Yokogawa MW Ethernet Driver Help

Table of Contents

Yokogawa MW Ethernet Driver Help

Help version 1.019

CONTENTS

Overview

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

Device Setup

Supported Yokogawa MW Ethernet Series Devices
MW100

Device ID
Yokogawa MW Ethernet series devices are networked using standard IP addressing. In general, the Device ID has the format YYY.YYY.YYY.YYY, where YYY designates the device IP address. Each YYY byte should be in the range of 0 to 255.

Timing
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 valid range is 1 to 30 seconds. The default setting is 3 seconds.

Request Timeout
This parameter specifies the amount of time that the driver will wait for a response from the device before giving up and going on to the next request. Long timeouts will only affect performance if a device is not responding. The valid range is 100 to 30000 milliseconds. The default setting is 1000 milliseconds.

Fail after ... successive timeouts
The valid range is 1 to 10. The default setting is 3 retries.

Note: For more information, refer to the OPC server's help documentation.

Cable Diagrams

Patch Cable (Straight Through)

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

RJ45   RJ45

10 BaseT

8-pin RJ45

Crossover Cable

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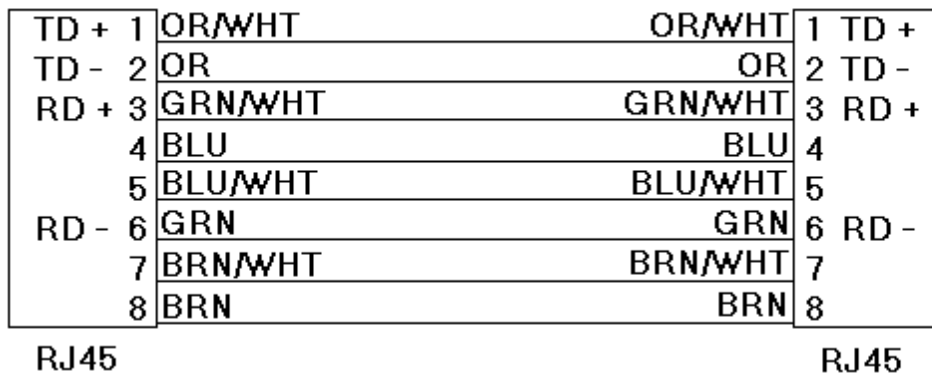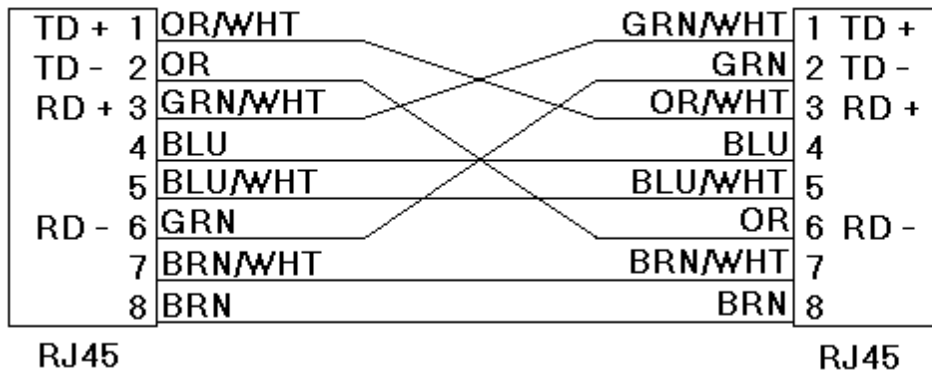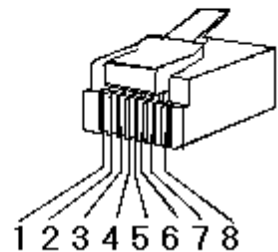
RJ45   RJ45

Communication Parameters

The Communications Parameters dialog is used to specify General, Time, Tagnames and Login settings.
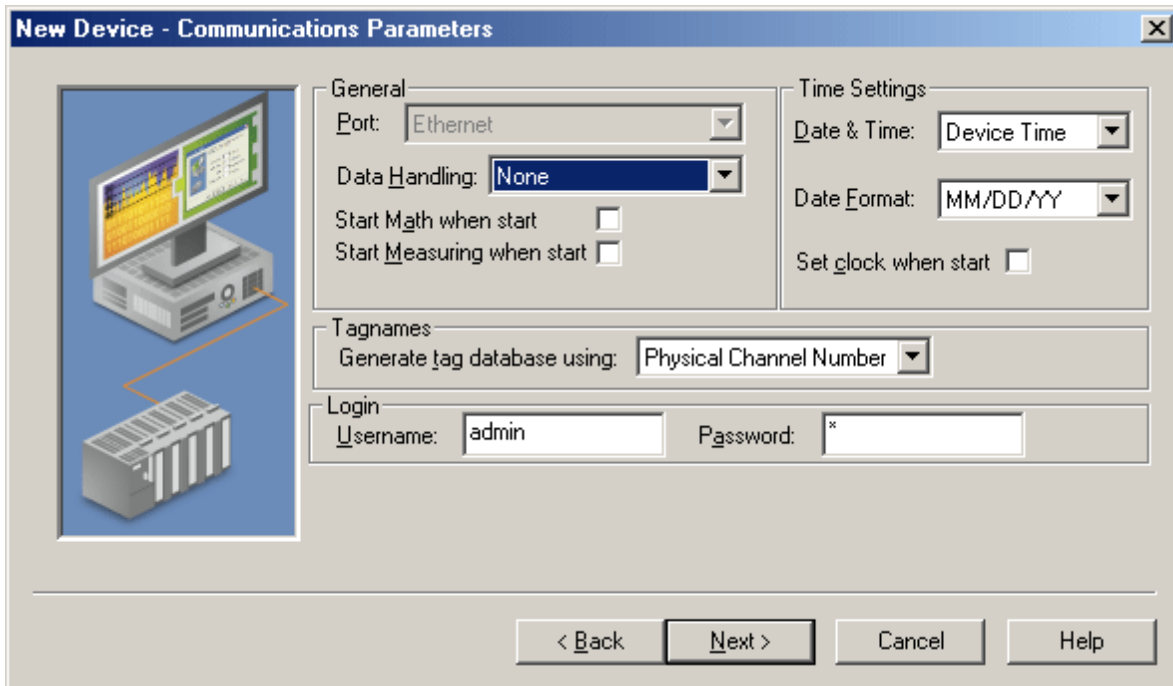
General
Port
This parameter specifies the Ethernet port supported by the driver.

Data Handling
This parameter allows the driver to be configured to return specific data values for numerical out of range and error conditions returned from the device. Special Data Handling options are NONE, +INF, and -INF. If Special Data Handling is set to NONE, special data values will be returned with the actual data value received from the device. For example, the data value of a measuring channel Over Range would be returned as 32,767. The data value of a math channel Over Range would be returned as 2,147,450,879.

If Special Data Handling is set to +INF, special data values will be returned as a numerical representation of positive infinity (#INF), with the exception of an Under Range condition that is always returned as negative infinity. When Special Data Handling is set to -INF, special data values will be returned as a numerical representation of negative infinity (-#INF), with the exception of an Over Range condition that is always returned as positive infinity.

Start Math when start
When checked, this option will inform the driver to send a command to the device at communication startup that will start the math computation.

Start Measuring when start
When checked, this option will inform the driver to send a command to the device at communication startup that will start the measuring option.

Time Settings
Date & Time
This parameter specifies the origin of the data value of the Date and Time data types which represent the date and time of the latest data. Options include **Device Time** and **System Time**. If Device Time is selected, the Date and Time tags will return the date and time read from the device. This represents the date and time of the latest data that was measured or computed based on the internal device clock. If System Time is selected, the Date and Time tags will return the date and time of the requested data that was returned from the device based on the internal system clock.

Date Format
This parameter specifies the format of the return string for the Date data type. Date formats can be specified as **MM/DD/YY**(month/day/year), **YY/MM/DD** (year/month/day) or **DD/MM/YY** (day/month/year).

Set clock when start
When checked, this option will inform the driver to send a command to the device at communication startup that will set the device clock to the date and time settings of the system clock.

Tagnames
Generate tag database using:
This parameter specifies the origin of the tag name that will be used when a tag database is automatically generated. Options include Physical Channel Number, Device Tagname and Device Tagname (Enhanced). Descriptions are as follows:

- **Physical Channel Number:** When selected, the driver will generate tag names based on the channel number of an item. For example, CH01 or CH01_alarm1.
- **Device Tagname:** When selected, the driver will generate tag names using the tag name returned by the device for a channel. For example, Flow or Flow_alarm1.

  Note: Special characters (such as slash or pound sign) are not allowed in the tag name.

- **Device Tagname (Enhanced):** When selected, the driver will generate tag names using the tag name returned by the device for a channel. For example, Flow_alarm/state or Flow_alarm#4.

  Note: Special characters are allowed in the tag name.

Login
Username
The MW devices require the user to log in with a username. When the device is configured with the login function enabled, only registered users can log in. This parameter is used to enter the registered username, which may be up to 16 alphanumeric characters. The username is case sensitive.

Note: If the device is configured with the login function enabled, the user must specify both a registered username and a password. If the device is configured with the login function disabled, users must still specify a user level in order to communicate with the MW. In this case, the username 'admin' or 'user' can be entered to indicate the user level. A password is not required.

Password
This parameter is used to enter the registered password, which may be up to 6 alphanumeric characters. Password entry will not be displayed on the Device Configuration screen.

Optimizing Your Ethernet Communications

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

Our server refers to communications protocols like Yokogawa MW Ethernet driver 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 Yokogawa MW Ethernet Device 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 Yokogawa MW Ethernet Driver or the network. An example of how the application may appear when configured using a single channel is shown below.
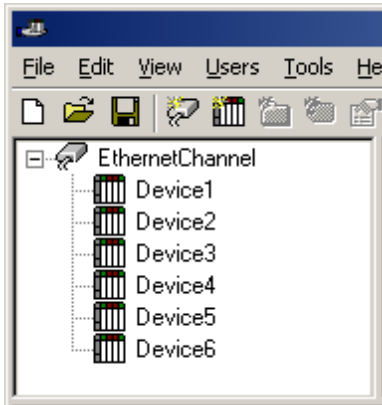
Each device appears under a single Yokogawa MW Ethernet channel. In this configuration, the driver must move from one device to the next as quickly as possible in order to gather information at an effective rate. As more devices are added or more information is requested from a single device, the overall update rate begins to suffer.

If the Yokogawa MW Ethernet Driver could only define one single channel, then the example shown above would be the only option available; however, the Yokogawa MW Ethernet Driver can define up to 16 channels. Using multiple channels distributes the data collection workload by simultaneously issuing multiple requests to the network. An example of how the same application may appear when configured using multiple channels to improve performance is shown below.
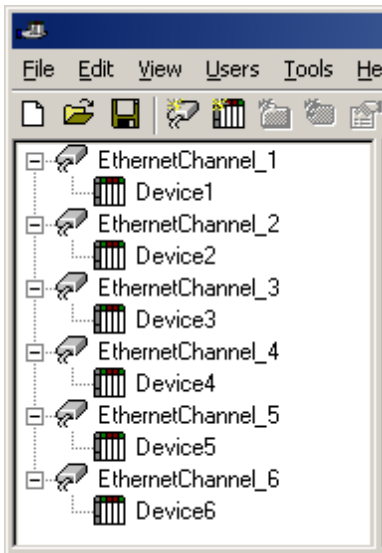
Each device has now been defined under its own channel. In this new configuration, a single path of execution is dedicated to the task of gathering data from each device. If the application has 16 or fewer devices, it can be optimized exactly how it is shown here.

The performance will improve even if the application has more than 16 devices. While 16 or fewer devices may be ideal, the application will still benefit from additional channels. Although by spreading the device load across all channels will cause the server to move from device to device again, it can now do so with far less devices to process on a single channel.

Data Types Description

| Data Type	Description
Boolean	Single bit
Byte	Unsigned 8 bit value
Word	Unsigned 16 bit value
Short	Signed 16 bit value
Long	Signed 32 bit value
Float	32 bit floating point value
Double	64 bit floating point value
String	Null terminated ASCII string

Automatic Tag Database Generation

The automatic OPC tag database generation features of this driver have been designed to make setting up the OPC application a Plug and Play operation. This driver can be configured to automatically build a list of OPC tags within the OPC Server that correspond to device specific data. The automatically generated OPC tags can then be browsed from the OPC client. The OPC tags that are generated depend on the nature of the driver.

If the target device supports its own local tag database, the driver will read the device's tag information and use this data to generate OPC tags within the OPC Server. If the device does not natively support its own named tags, the driver will create a list of tags based on information specific to the driver. An example of these two conditions may be as follows:

A data acquisition system that supports its own local tag database. The driver will use the tags names found in the device to build the OPC Server's OPC tags.

1. An Ethernet I/O system that supports the detection of an I/O module type. The driver in this case will automatically generate OPC tags in the OPC Server that are based on the types of I/O modules plugged into the Ethernet I/O rack.

2. The mode of operation for automatic tag database generation is completely configurable.

The following dialog is used to configure how the OPC Server and the associated communications driver will handle automatic OPC tag database generation:
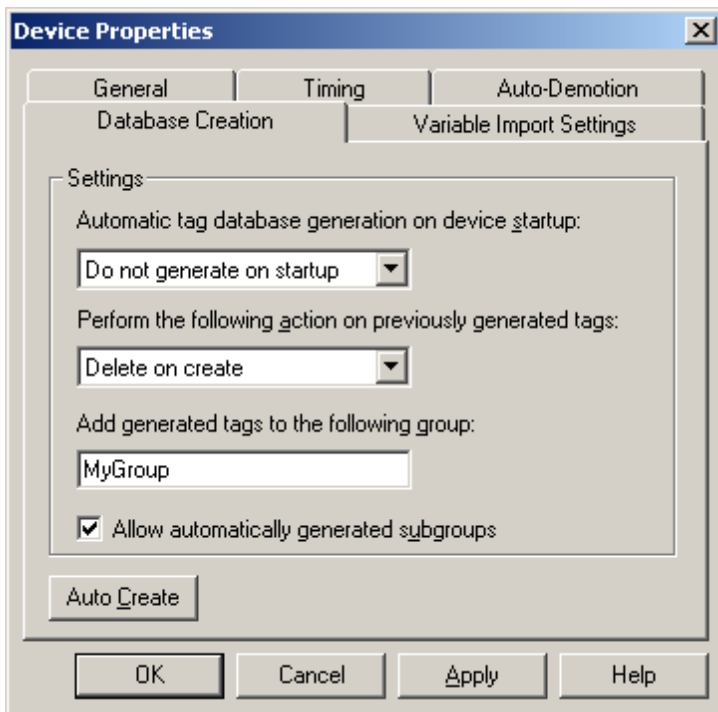
The **Automatic tag database generation on device startup** selection is used to configure when OPC tags will be automatically generated. There are three possible selections:

- **Do not generate on startup**, the default condition, prevents the driver from adding any OPC tags to tag space of the OPC Server.
- **Always generate on startup** causes the driver to always evaluate the device for tag information and to add OPC tags to the tag space of the server each time the server is launched.
- **Generate on first startup** causes the driver to evaluate the target device for tag information the first time this OPC Server project is run and to add any OPC tags to the server tag space as needed.

When the automatic generation of OPC tags is selected, any tags that are added to the server's tag space must be saved with the project. The OPC Server project can be configured to automatically save from the **Tools** | **Options** menu.

When automatic tag generation is enabled, the server needs to know what to do with OPC tags that it may have added from a previous run or with OPC tags that you may have added or modified after the communications driver

added them. **Perform the following action** is used to control how the server will handle OPC tags that were automatically generated and currently exist in the OPC Server project. This feature prevents automatically generated tags from piling up in the server. This would occur if, using the Ethernet I/O example above, you continued to change the I/O modules in the rack with the OPC Server configured to always generate new OPC tags on startup. Under this condition every time the communications driver detected a new I/O module, the tags would be added to the server. If the old tags are not removed, a number of unused tags could accumulate in the server's tag space. **Perform the following action** is used to tailor the server's operation to best fit the application's needs. Descriptions of the selections are as follows:

1. **Delete on create**, the default condition, allows the server to remove any tags that had previous been added to the tag space before the communications driver can add any new tags.

2. **Overwrite as necessary** allows the server to remove only tags the communications driver is replacing with new tags. Any tags that are not being overwritten will remain in the server's tag space.

3. **Do not overwrite** prevents the server from removing any tags that had been previous generated or may have already existed in the server. With this selection, the communications driver can only add tags that are completely new.

4. **Do not overwrite, log error**, has the same effect as the third with an addition: an error message will be posted to the OPC Server's event log when a tag overwrite would have occurred.

Note: The removal of OPC tags affects tags that have been automatically generated by the communications driver and any tags that have been added using names that match generated tags. It is recommended that users avoid adding own tags to the server using names that match tags that may be automatically generated by the driver.

Add generated tags to the following groupcan be used to keep automatically generated tags from mixing with tags that have been entered manually. This parameter is used to specify a sub group that will be used when adding all automatically generated tags for this device. The name of the sub group can be up to 31 characters in length. The following displays demonstrate how this parameter affects where automatically generated tags are placed in the server's tag space. As shown below, this parameter provides a root branch to which all automatically generated tags will be added.
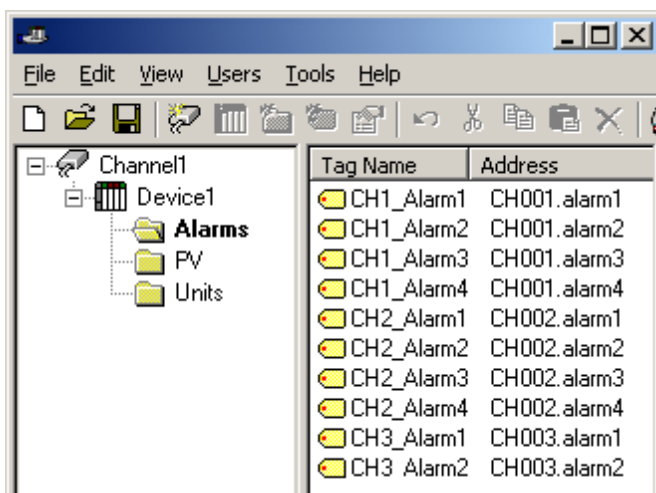
No sub group specified.
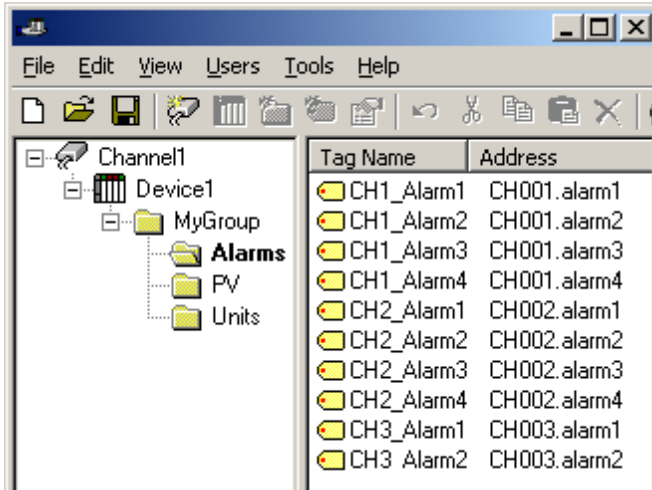
Sub group namedMyGroup specified.

Auto Create is used to manually initiate the creation of automatically generated OPC tags. It can be used to make the driver to reevaluate the device for possible tag changes. Auto Create can also be accessed from the System Tags for this device, allowing the OPC client application to initiate tag database creation.

MW100 Addressing

The driver supports the following addresses for this device. The default data type for each address type is shown in **bold**.

Measured Channels

Address Type	Format	Range	Data Types	Access
Process Value of Channel	CH**xxxxx** or C**Hxxxxx**.PV	00001-00060	**Double**, Float	Read Only
Alarm Summary of Channel	CH**xxxxx**.Alarm	00001-00060	**Short**, Word, Byte	Read Only
Alarm Level1 Status of Channel	CH**xxxxx**.Alarm1	00001-00060	**Short**, Word, Byte	Read Only
Alarm Level2 Status of Channel	CH**xxxxx**.Alarm2	00001-00060	**Short**, Word, Byte	Read Only
Alarm Level3 Status of Channel	CH**xxxxx**.Alarm3	00001-00060	**Short**, Word, Byte	Read Only
Alarm Level4 Status of Channel	CH**xxxxx**.Alarm4	00001-00060	**Short**, Word, Byte	Read Only
Alarm Level1 Setpoint*	CH**xxxxx**.ASP1	00001-00060	**Double**, Float	Read/Write
Alarm Level2 Setpoint*	CH**xxxxx**.ASP2	00001-00060	**Double**, Float	Read/Write
Alarm Level3 Setpoint*	CH**xxxxx**.ASP3	00001-00060	**Double**, Float	Read/Write
Alarm Level4 Setpoint*	CH**xxxxx**.ASP4	00001-00060	**Double**, Float	Read/Write
Upper Scale Value of Channel*	CH**xxxxx**.scale_Hi	00001-00060	**Double**, Float	Read Only
Lower Scale Value of Channel*	CH**xxxxx**.scale_Lo	00001-00060	**Double**, Float	Read Only
Status of Channel	CH**xxxxx**.status	00001-00060	**String**	Read Only
Tag name of Channel*	CH**xxxxx**.tag	00001-00060	**String**	Read Only
Unit String of Channel*	CH**xxxxx**.unit	00001-00060	**String**	Read Only
Precision of Channel*	CH**xxxxx**.Precision	00001-00060	**Short**, Word	Read Only
Precision of Channel*	CHA**xxxxx**.Precision	001-300	**Short**, Word	Read Only
Digital Input	CH**xxxxx**.DI	00001-00060	**Boolean**	Read Only
Alarm Type 1 for Channel (Num)	CH**xxxxx**.AlarmType1.Num	00001-00060	**Short**, Word, Byte	Read Only
Alarm Type 2 for Channel (Num)	CH**xxxxx**.AlarmType2.Num	00001-00060	**Short**, Word, Byte	Read Only
Alarm Type 3 for Channel (Num)	CH**xxxxx**.AlarmType3.Num	00001-00060	**Short**, Word, Byte	Read Only
Alarm Type 4 for Channel (Num)	CH**xxxxx**.AlarmType4.Num	00001-00060	**Short**, Word, Byte	Read Only
Alarm Type 1 for Channel (String)	CH**xxxxx**.AlarmType1.String	00001-00060	**String**	Read Only
Alarm Type 2 for Channel (String)	CH**xxxxx**.AlarmType2.String	00001-00060	**String**	Read Only
Alarm Type 3 for Channel (String)	CH**xxxxx**.AlarmType3.String	00001-00060	**String**	Read Only
Alarm Type 4 for Channel (String)	CH**xxxxx**.AlarmType4.String	00001-00060	**String**	Read Only
Process Value of Math	CHA**xxxxx**	001-300	**Double**,Float	Read Only
Process Value of Math	CHA**xxxxx**.PV	001-300	**Double**, Float	Read Only
Alarm Summary for Channel	CHA**xxxxx**.alarm	001-300	**Short**,Word, Byte	Read Only
Alarm 1 for Channel	CHA**xxxxx**.alarm1	001-300	**Short**, Word, Byte	Read Only
Alarm 2 for Channel	CHA**xxxxx**.alarm2	001-300	**Short**, Word, Byte	Read Only
Alarm 3 for Channel	CHA**xxxxx**.alarm3	001-300	**Short**, Word, Byte	Read Only
Alarm 4 for Channel	CHA**xxxxx**.alarm4	001-300	**Short**, Word, Byte	Read Only
Alarm Setpoint 1 for Channel	CHA**xxxxx**.ASP1	001-300	**Double**, Float	Read/Write
Alarm Setpoint 2 for Channel	CHA**xxxxx**.ASP2	001-300	**Double**, Float	Read/Write
Alarm Setpoint 3 for Channel	CHA**xxxxx**.ASP3	001-300	**Double**, Float	Read/Write
Alarm Setpoint 4 for Channel	CHA**xxxxx**.ASP4	001-300	**Double**, Float	Read/Write
Upper Scale Value for Channel	CHA**xxxxx**.scale_Hi	001-300	**Double**, Float	Read Only
Lower Scale Value for Channel	CHA**xxxxx**.scale_Lo	001-300	**Double**, Float	Read Only
Unit String of Channel	CHA**xxxxx**.unit	001-300	String	Read Only
Tagname of Channel	CHA**xxxxx**.tag	001-300	**String**	Read Only
Status of Channel	CHA**xxxxx**.status	001-300	**String**	Read Only
Last Math Channel	CHA.High	001-300	**Short**, Word	Read Only
First Math Channel	CHA.Low	001-300	**Short**, Word	Read Only
Alarm Type 1 for Channel (Numeric)	CHA**xxxxx**.AlarmType1.Num	001-300	Short	Read Only
Alarm Type 1 for Channel (String)	CHA**xxxxx**.AlarmType1.String	001-300	**String**	Read Only
Digital Output on Channel	CH**xxxxx**.DO	00001-00060	**Boolean**	Read/Write
Analog Output (Volt) on Channel**	CH**xxxxx**.AOVolt	00001-00060	**Float**	Read/Write
Analog Output (mA) on Channel**	CH**xxxxx**.AOmA	00001-00060	**Float**	Read/Write

Pulse Width Modulation on Channel	CHxxxxx.PWM	00001-00060	**Float**	Read/Write
Lowest Measuring Channel*	CH.Low		**Short**, Word	Read Only
Highest Measuring Channel*	CH.High		**Short**, Word	Read Only

*Initialized Data

Data associated with these addresses are read from the device only at the start of a communications session. Once read, the values will not be refreshed until the server has been restarted or the "Reset" tag has been invoked. To invoke a reset, a non zero value must be written to the Reset tag. Once the Reset tag has been invoked, the driver will reinitialize all startup data from the device. This will result in a pause of the update of process values and alarms until all initialized data has been read from the device.

**Channel Addressing

The Yokogawa MW Ethernet Device is arranged as a 6 slot (numbers 0-5) unit. Each slot may hold a module consisting of up to 10 channels (1-10). The first slot has possible channel numbers of CH00001-00010, the second slot CH000011-00020, the third slot CH00021-00030, the fourth slot CH00031-00040, the fifth slot CH00041-00050, and the last slot has possible channel numbers of 00051-00060. There is potential for gaps in channel numbering due to the fact that not all slots must contain a module, and not all modules are equipped with 10 channels (as in the case of a 4-channel module).

Tag Names

For channels that have unspecified tag names, the driver will construct an internal tag name based on the channel number. For example, the tag name of address 'CH00001' will be returned as 'CH00001'.

Digital Output

'CHxxxxx.DO' is only valid for channel numbers in slots containing a Digital Output Module.

Analog Output (Volt)

Depending upon how fast multiple writes are written to this tag, the response time during multiple and frequent writes might be slow.

General Device Data

Address Type	Format	Range	Data Types	Access
Date of Last Data	Date		**String**	Read Only
Time of Last Data	Time		**String**	Read Only
SetTime	SetTime		**Boolean**	Write Only
Model String	Model		**String**	Read Only
Math Communication Data	CDxxx	001-300	**Float**	Write Only
MathControl	Math Control		**Short**	Write Only
OperationCtrl	Operation Control		**Short**	Write Only
Serial Number of Unit	SerialNumber		**String**	Read Only
IP Address of Device	IP		**String**	Read Only
Reset Alarms	AlarmReset		**Boolean**, Short, Word, Byte	Write Only
Direct Reloading of Configuration	Reset		**Boolean**, Short, Word, Byte	Write Only
ClearError	Clear Error		**Boolean**	Write Only
Control Command and Response*	Command		**String**	Read/Write

*The Control Command and Response address type allows users to send a string command and receive a string response to and from the device. As such, users can send commands to the device that are not directly supported by the driver. This tag is only available to users logged in at the Administrator level; otherwise, write operations will return an error. Binary data is not supported.

Note 1: Addresses that have Write Only access are assigned a default access of Read/Write. However, data values are unreadable for these addresses and the associated tags are not included in the scan list. The current data value for these tags will always be 0 for numeric data types (with the exception of 'Disp.Interval,' which has a default value of 1000).

Note 2: The actual number of addresses available for of each type is dependent on the configuration of the Yokogawa device. If the driver finds that an address is not present in the device at Runtime, it will post an error message and remove the tag from its scan list.

Caution: It is recommended that users be cautious when performing write operations using the Command address.

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
Address '<address>' is out of range for the specified device or register
Data Type '<type>' is not valid for device address '<address>'
Device address '<address>' is Read Only

Device Status Messages
Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>
Device '<device>' responded with error '<error>'
Device '<device>' responded with error '<errnum>' (Tag '<address>')
Device '<device>' login failed. Check username and password
Device <device name> login failed. No more logins at this user level
Unable to start measuring for device '<device>'
Unable to start math for device '<device>'
Unable to set clock for device '<device>'
Device '<device>' returned error '2 <Value exceeds setting range>'. (Tag '<address>')

Driver Error Messages
Winsock initialization failed (OS Error = n)
Winsock V1.1 or higher must be installed to use the Yokogawa MW Ethernet Device driver

Address Validation

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
Address '<address>' is out of range for the specified device or register
Data Type '<type>' is not valid for device address '<address>'
Device address '<address>' is Read Only

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.

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

Error Type:
Warning

Possible Cause:

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

Solution:

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

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 Status Messages

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

Device Status Messages

Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>
Device '<device>' responded with error '<error>'
Device '<device>' responded with error '<errnum>' (Tag '<address>')
Device '<device>' login failed. Check username and password
Device <device name> login failed. No more logins at this user level
Unable to start measuring for device '<device>'
Unable to start math for device '<device>'
Unable to set clock for device '<device>'
Device '<device>' returned error '2 <Value exceeds setting range>'. (Tag '<address>')

Device '<device name>' is not responding

Error Type:

Serious

Possible Cause:

1. The connection between the device and the Host PC is broken.
2. The IP address assigned to the device is incorrect.
3. The device Ethernet port is already in use.
4. The connection cannot be established in the specified timeout period.
5. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify that the IP address given to the named device matches that configured in the actual device. If applicable, verify the subnet mask and default gateway settings configured in the actual device.
3. Check for another connected application (such as MW Standard software) and disconnect.

4. Increase the Connect Timeout value in the Timeout page of Device Properties.
5. Increase the Request Timeout setting so that the entire response can be handled.

Note:
The MW device's TCP/IP port supports a single connection.

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

Error Type:
Serious

Possible Cause:
1. The connection between the device and the Host PC is broken.
2. The named device may have been assigned an incorrect IP address.
3. The address specified may be Read Only or may not exist in the current device.

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

Device '<device>' responded with error '<error>'

Error Type:
Warning

Possible Cause:
The device has failed to automatically generate a tag.

Solution:
Look up the error number in the device manual.

Device '<device>' responded with error '<errnum>' (Tag '<address>')

Error Type:
Warning

Possible Cause:
An error has been returned by the device that is not specified in the driver.

Solution:
Look up the error number in the device manual.

Device '<device name>' login failed. Check username and password

Error Type:
Warning

Possible Cause:
An invalid username has been specified.

Solution:
Retry with a valid username and password.

Device '<device name>' login failed. No more logins at this user level

Error Type:
Warning

Possible Cause:
The device is already connected to the maximum number of clients that is allowed at a time.

Solution:
Wait until a connection becomes available for that device.

Unable to start measuring for device '<device>'

Error Type:
Warning

Possible Cause:
The driver has failed to start measuring on the device.

Solution:
1. Check connections to the device.
2. Measuring cannot be started when certain other features are already on. Check the device manual for details.

Unable to start math for device '<device>'

Error Type:
Warning

Possible Cause:
1. The driver has failed to start math on the device.
2. The math option is missing from the device.

Solution:
1. Check connections to the device.
2. Math cannot be started when certain other features are already on. Check the device manual for details.
3. Use a device with the math option.

Unable to set clock for device '<device>'

Error Type:
Warning

Possible Cause:
The driver has failed to set the clock in the device.

Solution:
1. Check connections to the device.
2. The device may have been busy performing other functions. Retry.

Device '<device>' returned error '2 <Value exceeds setting range>'. (Tag '<address>')

Error Type:
Warning

Possible Cause:
The driver is attempting to write a value to a tag that exceeds its limit.

Solution:
Retry writes that are within the accepted range.

Driver Error Messages

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

Driver Error Messages
Winsock initialization failed (OS Error = n)
Winsock V1.1 or higher must be installed to use the Yokogawa MW Ethernet Device driver

Winsock initialization failed (OS Error = n)

Error Type:
Fatal

OS Error	Indication	Possible Solution
10091	Indicates that the underlying network	Wait a few seconds and restart

	subsystem is not ready for network com-munication.	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 Yokogawa MW Ethernet Device driver

Error Type:
Fatal

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

Solution:
Upgrade Winsock to version 1.1 or higher.

Index

U

W