Krauss Maffei MC4 Ethernet Driver

Table of Contents

Krauss Maffei MC4 Ethernet Driver

Help version 1.019

CONTENTS

Overview

The Krauss Maffei MC4 Ethernet Driver provides a reliable way to connect Krauss Maffei MC4 Ethernet devices to OPC client applications; including HMI, SCADA, Historian, MES, ERP, and countless custom applications. It is intended for communicating with Krauss Maffei MC4-Injection Moulding Machines. The driver supports the TCP/IP transport protocol.

Setup

Supported Devices

MC4-Injection Moulding Machine

Connection Limitations

Simultaneous device connections are not allowed. The MC4-Injection Moulding Machine only supports 1 connection from a host device at a time.

Supported Protocols

TCP/IP only.

Connection Timeout

This parameter specifies the time that the driver will wait for a connection to be made with a device. Depending on network load, the connect time may vary with each connection attempt. The default setting is 3 seconds. The valid range is 1 to 60 seconds.

Request Timeout

This parameter specifies the time that the driver will wait on a response from the device before giving up and going on to the next request. Longer timeouts only affect performance if a device is not responding. The default setting is 1000 milliseconds. The valid range is 100 to 9999 milliseconds.

Retry Attempts

This parameter specifies the number of times that the driver will retry a message before giving up and going on to the next message. The default setting is 3 retries. The valid range is 1 to 10.

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 255 per channel.

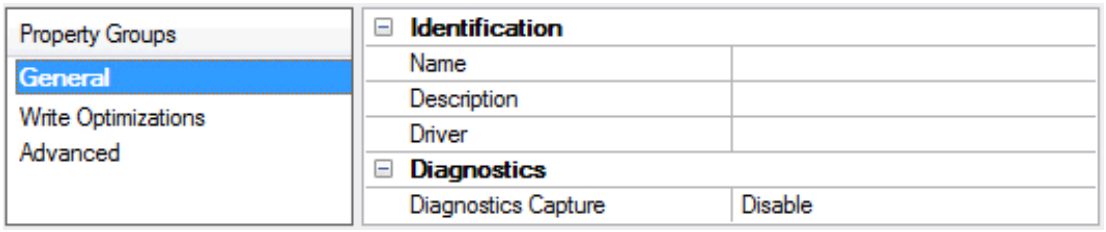
Device IDs

Each device on the channel must be uniquely identified by its own IP address.

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

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.



Identification

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

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. The property is required for creating a channel.

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

Diagnostics

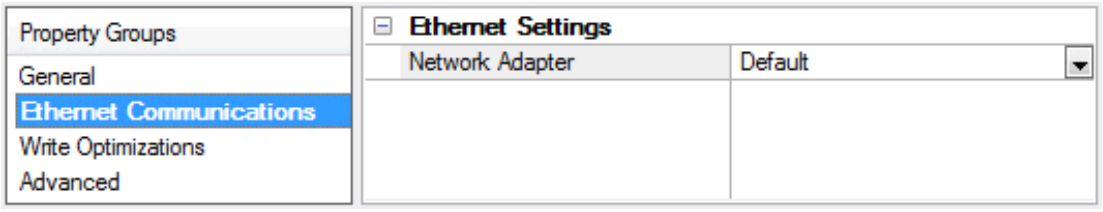
Diagnostics Capture: When enabled, this option makes the channel's diagnostic information available to OPC applications allows the usage of statistics tags that provide feedback to client applications regarding the operation of the channel. Because the server's diagnostic features require a minimal amount 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" and "Statistics Tags" in the server help.

Channel Properties — Ethernet Communications

Ethernet Communication can be used to communicate with devices.

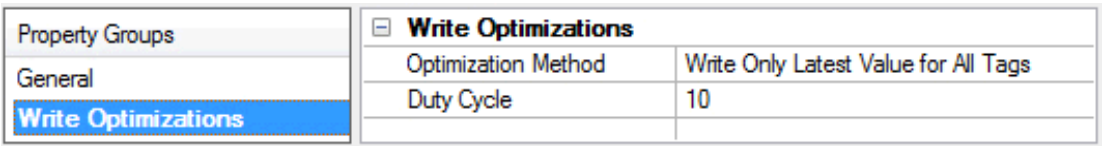


Ethernet Settings

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

Channel Properties — Write Optimizations

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



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.

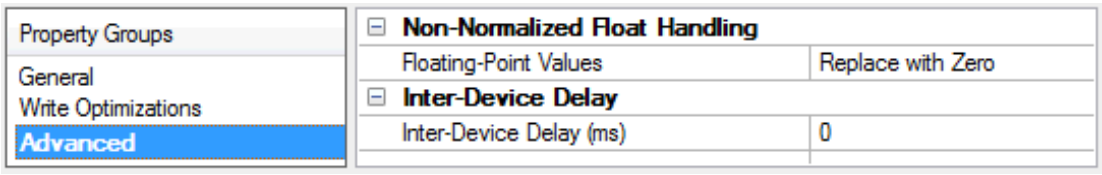
- **Write Only Latest Value for All Tags**: This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write 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.



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

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

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

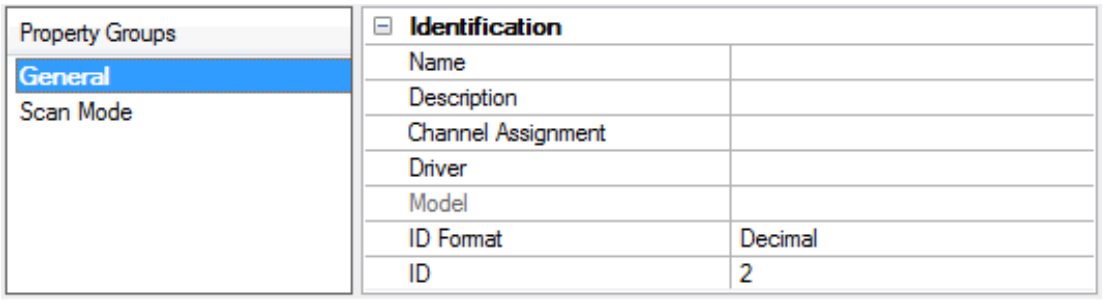
For more information on the floating point values, refer to "How To ... Work with Non-Normalized Floating Point Values" in the server help.

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

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

Device Properties — General

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.



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.

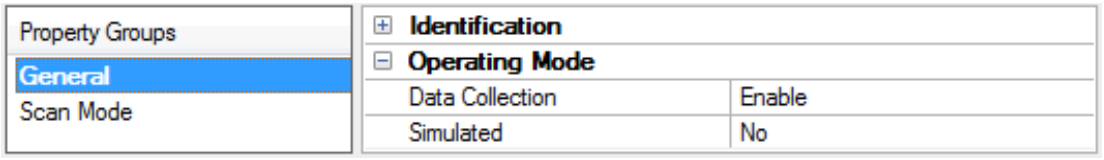
Model: This property specifies the specific type of device that is associated with this ID. The contents of the drop-down menu depends on the type of communications driver being used. Models that are not supported by a driver are disabled. If the communications driver supports multiple device models, 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 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. *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.

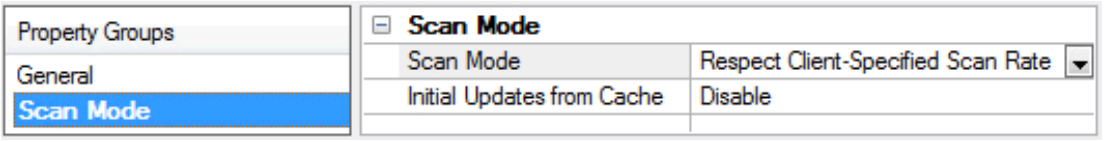
Notes:

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.



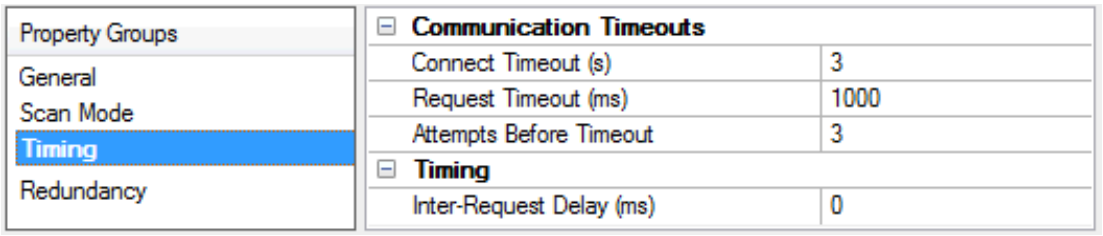
Scan Mode: Specifies how tags in the device are scanned for updates sent to subscribing clients. Descriptions of the options are:

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

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

Device Properties — Timing

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



Communications Timeouts

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

Request Timeout: This property specifies an interval used by all drivers to determine how long the driver waits for a response from the target device to complete. The valid range is 50 to 9,999,999 milliseconds (167.6667 minutes). The default is usually 1000 milliseconds, but can vary depending on the driver. The

default timeout for most serial drivers is based on a baud rate of 9600 baud or better. When using a driver at lower baud rates, increase the timeout to compensate for the increased time required to acquire data.

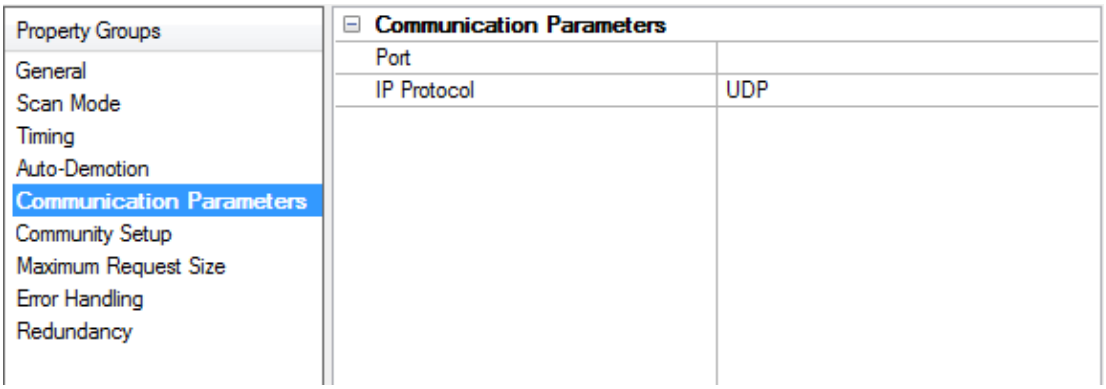
Attempts Before Timeout: This property specifies how many times the driver issues a communications request before considering the request to have failed and the device to be in error. The valid range is 1 to 10. The default is typically 3, but can vary depending on the driver's specific nature. The number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

Timing

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

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

Device Properties — Communications Parameters

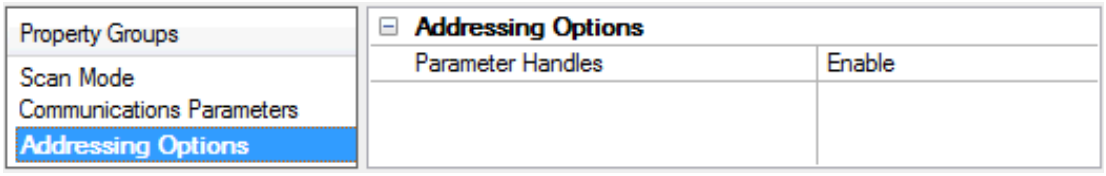


Port Number: This parameter specifies the port number that the remote device is configured to use. The default setting is 18901.

Protocol: This parameter specifies which protocol the driver should use to connect to the remote device. Although User Datagram Protocol (UDP) is listed, only Transfer Control Protocol (TCP) is supported.

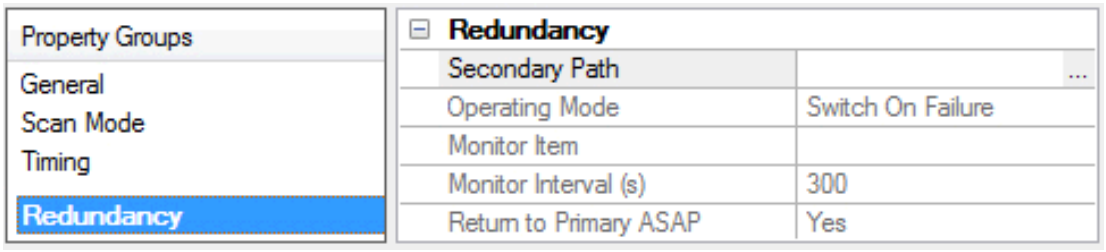
Request Size Mode: This parameter specifies the number of objects that may be requested from a device at one time. To refine performance, the request size may be configured as **Standard Mode** or **Extended Mode**. In Standard Mode, up to four objects can be requested at one time. In Extended Mode, up to 16 objects can be requested. The default is Extended Mode.

Device Properties — Addressing Options



Parameter Handles: A parameter object name can be up to 36 characters long. To reduce the number of transmission data, the MC4 computer interface has the facility for allocating an identification number (called a Parameter Handle) to each parameter object name. Parameter handles consist of 4-figure hexadecimal numbers that carry the '$' sign as prefix. This means that the parameter object name can be reduced from the maximum 36 characters down to 5. For example, "$0001" or "$001B". When enabled, the driver acquires the parameter handle for each tag and then use it to make the request to the MC4 controller. When disabled, the full 3-part parameter object name is used. The default setting is enabled.

Device Properties — Redundancy



Redundancy is available with the Media-Level Redundancy Plug-In.
Consult the website, a sales representative, or the user manual for more information.

Data Types Description

Data Type	Description
Double	64-bit floating point value The driver converts the Decimal ASCII string response to a Double.
Float	32-bit floating point value The driver converts the Decimal ASCII string response to a Float.
String	Null-terminated ASCII string

Address Descriptions

Named parameter objects are used to access data in the MC4 controller. Each parameter object is composed of the following three name sections:

<parameter object name>.<element name>.<value name>

The number of possible elements within an object and the names of these elements are subject to the object type. The possible values of a parameter element are a function of the element type.

All functionally related parameters are united in a <parameter object>. For instance, parameter object SCRW1_H_BAR_Z01 contains all parameters that are connected with the barrel heating zone 1 of injection unit 1. A single element of the parameter object is designated <parameter element>. The number of possible parameter elements within an object is subject to its type. In the case of a barrel heating zone:

- **SET:** Set-value.
- **ACT:** Actual-value.

With the <parameter value> detail, it is determined that the information of the parameter element is to be read or written. Conventional parameter values are as follows:

- **VAL:** Current value.
- **MIN:** Minimum value.
- **MAX:** Maximum value.
- **LTXT:** Parameter plain text.
- **DIM:** Dimension (= physical unit).

To read out the maximum value of the set value of barrel zone 1, the complete parameter name must read as "SCRW1_H_BAR_Z01.SET.MAX."

Parameter Element Types

Set Analog Value Parameter <Objectname>.SET.<xxx>

The following values are available with an analog value parameter.

Name	Description	Access	Data Type
VAL	Current value of the parameters.	Read/Write	Float or Double
MIN	Lower limiting value of the parameters.	Read Only	Float or Double
MAX	Upper limiting value of the parameter.	Read Only	Float or Double

Name	Description	Access	Data Type
DIM	Physical unit in the language of the actual country.	Read Only	String
LTXT	Parameter text in the current country's language.	Read Only	String

Actual Analog Value Parameter <Objectname>.ACT.<xxx>
The following values are available with an analog value parameter.

Name	Description	Access	Data Type
VAL	Current value of the parameters.	Read Only	Float or Double
MIN	Lower limiting value of the parameters.	Read Only	Float or Double
MAX	Upper limiting value of the parameter.	Read Only	Float or Double
DIM	Physical unit in the language of the actual country.	Read Only	String
LTXT	Parameter text in the current country's language.	Read Only	String

Switching Function / Status <Objectname>.SEL.<xxx>
The following values are available with a switching function.

Name	Description	Access	Data Type
VAL	Current value of the parameter.	Read/Write	Float or Double
MAX	Maximum selection function value.	Read Only	Float or Double
LTXT	Parameter text in the current country's language.	Read Only	String

String Parameter <Objectname>.STR.<xxx>
The following values are available with a string parameter.

Name	Description	Access	Data Type
VAL	Current string of the parameters.	Read/Write	String
LEN	Maximum permissible length of the string.	Read Only	Float or Double
LTXT	Parameter text in the current country's language.	Read Only	String

Date/Time Parameter <Objectname>.TIM.<xxx>
The following values are available with a date parameter.

Name	Description	Access	Data Type
VAL	Value as no. of seconds since 01.01.1980.	Read/Write	Float or Double
DATE	Date in the format "tt.mm.jj".	Read Only	String
CLOCK	Time in the "hh:mm" format.	Read Only	String
CLKSEC	Time as seconds in the "hh:mm:ss" format.	Read Only	String
LTXT	Parameter text in the current country's language.	Read Only	String

Duration Parameter <Objectname>.TIM.<xxx>
The following values are available with a time parameter.

Name	Description	Access	Data Type
VAL	Value as number of seconds.	Read/Write	Float or Double
TIME	Duration as "hhh:mm" format.	Read Only	String
TIMSEC	Duration in seconds as "hhh:mm:ss" format.	Read Only	String
LTXT	Parameter text in the current country's language.	Read Only	String

Cycle Actual Parameter <Objectname>.CYCACT.<xxx>

The following values are available with the following parameters:

Name	Description	Access	Data Type
CYCVAL	Cycle value of the parameter of cycle CYCNUM.	Read Only	Float or Double
CYCNUM	Cycle number of the cycle parameter.	Read Only	Float or Double
CYCFLG	Flags: Bit 0 – Quality control active Bit 1 – +Tolerance fault Bit 2 – - Tolerance fault Bit 3 – Startup scrap cycle Bit 8 – value not guilty	Read Only	Float or Double
DIM	Physical dimension in current country's language.	Read Only	String
LTXT	Parameter text in the current country's language.	Read Only	String

<Objectname>.CYCCNT.<xxx>

The following values are available with the following parameters:

Name	Description	Access	Data Type
VAL	Current value of the parameter.	Read Only	Float or Double
CYCNUM	Cycle number of the cycle parameter.	Read Only	Float or Double
DATE	Date in the format "tt.mm.jj".	Read Only	String
CLOCK	Time in the "hh:mm" format.	Read Only	String
CLKSEC	Time as seconds in the "hh:mm:ss" format.	Read Only	String
TIME	Duration as "hhh:mm" format.	Read Only	String
TIMSEC	Duration in seconds as "hhh:mm:ss" format.	Read Only	String
LTXT	Parameter text in the current country's language.	Read Only	String

<Objectname>.CAVSET.<xxx>

The following values are available with the following parameters:

Name	Description	Access	Data Type
VAL	Current value of the parameter.	Read/Write	Float or Double
MIN	Lower limiting value of the parameters.	Read Only	Float or Double
MAX	Upper limiting value of the parameter.	Read Only	Float or Double
DIM	Physical unit in the language of the actual country.	Read Only	String
LTXT	Parameter text in the current country's language.	Read Only	String

<Objectname>.CAVCNT.<xxx>
The following values are available with the following parameters:

Name	Description	Access	Data Type
VAL	Current value of the parameter.	Read Only	Float or Double
MIN	Lower limiting value of the parameters.	Read Only	Float or Double
MAX	Upper limiting value of the parameter.	Read Only	Float or Double
DIM	Physical unit in the language of the actual country.	Read Only	String
LTXT	Parameter text in the current country's language.	Read Only	String

<Objectname>.ALARM.<xxx>
The following values are available with the parameters.

Name	Description	Access	Data Type
NUM	Alarm number.	Read Only	Float or Double
TIME	Time of Occurrence as "tt.mm.jj hh:mm:ss" format.*	Read Only	String
LTXT	Parameter text in the current country's language.	Read Only	String

* Value of <Objectname>.ALARM.TIME can be used to determine if the alarm is active or inactive. If the alarm has been acknowledged and is in an inactive state, the value will be 01/01/80 00:00:00. If the alarm is active, the value will be the date and time that the alarm occurred (such as 05/01/05 03:17:34).

Caution: Parameter objects that can be written to become effective immediately in the MC4 and will influence the extrusion process if it is running. Any writes must be treated with care and should be avoided when production is running.

For a list of parameter objects, refer to the specific Krauss Maffei MC4 device's documentation.

Error Descriptions

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

Address Validation

Missing address
Device address <address> contains a syntax error
Data type <type> is not valid for device address <address>
Device address <address> is read only

Device Status Messages

Device <device name> is not responding
Unable to write to <address> on device <device name>

Driver Error Messages

Winsock initialization failed (OS Error = n)
Winsock V1.1 or higher must be installed to use the Krauss Maffei MC4 Ethernet device driver
Device <device name> detected a message size error (Tag <address>)
Device <device name> detected a number of parameters error in message (Tag <address>)
Device <device name> detected a parameter size error in message (Tag <address>)
Device <device name> detected a parameter value that was Too Large (Tag <address>)
Device <device name> detected a parameter value that was Too Small (Tag <address>)
Device <device name> detected an invalid data character in message (Tag <address>)
Device <device name> detected an invalid memory address (Tag <address>)
Device <device name> detected an invalid memory address Or address can't be written (Tag <address>)
Device <device name> detected an invalid parameter Handle in message (Tag <address>)
Device <device name> Failed to convert decimal response to a Float or Double number (Tag <address>)

Missing address

Error Type:

Warning

Possible Cause:

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

Solution:

Re-enter the address in the client application.

Device address <address> contains a syntax error

Error Type:

Warning

Possible Cause:
A tag address that has been specified dynamically contains one or more invalid characters.

Solution:
Re-enter the address in the client application.

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

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically has been assigned an invalid data type.

Solution:
Modify the requested data type in the client application.

Device address <address> is read only

Error Type:
Warning

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

Solution:
Change the access mode in the client application.

Device <device name> is not responding

Error Type:
Serious

Possible Cause:
The connection between the device and the Host PC is broken.

1. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.
2. The IP address assigned to the device is incorrect.
3. A second connection to the same device was attempted. The MC4 only supports one connection at a time.

Solution:

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

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.

Winsock initialization failed (OS error = n)

Error Type:
Fatal

OS Error:	Indication	Possible Solution
10091	Indicates that the underlying network subsystem is not ready for network communication.	Wait a few seconds and restart the driver.
10067	Limit on the number of tasks supported by the Windows Sockets implementation has been reached.	Close one or more applications that may be using Winsock and restart the driver.

Winsock V1.1 or higher must be installed to use the Krauss Maffei MC4 Ethernet Driver

Error Type:
Fatal

Possible Cause:
The version number of the Winsock DLL found on the system is less than 1.1.

Solution:
Upgrade Winsock to version 1.1 or higher.

Device <device name> detected a message size error (Tag <address>)

Error Type:
Serious

Possible Cause:

1. The connection between the device and the Host PC is broken or intermittent.
2. The IP address assigned to the device is incorrect.

Solution:

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

Device <device name> detected an invalid parameter handle in message (Tag <address>)

Error Type:
Serious

Possible Cause:

1. The connection between the device and the Host PC is broken or intermittent.
2. The IP address assigned to the device is incorrect.

Solution:

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

Note:
This message is only possible if "Parameter Handles" is enabled in Addressing Options.

See Also:
Addressing Options

Device <device name> detected an invalid memory address or address can't be written (Tag <address>)

Error Type:
Warning

Possible Cause:

1. Tag address specifies a parameter object name that does not exist in the device.
2. The name may be spelled incorrectly.
3. The Write value is incorrect and can not be written.

Solution:

1. Determine the correct name of the parameter object.
2. Determine the correct range of write values.

Device <device name> detected an invalid memory address (Tag <address>)

Error Type:
Warning

Possible Cause:

1. Tag address specifies a parameter object name that does not exist in the device.
2. The name may be spelled incorrectly.

Solution:
Determine the name of the correct parameter object.

Device <device name> detected an invalid data character in message (Tag <address>)

Error Type:
Serious

Possible Cause:

1. The connection between the device and the Host PC is broken or intermittent.
2. The IP address assigned to the device is incorrect.

Solution:

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

Device <device name> detected a number of parameters error in message (Tag <address>)

Error Type:
Serious

Possible Cause:

1. The connection between the device and the Host PC is broken or intermittent.
2. The IP address assigned to the device is incorrect.

Solution:

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

Device <device name> failed to convert decimal response to a Float or Double number (Tag <address>)

Error Type:
Serious

Possible Cause:
1. The connection between the device and the Host PC is broken or intermittent.
2. The IP address assigned to the device is incorrect.

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

Device <device name> detected a parameter size error in message (Tag <address>)

Error Type:
Serious

Possible Cause:
1. The connection between the device and the Host PC is broken or intermittent.
2. The IP address assigned to the device is incorrect.

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

Device <device name> detected a parameter value that was too large (Tag <address>)

Error Type:
Warning

Possible Cause:
The value written to the tag is too large. Use the tag's "XXX.XXX.MAX" value to determine the upper limit.

Solution:
Write a value less than or equal to max.

Device <device name> detected a parameter value that was too small (Tag <address>)

Error Type:
Warning

Possible Cause:
The value written to the tag is too small. Use the tag's "XXX.XXX.MIN" value to determine the lower limit.

Solution:
Write a value greater than or equal to min.

Index

E

F

G

H

I

L

M

N

O

P

R

S

T

U

W