxxx.7	Boolean	
	K00000-K00909	Byte, Char	
	K00000-K00908	Word, Short	
K (Keep relay)	K00000-K00906	DWord, Long, Float	Read/Write
	Kxxxxx.0-Kxxxxx.7	Boolean	
	M00000-M00511	Byte, Char	
M (Input signal from other	M00000-M00510	Word, Short	
M (Input signal from other devices)	M00000-M00508	DWord, Long, Float	Read Only
	Mxxxxx.0-Mxxxxx.7	Boolean	
	N00000-N00511	Byte, Char	
N (Output signal from	N00000-N00510	Word, Short	
N (Output signal from other devices)	N00000-N00508	DWord, Long, Float	Read/Write
	NXXXXX.0-NXXXXX.7	Boolean	
	R00000-R09119	Byte, Char	
	R00000-R09119	Word, Short	
R (Internal relay)	R00000-R09118	DWord, Long, Float	Read/Write
	RXXXXX.0-RXXXXX.7	Boolean	
	T00000-T00299	Byte, Char	
T (Changeable timer)	T00000-T00298 T00000-T00296	Word, Short	Read/Write
	Txxxxx.0-Txxxxx.7	DWord, Long, Float	
		Boolean	
X (0)	X00000-X00127	Byte, Char	
X (Signal to machine- >PMC)	X00000-X00126	Word, Short	Read Only
	X00000-X00124	DWord, Long, Float	
	Xxxxxx.0-Xxxxxx.7	Boolean	
Y (Signal to PMC->ma-	Y00000-Y00127	Byte, Char	Read/Write
chine)	Y00000-Y00126	Word, Short	

Address Type	Range	Data Type	Access
	Y00000-Y00124	DWord, Long, Float	
	Yxxxxx.0-Yxxxxx.7	Boolean	
Custom Macro Value (com- mon range)	#0100-#0999	Float	Read/Write
Custom Macro Value (local range)	#0001-#0033	Float	Read Only
Custom Macro Value (sys- tem range)	#1000-#9999	Float	Read/Write

```
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. *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 could 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 a particular device. For more information, refer to the specific device's documentation. Click the following links to jump to a specific section.

CNC Data

<u>Arrays</u> Strings

PMC Data

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

Address Type	Range	Data Type	Access
	A00000-A32767	Byte, Char	
A (Message demand)	A00000-A32766	Word, Short	Read/Write
	A00000-A32764	DWord, Long, Float	

Address Type	Range	Data Type	Access
	Axxxxx.0-Axxxxx.7	Boolean	
	C00000-C32767	Byte, Char	
$O(O_{1})$	C00000-C32766	Word, Short	
C (Counter)	C00000-C32764	DWord, Long, Float	Read/Write
	Cxxxxx.0-Cxxxxx.7	Boolean	
	D00000-D32767	Byte, Char	
\mathbf{D} (Dete tel.)	D00000-D32766	Word, Short	
D (Data table)	D00000-D32764	DWord, Long, Float	Read/Write
	Dxxxxx.0-Dxxxxx.7	Boolean	
	E00000-E32767	Byte, Char	
	E00000-E32766	Word, Short	
E (Extended relay)	E00000-E32764	DWord, Long, Float	Read/Write
	Exxxxx.0-Exxxxx.7	Boolean	
	F00000-F32767	Byte, Char	
_ /	F00000-F32766	Word, Short	
F (Signal to CNC->PMC)	F00000-F32764	DWord, Long, Float	Read Only
	Fxxxxx.0-Fxxxxx.7	Boolean	
	G00000-G32767	Byte, Char	
	G00000-G32766	Word, Short	
G (Signal to PMC->CNC)	G00000-G32764	DWord, Long, Float	Read/Write
	Gxxxxx.0-Gxxxxx.7	Boolean	
	K00000-K32767	Byte, Char	
	K00000-K32766	Word, Short	
K (Keep relay)	K00000-K32764	DWord, Long, Float	Read/Write
	Kxxxxx.0-Kxxxxx.7	Boolean	
	M00000-M32767	Byte, Char	
M (Input signal from other	M00000-M32766	Word, Short	
devices)	M00000-M32764	DWord, Long, Float	Read Only
····)	Mxxxxx.0-Mxxxxx.7	Boolean	
	N00000-N32767	Byte, Char	
N (Output signal from	N00000-N32766	Word, Short	
other devices)	N00000-N32764	DWord, Long, Float	Read/Write
,	Nxxxxx.0-Nxxxxx.7	Boolean	
	R00000-R32767	Byte, Char	
	R00000-R32766	Word, Short	
R (Internal relay)	R00000-R32764	DWord, Long, Float	Read/Write
	Rxxxxx.0-Rxxxxx.7	Boolean	
	T00000-T32767	Byte, Char	
	T00000-T32766	Word, Short	
T (Changeable timer)	T00000-T32764	DWord, Long, Float	Read/Write
	Txxxxx.0-Txxxxx.7	Boolean	
	X00000-X32767	Byte, Char	
X (Signal to machine-	X00000-X32766	Word, Short	
>PMC)	X00000-X32764	DWord, Long, Float	Read Only
,	Xxxxxx.0-Xxxxxx.7	Boolean	
	Y00000-Y32767	Byte, Char	
Y (Signal to PMC->ma-	Y00000-Y32766	Word, Short	
chine)	Y00000-Y32764	DWord, Long, Float	Read/Write
	Yxxxxx.0-Yxxxxx.7	Boolean	
	1 *****		

Address Type	Range	Data Type	Access
Custom Macro Value (com- mon range)	#0100-#0999	Float	Read/Write
Custom Macro Value (local range)	#0001-#0033	Float	Read Only
Custom Macro Value (sys- tem range)	#1000-#9999	Float	Read/Write

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 could 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	6004#0	6002#1	6002#0	Linear axis mm	Linear axis inch	Rotation axis
(OFE)	(OFD)	(OFC)	(OFA)	input [mm]	input [inch]	[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	1004#0	Linear axis mm	Linear axis inch	Rotation axis [deg]
	(ISC)	(ISA)	input [mm]	input [inch]	Notation 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
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

Event Log Messages

The following information concerns messages posted to the Event Log pane in the main user interface. Consult the OPC 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.

Tip: Messages that originate from a data source (such as third-party software, including databases) are presented through the Event Log. Troubleshooting steps should include researching those messages online and in vendor documentation.

Unable to start the GE Focas Data Window Library services.

Error Type:

Error

Possible Cause:

The driver was unable to load the Fanuc Foacs1 Data Window Library.

Possible Solution:

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

Could not acquire library handle for device. | FWLIB error = <error>.

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.

Possible 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 set request timeout for device. | FWLIB error = <error>.

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.

Possible 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

Exception occurred while processing read request for address on device. | Start address = '<address>'

Error Type:

Warning

Possible Cause:

There is an error with the third-party DLL file.

Possible Solution:

Validate address or refer to the DLL file source.

Read error occurred for address on device. | Start address = '<address>', FWLIB error = <error>.

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.

Possible 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

Exception occurred while processing write request on device. | Address = '<address>'

Error Type:

Warning

Possible Cause: There is an error with the third-party DLL file.

Possible Solution:

Validate address or refer to the DLL file source.

Write error occurred for address on device. | Address = '<address>', FWLIB error = <error>.

Error Type: Warning

Possible Cause:

- 1. Call to Focas1 Data Window Library to write 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.

Possible 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

Device ID is too large for device. | Specified ID = <id>, Maximum allowed ID = <max id>

Error Type:

Warning

Possible Cause:

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

Possible Solution:

Set the Device ID to a compatible node number.

Failed to read maximum node id for device. | FWLIB error = <error>.

Error Type:

Warning

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.

Possible 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

Could not read one or more vacant macros in address range. | Range start address = '<address>'

Error Type: Warning

Possible Cause:

The macro number is not configured in the device.

Possible Solution:

Check the tag address and device configuration.

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.

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 lib- rary 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 dur- ing 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 Unso- licited 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)	Туре	Invalid address type.

Error Code	Error Type	Description
5	Data	Invalid data.
6	No Option	Invalid CNC option.
7	Protection	Write operation is prohibited.
8	Overflow	CNC tape memory is overflowed.
9	Parameter	CNC parameter is set incorrectly.
10	Buffer	The buffer is empty or full. This can occur if there are more 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

Α

Address Descriptions 15 Attempts Before Timeout 11 Auto-Demotion 11

В

Boolean 14

С

Channel Assignment 8 Channel Properties – Advanced 7 Channel Properties – General 6 Channel Properties – Write Optimizations 7 Communications Parameters 12 Communications Timeouts 10 Connect Timeout 10 Could not acquire library handle for device. | FWLIB error = <error>. 29 Could not read one or more vacant macros in address range. | Range start address = '<address>' 32 Could not set request timeout for device. | FWLIB error = <error>. 29

D

Data Collection 9 Data Types Description 14 Demote on Failure 11 Demotion Period 11 Device ID 5 Device ID is too large for device. | Specified ID = <id>, Maximum allowed ID = <max id> 31 Device Properties – Auto-Demotion 11 Device Properties – General 8 Device Properties – General 8 Device Properties – Timing 10 Diagnostics 6 Discard Requests when Demoted 11 Do Not Scan, Demand Poll Only 10 Driver 8 Duty Cycle 7 DWord 14

Ε

Event Log Messages 29

Exception occurred while processing read request for address on device. | Start address = '<address>' 30 Exception occurred while processing write request on device. | Address = '<address>' 30 External Dependencies 4

F

Failed to read maximum node id for device. | FWLIB error = <error>. 31 Float 14 Focas1 Data Window Library Error Codes 32

G

General 8

I

ID 9 Identification 6, 8 Initial Updates from Cache 10 Install Focas Library 5 Inter-Device Delay 8

L

Long 14

Μ

Model 8

Ν

Name 8 Non-Normalized Float Handling 7

0

Open 24

Operating Mode 9 Optimization Method 7 Optimizing Fanuc Focas HSSB Communications 13 Overview 4

Ρ

Power Mate i 22

R

Read error occurred for address on device. | Start address = '<address>', FWLIB error = <error>. 30 Replace with Zero 8 Request Timeout 11 Respect Tag-Specified Scan Rate 10

S

Scan Mode 10 Series 15i 15 Series 16i 16 Series 18i 18 Series 21i 20 Setup 5 Short 14 Simulated 9

Т

Tag Counts 6, 9 Timeouts to Demote 11 Timing 10 Tool Offset Tags 26

U

Unable to start the GE Focas Data Window Library services. 29 Unmodified 8

W

Word 14

Workpiece Zero Offset Tags 27 Write All Values for All Tags 7 Write error occurred for address on device. | Address = '<address>', FWLIB error = <error>. 31 Write Only Latest Value for All Tags 7 Write Only Latest Value for Non-Boolean Tags 7

37 _