

# Yokogawa Darwin Ethernet Driver

© 2023 PTC Inc. All Rights Reserved.

# Table of Contents

|  |           |
|--|-----------|
| <b>Yokogawa Darwin Ethernet Driver</b> .....                                   | <b>1</b>  |
| <b>Table of Contents</b> .....   | <b>2</b>  |
| Yokogawa Darwin Ethernet Driver .....  | 4         |
| Overview .....   | 4         |
| <b>Setup</b> .....   | <b>5</b>  |
| Channel Properties — General .....   | 6         |
| Tag Counts .....   | 7         |
| Channel Properties — Ethernet Communications .....                             | 8         |
| Channel Properties — Write Optimizations .....                                 | 8         |
| Channel Properties — Advanced .....  | 9         |
| Device Properties — General .....  | 10        |
| Operating Mode .....   | 11        |
| Tag Counts .....   | 11        |
| Device Properties — Scan Mode .....  | 11        |
| Device Properties — Timing .....   | 12        |
| Device Properties — Auto-Demotion .....  | 13        |
| Device Properties — Tag Generation .....                                       | 14        |
| Device Properties — Device Configuration .....                                 | 17        |
| Device Properties — Redundancy .....   | 18        |
| <b>Optimizing Communications</b> .....   | <b>19</b> |
| <b>Data Types Description</b> .....  | <b>20</b> |
| <b>Address Descriptions</b> .....  | <b>21</b> |
| DA100-1 Addressing .....   | 21        |
| DA100-2 Addressing .....   | 24        |
| DR231 Addressing .....   | 28        |
| DR232 Addressing .....   | 31        |
| DR241 Addressing .....   | 35        |
| DR242 Addressing .....   | 38        |
| DR130 Addressing .....   | 42        |
| DC100-1 Addressing .....   | 45        |
| DC100-2 Addressing .....   | 48        |
| <b>Automatic Tag Database Generation</b> .....                                 | <b>53</b> |
| <b>Error Descriptions</b> .....  | <b>55</b> |
| Address '<address>' is out of range for the specified device or register ..... | 55        |
| Data Type '<type>' is not valid for device address '<address>' .....           | 55        |

|  |           |
|--|-----------|
| Device address '<address>' contains a syntax error .....   | 55        |
| Device address '<address>' is Read Only .....  | 56        |
| Missing address .....  | 56        |
| Device '<device name>' is not responding .....   | 56        |
| Unable to write to '<address>' on device '<device name>' .....                                   | 57        |
| Winsock initialization failed (OSError = n) .....  | 57        |
| Winsock V1.1 or higher must be installed to use the Yokogawa Darwin Ethernet device driver ..... | 57        |
| <b>Index .....</b>   | <b>58</b> |

## Yokogawa Darwin Ethernet Driver

---

Help version [1.024](#)

### CONTENTS

#### [Overview](#)

What is the Yokogawa Darwin Ethernet Driver?

#### [Setup](#)

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

#### [Optimizing Communications](#)

How do I get the best performance from the Yokogawa Darwin Ethernet Driver?

#### [Data Types Description](#)

What data types does this driver support?

#### [Address Descriptions](#)

How do I address a data location on a Yokogawa Darwin device?

#### [Automatic Tag Database Generation](#)

How can I easily configure tags for the Yokogawa Darwin Ethernet Driver?

#### [Error Descriptions](#)

What error messages does the Yokogawa Darwin Ethernet Driver produce?

### Overview

---

The Yokogawa Darwin Ethernet Driver provides a reliable way to connect Yokogawa Darwin Ethernet devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It is intended for use with Yokogawa Data Acquisition and Data Recorder devices that support Ethernet TCP communications.

---

## Setup

---

### Supported Yokogawa Devices

DA100-1, DA100-2  
DR231, DR232  
DR241, DR242  
DR130  
DC100-1, DC100-2

### Connection Timeout (s)


This property 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 (ms)

This property 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 5000 milliseconds.

### Retry Attempts

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

 For more information on timeouts and retries, refer to the server help file.

### 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 1024 per channel.

### Device ID

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

## 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     | [-] <b>Identification</b> |         |
| General             | Name                      |         |
| Write Optimizations | Description               |         |
| Advanced            | Driver                    |         |
|                     | [-] <b>Diagnostics</b>    |         |
|                     | Diagnostics Capture       | Disable |
|                     | [-] <b>Tag Counts</b>     |         |
|                     | 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 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.

## 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 — Ethernet Communications

Ethernet Communication can be used to communicate with devices.

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

### Ethernet Settings

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

## Channel Properties — Write Optimizations

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 |
| <b>Write Optimizations</b> | Duty Cycle          | 10                                   |
|                            |                     |                                      |

### Write Optimizations

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

- **Write All Values for All Tags:** This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.
- **Write Only Latest Value for Non-Boolean Tags:** Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.
  - **Note:** This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.
- **Write Only Latest Value for All Tags:** This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest



value to the device. This mode optimizes all writes by updating the tags currently in the write queue before they are sent. This is the default mode.

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

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

## Channel Properties — Advanced

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

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

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

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

● **Note:** This property is 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.

|   |                 |         |           |   |  |      |  |             |  |                    |  |        |  |       |  |           |         |    |   |
|---|-----------------|---------|-----------|---|--|------|--|-------------|--|--------------------|--|--------|--|-------|--|-----------|---------|----|---|
| <table border="1"> <tr><td>Property Groups</td></tr> <tr><td>General</td></tr> <tr><td>Scan Mode</td></tr> </table> | Property Groups | General | Scan Mode | <table border="1"> <tr><td><input checked="" type="checkbox"/> Identification</td></tr> <tr><td>Name</td><td></td></tr> <tr><td>Description</td><td></td></tr> <tr><td>Channel Assignment</td><td></td></tr> <tr><td>Driver</td><td></td></tr> <tr><td>Model</td><td></td></tr> <tr><td>ID Format</td><td>Decimal</td></tr> <tr><td>ID</td><td>2</td></tr> </table> | <input checked="" type="checkbox"/> Identification | Name |  | Description |  | Channel Assignment |  | Driver |  | Model |  | ID Format | Decimal | ID | 2 |
| Property Groups   |                 |         |           |   |  |      |  |             |  |                    |  |        |  |       |  |           |         |    |   |
| General   |                 |         |           |   |  |      |  |             |  |                    |  |        |  |       |  |           |         |    |   |
| Scan Mode   |                 |         |           |   |  |      |  |             |  |                    |  |        |  |       |  |           |         |    |   |
| <input checked="" type="checkbox"/> Identification  |                 |         |           |   |  |      |  |             |  |                    |  |        |  |       |  |           |         |    |   |
| Name  |                 |         |           |   |  |      |  |             |  |                    |  |        |  |       |  |           |         |    |   |
| 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 help 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 |        |
| Scan 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. 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. When a device is simulated, updates may not appear faster than one (1) second 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                              |

**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 | <input type="checkbox"/> <b>Communication Timeouts</b> |      |
| General         | Connect Timeout (s)                                    | 3    |
| Scan Mode       | Request Timeout (ms)                                   | 1000 |
| <b>Timing</b>   | Attempts Before Timeout                                | 3    |

### 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 9999 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 | [-] <b>Timing</b>        |   |
| General         | Inter-Request Delay (ms) | 0 |
| Scan Mode       |                          |   |
| <b>Timing</b>   |                          |   |

## Device Properties — Auto-Demotion

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

|                      |                               |          |
|----------------------|-------------------------------|----------|
| Property Groups      | [-] <b>Auto-Demotion</b>      |          |
| General              | Demote on Failure             | Enable ▾ |
| Scan Mode            | Timeouts to Demote            | 3        |
| Timing               | Demotion Period (ms)          | 10000    |
| <b>Auto-Demotion</b> | Discard Requests when Demoted | Disable  |

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

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

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

**Demotion Period:** Indicate how long the device should be placed off-scan when the timeouts value is reached. During this period, no read requests are sent to the device and all data associated with the read

requests are set to bad quality. When this period expires, the driver places the device on-scan and allows for another attempt at communications. The valid range is 100 to 3600000 milliseconds. The default is 10000 milliseconds.

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

## Device Properties — Tag Generation

The automatic tag database generation features make setting up an application a plug-and-play operation. Select communications drivers can be configured to automatically build a list of tags that correspond to device-specific data. These automatically generated tags (which depend on the nature of the supporting driver) can be browsed from the clients.

⚙️ *Not all devices and drivers support full automatic tag database generation and not all support the same data types. Consult the data types descriptions or the supported data type lists for each driver for specifics.*

If the target device supports its own local tag database, the driver reads the device's tag information and uses the data to generate tags within the server. If the device does not natively support named tags, the driver creates a list of tags based on driver-specific information. An example of these two conditions is as follows:

1. If a data acquisition system supports its own local tag database, the communications driver uses the tag names found in the device to build the server's tags.
2. If an Ethernet I/O system supports detection of its own available I/O module types, the communications driver automatically generates tags in the server that are based on the types of I/O modules plugged into the Ethernet I/O rack.

⚙️ **Note:** Automatic tag database generation's mode of operation is completely configurable. *For more information, refer to the property descriptions below.*

|                       |   |                            |
|-----------------------|---|----------------------------|
| Property Groups       | ☐ <b>Tag Generation</b>                 |                            |
| General               | On Device Startup                       | Do Not Generate on Startup |
| Scan Mode             | On Duplicate Tag                        | Delete on Create           |
| Timing                | Parent Group                            |                            |
| Auto-Demotion         | Allow Automatically Generated Subgroups | Enable                     |
| <b>Tag Generation</b> | Create                                  | Create tags                |
| Communications        |   |                            |
| Redundancy            |   |                            |

**On Property Change:** If the device supports automatic tag generation when certain properties change, the **On Property Change** option is shown. It is set to **Yes** by default, but it can be set to **No** to control over when tag generation is performed. In this case, the **Create tags** action must be manually invoked to perform tag generation. To invoke via the Configuration API service, access `/config/v1/project/channels/{name}/devices/{name}/services/TagGeneration`.

**On Device Startup:** Specify when OPC tags are automatically generated. Descriptions of the options are as follows:

- **Do Not Generate on Startup:** This option prevents the driver from adding any OPC tags to the tag space of the server. This is the default setting.
- **Always Generate on Startup:** This option causes the driver to evaluate the device for tag information. It also adds tags to the tag space of the server every time the server is launched.
- **Generate on First Startup:** This option causes the driver to evaluate the target device for tag information the first time the project is run. It also adds any OPC tags to the server tag space as needed.

● **Note:** When the option to automatically generate OPC tags is selected, any tags that are added to the server's tag space must be saved with the project. Users can configure the project to automatically save from the **Tools | Options** menu.

**On Duplicate Tag:** When automatic tag database generation is enabled, the server needs to know what to do with the tags that it may have previously added or with tags that have been added or modified after the communications driver since their original creation. This setting controls how the server handles OPC tags that were automatically generated and currently exist in the project. It also prevents automatically generated tags from accumulating in the server.

For example, if a user changes the I/O modules in the rack with the server configured to **Always Generate on Startup**, new tags would be added to the server every time the communications driver detected a new I/O module. If the old tags were not removed, many unused tags could accumulate in the server's tag space. The options are:

- **Delete on Create:** This option deletes any tags that were previously added to the tag space before any new tags are added. This is the default setting.
- **Overwrite as Necessary:** This option instructs the server to only remove the tags that the communications driver is replacing with new tags. Any tags that are not being overwritten remain in the server's tag space.
- **Do not Overwrite:** This option prevents the server from removing any tags that were previously generated or already existed in the server. The communications driver can only add tags that are completely new.
- **Do not Overwrite, Log Error:** This option has the same effect as the prior option, and also posts an error message to the server's Event Log when a tag overwrite would have occurred.

● **Note:** Removing OPC tags affects tags that have been automatically generated by the communications driver as well as any tags that have been added using names that match generated tags. Users should avoid adding tags to the server using names that may match tags that are automatically generated by the driver.

**Parent Group:** This property keeps automatically generated tags from mixing with tags that have been entered manually by specifying a group to be used for automatically generated tags. The name of the group can be up to 256 characters. This parent group provides a root branch to which all automatically generated tags are added.

**Allow Automatically Generated Subgroups:** This property controls whether the server automatically creates subgroups for the automatically generated tags. This is the default setting. If disabled, the server generates the device's tags in a flat list without any grouping. In the server project, the resulting tags are named with the address value. For example, the tag names are not retained during the generation process.

● **Note:** If, as the server is generating tags, a tag is assigned the same name as an existing tag, the system automatically increments to the next highest number so that the tag name is not duplicated. For example, if the generation process creates a tag named "AI22" that already exists, it creates the tag as "AI23" instead.

**Create:** Initiates the creation of automatically generated OPC tags. If the device's configuration has been modified, **Create tags** forces the driver to reevaluate the device for possible tag changes. Its ability to be accessed from the System tags allows a client application to initiate tag database creation.

• **Note:** **Create tags** is disabled if the Configuration edits a project offline.



## Device Properties — Device Configuration

|                             |   |                         |
|-----------------------------|---|-------------------------|
| Property Groups             | <input type="checkbox"/> <b>General</b>       |                         |
| General                     | Port  | Ethernet Exclusive Port |
| Scan Mode                   | Special Data Handling                         | None                    |
| Timing                      | Polling Interval (ms)                         | 1000                    |
| Auto-Demotion               | Start Math on Start                           | Disable                 |
| Tag Generation              | <input type="checkbox"/> <b>Time Settings</b> |                         |
| <b>Device Configuration</b> | Date and Time                                 | Device Time             |
| Redundancy                  | Date Format                                   | MM/DD/YY                |
|                             | Set Clock on Start                            | Disable                 |
|                             | <input type="checkbox"/> <b>Tagnames</b>      |                         |
|                             | Generate Tag Database Using                   | Physical Channel Number |

### General

**Port:** This property specifies the port number that the remote device will use. Options include Ethernet Exclusive Port and Ethernet Shared Port. The default setting is Ethernet Exclusive Port. Descriptions of the properties are as follows:

- **Ethernet Exclusive Port:** When selected, this port does not allow any other machines or host to connect to the Darwin system. Port number 34150 only supports a single connected host (such as this driver) or the DAQ32 software. If a host is attached to the 34150 port when the Ethernet Exclusive port is selected, the Darwin system cannot be communicated with until the connected host releases the port.
  - **Ethernet Shared Port:** When selected, this port allows more than one OPC server to connect to the Darwin system. Port number 34151 supports up to four simultaneous connections. This driver must be able to communicate with the Ethernet Exclusive Port at least once when establishing basic driver initialization to use this port. The DAQ32 software must be taken offline to allow this driver to initialize during OPC server start. In an ideal system, this driver would be configured to use the Ethernet Shared Port so that the Ethernet Exclusive Port is available for the DAQ32 software.
- **Note:** This driver requires Winsock V1.1 or higher.

**Special Data Handling:** This property allows the driver to be configured to return specific data values for the numerical out of range and error conditions returned from the device. Options include None, +INF, and -INF. The default setting is None. *For more information, refer to [Special Data Handling](#).*

**Polling Interval (ms):** This property specifies a fixed time interval for all communications with a device. It can be used to prevent the driver from making excessive requests to the device, and to prevent the OPC client from forcing the driver to run at its maximum update rate. The default setting is 1000 milliseconds.

**Start Math on 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. The default setting is unchecked.

### Time Settings

**Date and Time:** This property 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. The default setting is Device Time. Descriptions of the options are as follows:

- **Device Time:** When selected, the Date and Time tags will return the date and time read from the device. This date and time represents the date and time that the latest data was measured or computed based on the internal device clock.
- **System Time:** When selected, the Date and Time tags will return the date and time that the requested data was returned from the device based on the PC system clock.

**Date Format:** This property specifies the format of the return string for the Date data type. Options include MM/DD/YY (month/day/year), YY/MM/DD (year/month/day), or DD/MM/YY (day/month/year). The default setting is MM/DD/YY.

**Set Clock on Start:** When checked, this option informs 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. The default setting is unchecked.

## Tagnames

**Generate Tag Database Using:** This property specifies the origin of the tag name used when automatically generating a tag database. Options include Physical Channel Number and Device Tag Name. The default setting is Physical Channel Number. Descriptions of the options are as follows:

- **Physical Channel Number:** When selected, the driver will generate tag names based on the channel number of an item. For example, "CH001" or "CH001\_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".

## Device Properties — Redundancy

|                     |  |                      |
|---------------------|--|----------------------|
| Property Groups     | <input type="checkbox"/> <b>Redundancy</b> |                      |
| General             | Secondary Path                             | Channel.Device 1 ... |
| Scan Mode           | Operating Mode                             | Switch On Failure    |
| Timing              | Monitor Item                               |                      |
| Auto-Demotion       | Monitor Interval (s)                       | 300                  |
| Tag Generation      | Return to Primary ASAP                     | Yes                  |
| Tag Import Settings |  |                      |
| <b>Redundancy</b>   |  |                      |

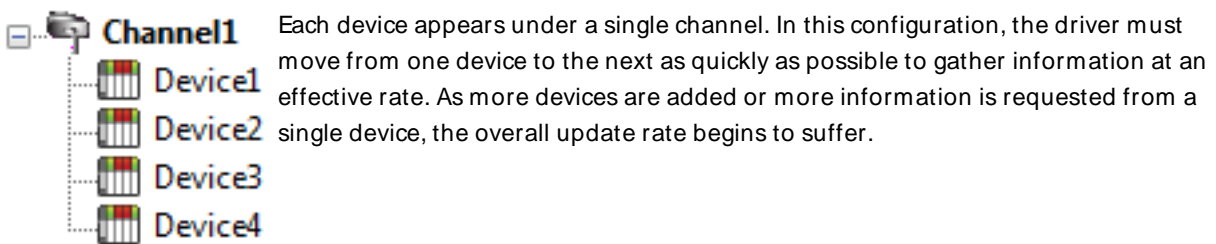
Redundancy is available with the Media-Level Redundancy Plug-In.

• Consult the website, a sales representative, or the [user manual](#) for more information.

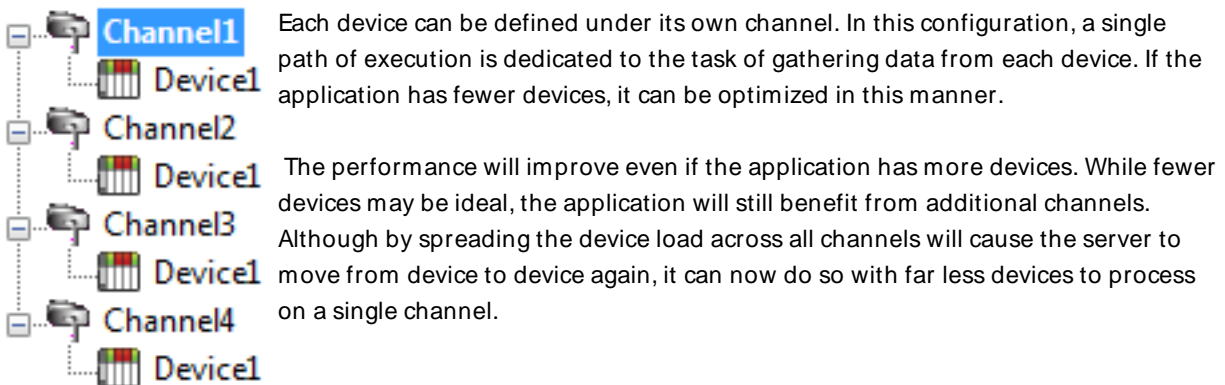
## Optimizing Communications

The Yokogawa Darwin 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 Darwin Ethernet Driver is fast, there are a couple of guidelines that can be used to control and optimize the application and gain maximum performance.

This server refers to communications protocols like Yokogawa Darwin Ethernet as a channel. Each channel defined in the application represents a separate path of execution in the server. Once a channel has been defined, a series of devices must then be defined under that channel. Each of these devices represents a single 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 Darwin Ethernet Driver or the network. An example of how the application may appear when configured using a single channel is shown below.



If the Yokogawa Darwin Ethernet Driver could only define one single channel, then the example shown above would be the only option available; however, the Yokogawa Darwin Ethernet 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.



## Data Types Description

---

| Data Type | Description   |
|-----------|---|
| Boolean   | Single bit  |
| Byte      | Unsigned 8-bit value<br>bit 0 is the low bit<br>bit 7 is the high bit                           |
| Word      | Unsigned 16-bit value<br>bit 0 is the low bit<br>bit 15 is the high bit                         |
| Short     | Signed 16-bit value<br>bit 0 is the low bit<br>bit 14 is the high bit<br>bit 15 is the sign bit |
| Float     | 32-bit floating point value<br>bit 0 is the low bit<br>bit 31 is the high bit                   |
| Double    | 64-bit floating point value<br>bit 0 is the low bit<br>bit 63 is the high bit                   |
| String    | Null-terminated ASCII string  |

## Address Descriptions

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

[DA100-1 Addressing](#)

[DA100-2 Addressing](#)

[DR231 Addressing](#)

[DR232 Addressing](#)

[DR241 Addressing](#)

[DR242 Addressing](#)

[DR130 Addressing](#)

[DC100-1 Addressing](#)

[DC100-2 Addressing](#)

## DA100-1 Addressing

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

### Measured Channels

| Address Type  | Format                   | Range   | Data Types                | Access    |
|---|--------------------------|---------|---------------------------|-----------|
| Process Value of Channel                                | CHxxx or CHxxx.PV        | 001-060 | <b>Double</b> , Float     | Read Only |
| Alarm Summary of Channel                                | CHxxx.Alarm              | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level1 Status of Channel                          | CHxxx.Alarm 1            | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level2 Status of Channel                          | CHxxx.Alarm2             | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level3 Status of Channel                          | CHxxx.Alarm 3            | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level4 Status of Channel                          | CHxxx.Alarm 4            | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level1 of Channel*<br>(numeric, ex. 0 = Off) | CHxxx.Alarm Type1.Num    | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level2 of Channel*<br>(numeric)              | CHxxx.Alarm Type2.Num    | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level3 of Channel*<br>(numeric)              | CHxxx.Alarm Type3.Num    | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level4 of Channel*<br>(numeric)              | CHxxx.Alarm Type4.Num    | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level1 of Channel*                           | CHxxx.Alarm Type1.String | 001-060 | <b>String</b>             | Read Only |

| Address Type                              | Format                  | Range   | Data Types                | Access     |
|---|-------------------------|---------|---------------------------|------------|
| (string, ex. "OFF")                       |                         |         |                           |            |
| Alarm Type Level2 of Channel*<br>(string) | CHxxx.AlarmType2.String | 001-060 | <b>String</b>             | Read Only  |
| Alarm Type Level3 of Channel*<br>(string) | CHxxx.AlarmType3.String | 001-060 | <b>String</b>             | Read Only  |
| Alarm Type Level4 of Channel*<br>(string) | CHxxx.AlarmType4.String | 001-060 | <b>String</b>             | Read Only  |
| Set and Read Level1 Alarm Set-point       | CHxxx.ASP1              | 001-060 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Set-point       | CHxxx.ASP2              | 001-060 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Set-point       | CHxxx.ASP3              | 001-060 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Set-point       | CHxxx.ASP4              | 001-060 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Channel*             | CHxxx.scale_Hi          | 001-060 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Channel*             | CHxxx.scale_Lo          | 001-060 | <b>Double</b> , Float     | Read Only  |
| Unit String of Channel*                   | CHxxx.unit              | 001-060 | <b>String</b>             | Read Only  |
| Tagname of Channel*                       | CHxxx.tag               | 001-060 | <b>String</b>             | Read Only  |
| Status of Channel*                        | CHxxx.status            | 001-060 | <b>String</b>             | Read Only  |
| Lowest Measuring Channel*                 | CH.Low                  |         | <b>Short</b> , Word, Byte | Read Only  |
| Highest Measuring Channel*                | CH.High                 |         | <b>Short</b> , Word, Byte | Read Only  |

### Math Channels

| Address Type                        | Format            | Range | Data Types                | Access     |
|-------------------------------------|-------------------|-------|---------------------------|------------|
| Process Value of Math Channel       | CHAxX or CHAxX.PV | 01-30 | <b>Double</b> , Float     | Read Only  |
| Alarm Summary of Math Channel       | CHAxX.Alarm       | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level1 Status of Math Channel | CHAxX.Alarm1      | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level2 Status of Math Channel | CHAxX.Alarm2      | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level3 Status of Math Channel | CHAxX.Alarm3      | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level4 Status of Math Channel | CHAxX.Alarm4      | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Set and Read Level1 Alarm Setpoint  | CHAxX.ASP1        | 01-30 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Setpoint  | CHAxX.ASP2        | 01-30 | <b>Double</b> , Float     | Read/Write |

| Address Type                       | Format         | Range | Data Types                | Access     |
|------------------------------------|----------------|-------|---------------------------|------------|
| Set and Read Level3 Alarm Setpoint | CHAxX.ASP3     | 01-30 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Setpoint | CHAxX.ASP4     | 01-30 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Math Channel* | CHAxX.scale_Hi | 01-30 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Math Channel* | CHAxX.scale_Lo | 01-30 | <b>Double</b> , Float     | Read Only  |
| Unit String of Math Channel*       | CHAxX.unit     | 01-30 | <b>String</b>             | Read Only  |
| Tagname of Math Channel*           | CHAxX.tag      | 01-30 | <b>String</b>             | Read Only  |
| Status of Math Channel*            | CHAxX.status   | 01-30 | <b>String</b>             | Read Only  |
| Lowest Math Channel*               | CHA.Low        |       | <b>Short</b> , Word, Byte | Read Only  |
| Highest Math Channel*              | CHA.High       |       | <b>Short</b> , Word, Byte | Read Only  |

\* Data associated with these addresses are only read at the device 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.

### Alarm Setpoints

Data values for Alarm Setpoints that are undefined in the device will be returned as +INF. Data values can only be written to Alarm Setpoints that are defined in the device. Write operations to undefined Alarm Setpoints will return an error.

### Scales

Data values for Scale\_Hi and Scale\_Lo for channels that are skipped will be returned as +INF.

### Tag Names

For devices that do not support tag names and 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 'CH001' will be returned as 'CH001'.

### General Device Data

| Address Type                      | Format      | Range   | Data Types                | Access     |
|-----------------------------------|-------------|---------|---------------------------|------------|
| Date of Last Data                 | Date        |         | <b>String</b>             | Read Only  |
| Time of Last Data                 | Time        |         | <b>String</b>             | Read Only  |
| Model Name of Device*             | Model       |         | <b>String</b>             | Read Only  |
| Digital Output (Main Unit)        | DOxxx       | 001-060 | <b>Boolean</b>            | Read/Write |
| Digital Output (Internal Switch)  | DOSxx       | 01-60   | <b>Boolean</b>            | Read/Write |
| Math Communication Data*          | CDxx        | 01-30   | <b>Short</b> , Word, Byte | Write Only |
| Control Math Execution            | MathControl |         | <b>Short</b> , Word, Byte | Write Only |
| Reset Alarms                      | AlarmReset  |         | <b>Boolean</b>            | Write Only |
| Control Command and Response      | Command     |         | <b>String</b>             | Read/Write |
| Direct Reloading of Configuration | Reset       |         | <b>Boolean</b>            | Write Only |
| SetTime*                          | Tag         |         | <b>Boolean</b>            | Write Only |

\* The SetTime tag will cause the device time to be updated. Writing 0 or 1 to the tag will update the Device Date and Time which can be verified from the Date tag and the Time tag. The SetTime tag will always display

0 because it is a Write Only tag. After a successful update, the following message will be posted: "Device Clock set to system time [Device <device\_name>]."

● **Note:** The SetTime tag requires that the device's Port property be set to Ethernet Exclusive Port. For more information, refer to [Communications Properties](#).

### Math Communication Data

The CD address type is only valid for devices equipped with the math option and write operations to CD addresses for non-math equipped devices will return an error. The range of valid data for addresses 'CDxx' is -32000 to 32000. CD data values are converted in the device from a whole number to a real number based on the decimal point placement specified for the associated math channel. For example, a data value of 32000 written to a CD address location with a decimal point placement of .000 would be interpreted by the device as 32.000

### Model Name of Device

The Model address type that returns the model name of the device may return 'DR231' for models of type DR241 and 'DR232' for models of type DR242.

### Control Math Execution

The MathControl address type is only available for devices equipped with the math option and write operations to the MathControl tag for non-math equipped devices will return an error.

### Control Command and Response

The Command address allows the user to send a string command and receive a string response to and from the device. This allows the user to send any command to the device, including commands not directly supported by the driver.

● **Caution:** Write operations using the Command address should be performed with care.

#### ● Notes:

1. The actual number of addresses available for of each type depends on the configuration of the Yokogawa device. If at runtime the driver finds that an address is not present in the device, the driver will post an error message and remove the tag from its scan list.
2. 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 and null string for string data types.

## DA100-2 Addressing

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

### Measured Channels

| Address Type             | Format            | Range   | Data Types                | Access    |
|--------------------------|-------------------|---------|---------------------------|-----------|
| Process Value of Channel | CHxxx or CHxxx.PV | 001-560 | <b>Double</b> , Float     | Read Only |
| Alarm Summary of Channel | CHxxx.Alarm       | 001-560 | <b>Short</b> , Word, Byte | Read Only |



| Address Type  | Format                  | Range   | Data Types                | Access     |
|---|-------------------------|---------|---------------------------|------------|
| Alarm Level1 Status of Channel                          | CHxxx.Alarm1            | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level2 Status of Channel                          | CHxxx.Alarm2            | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level3 Status of Channel                          | CHxxx.Alarm3            | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level4 Status of Channel                          | CHxxx.Alarm4            | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level1 of Channel*<br>(numeric, ex. 0 = Off) | CHxxx.AlarmType1.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level2 of Channel*<br>(numeric)              | CHxxx.AlarmType2.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level3 of Channel*<br>(numeric)              | CHxxx.AlarmType3.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level4 of Channel*<br>(numeric)              | CHxxx.AlarmType4.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level1 of Channel*<br>(string, ex. "OFF")    | CHxxx.AlarmType1.String | 001-560 | <b>String</b>             | Read Only  |
| Alarm Type Level2 of Channel*<br>(string)               | CHxxx.AlarmType2.String | 001-560 | <b>String</b>             | Read Only  |
| Alarm Type Level3 of Channel*<br>(string)               | CHxxx.AlarmType3.String | 001-560 | <b>String</b>             | Read Only  |
| Alarm Type Level4 of Channel*<br>(string)               | CHxxx.AlarmType4.String | 001-560 | <b>String</b>             | Read Only  |
| Set and Read Level1 Alarm Set-point                     | CHxxx.ASP1              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Set-point                     | CHxxx.ASP2              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Set-point                     | CHxxx.ASP3              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Set-point                     | CHxxx.ASP4              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Channel*                           | CHxxx.scale_Hi          | 001-560 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Channel*                           | CHxxx.scale_Lo          | 001-560 | <b>Double</b> , Float     | Read Only  |
| Unit String of Channel*                                 | CHxxx.unit              | 001-    | <b>String</b>             | Read Only  |

| Address Type               | Format       | Range   | Data Types                | Access    |
|----------------------------|--------------|---------|---------------------------|-----------|
|                            |              | 560     |                           |           |
| Tagname of Channel*        | CHxxx.tag    | 001-560 | <b>String</b>             | Read Only |
| Status of Channel*         | CHxxx.status | 001-560 | <b>String</b>             | Read Only |
| Lowest Measuring Channel*  | CH.Low       |         | <b>Short</b> , Word, Byte | Read Only |
| Highest Measuring Channel* | CH.High      |         | <b>Short</b> , Word, Byte | Read Only |

### Math Channels

| Address Type                        | Format            | Range | Data Types                | Access     |
|-------------------------------------|-------------------|-------|---------------------------|------------|
| Process Value of Math Channel       | CHAxX or CHAxX.PV | 01-60 | <b>Double</b> , Float     | Read Only  |
| Alarm Summary of Math Channel       | CHAxX.Alarm       | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level1 Status of Math Channel | CHAxX.Alarm1      | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level2 Status of Math Channel | CHAxX.Alarm2      | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level3 Status of Math Channel | CHAxX.Alarm3      | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level4 Status of Math Channel | CHAxX.Alarm4      | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Set and Read Level1 Alarm Setpoint  | CHAxX.ASP1        | 01-60 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Setpoint  | CHAxX.ASP2        | 01-60 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Setpoint  | CHAxX.ASP3        | 01-60 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Setpoint  | CHAxX.ASP4        | 01-60 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Math Channel*  | CHAxX.scale_Hi    | 01-60 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Math Channel*  | CHAxX.scale_Lo    | 01-60 | <b>Double</b> , Float     | Read Only  |
| Unit String of Math Channel*        | CHAxX.unit        | 01-60 | <b>String</b>             | Read Only  |
| Tagname of Math Channel*            | CHAxX.tag         | 01-60 | <b>String</b>             | Read Only  |
| Status of Math Channel*             | CHAxX.status      | 01-60 | <b>String</b>             | Read Only  |
| Lowest Math Channel*                | CHA.Low           |       | <b>Short</b> , Word, Byte | Read Only  |
| Highest Math Channel*               | CHA.High          |       | <b>Short</b> , Word, Byte | Read Only  |

\* Data associated with these addresses are only read at the device 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.

### Alarm Setpoints

Data values for Alarm Setpoints that are undefined in the device will be returned as +INF. Data values can only be written to Alarm Setpoints that are defined in the device. Write operations to undefined Alarm Setpoints will return an error.

### Scales

Data values for Scale\_Hi and Scale\_Lo for channels that are skipped will be returned as +INF.

## Tag Names

For devices that do not support tag names and 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 'CH001' will be returned as 'CH001'.

## 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  |
| Model Name of Device*             | Model       |         | String            | Read Only  |
| Digital Output (Main Unit)        | DOlxx       | 01-60   | Boolean           | Read/Write |
| Digital Output (Internal Switch)  | DOSxx       | 01-60   | Boolean           | Read/Write |
| Digital Output (Sub Units)        | DOxxx       | 001-560 | Boolean           | Read/Write |
| Math Communication Data*          | CDxx        | 01-60   | Short, Word, Byte | Write Only |
| Control Math Execution            | MathControl |         | Short, Word, Byte | Write Only |
| Reset Alarms                      | AlarmReset  |         | Boolean           | Write Only |
| Control Command and Response      | Command     |         | String            | Read/Write |
| Direct Reloading of Configuration | Reset       |         | Boolean           | Write Only |
| SetTime*                          | Tag         |         | Boolean           | Write Only |

\* The SetTime tag will cause the device time to be updated. Writing 0 or 1 to the tag will update the Device Date and Time which can be verified from the Date tag and the Time tag. The SetTime tag will always display 0 because it is a Write Only tag. After a successful update, the following message will be posted: "Device Clock set to system time [Device <device\_name>]."

**Note:** The SetTime tag requires that the device's Port property be set to Ethernet Exclusive Port. For more information, refer to [Communications Properties](#).

## Math Communication Data

The CD address type is only valid for devices equipped with the math option and write operations to CD addresses for non-math equipped devices will return an error. The range of valid data for addresses 'CDxx' is -32000 to 32000. CD data values are converted in the device from a whole number to a real number based on the decimal point placement specified for the associated math channel. For example, a data value of 32000 written to a CD address location with a decimal point placement of .000 would be interpreted by the device as 32.000

## Model Name of Device

The Model address type that returns the model name of the device may return 'DR231' for models of type DR241 and 'DR232' for models of type DR242.

## Control Math Execution

The MathControl address type is only available for devices equipped with the math option and write operations to the MathControl tag for non-math equipped devices will return an error.

## Control Command and Response

The Command address allows the user to send a string command and receive a string response to and from the device. This allows the user to send any command to the device, including commands not directly supported by the driver.

**Caution:** Write operations using the Command address should be performed with care.

**Notes:**

1. The actual number of addresses available for of each type depends on the configuration of the Yokogawa device. If at runtime the driver finds that an address is not present in the device, the driver will post an error message and remove the tag from its scan list.
2. 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 and null string for string data types.

## DR231 Addressing

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

### Measured Channels

| Address Type  | Format                   | Range   | Data Types                | Access    |
|---|--------------------------|---------|---------------------------|-----------|
| Process Value of Channel                                | CHxxx or CHxxx.PV        | 001-040 | <b>Double</b> , Float     | Read Only |
| Alarm Summary of Channel                                | CHxxx.Alarm              | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level1 Status of Channel                          | CHxxx.Alarm 1            | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level2 Status of Channel                          | CHxxx.Alarm2             | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level3 Status of Channel                          | CHxxx.Alarm3             | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level4 Status of Channel                          | CHxxx.Alarm4             | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level1 of Channel*<br>(numeric, ex. 0 = Off) | CHxxx.Alarm Type1.Num    | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level2 of Channel*<br>(numeric)              | CHxxx.Alarm Type2.Num    | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level3 of Channel*<br>(numeric)              | CHxxx.Alarm Type3.Num    | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level4 of Channel*<br>(numeric)              | CHxxx.Alarm Type4.Num    | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level1 of Channel*                           | CHxxx.Alarm Type1.String | 001-040 | <b>String</b>             | Read Only |

| Address Type                              | Format                  | Range   | Data Types                | Access     |
|---|-------------------------|---------|---------------------------|------------|
| (string, ex. "OFF")                       |                         |         |                           |            |
| Alarm Type Level2 of Channel*<br>(string) | CHxxx.AlarmType2.String | 001-040 | <b>String</b>             | Read Only  |
| Alarm Type Level3 of Channel*<br>(string) | CHxxx.AlarmType3.String | 001-040 | <b>String</b>             | Read Only  |
| Alarm Type Level4 of Channel*<br>(string) | CHxxx.AlarmType4.String | 001-040 | <b>String</b>             | Read Only  |
| Set and Read Level1 Alarm Set-point       | CHxxx.ASP1              | 001-040 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Set-point       | CHxxx.ASP2              | 001-040 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Set-point       | CHxxx.ASP3              | 001-040 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Set-point       | CHxxx.ASP4              | 001-040 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Channel*             | CHxxx.scale_Hi          | 001-040 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Channel*             | CHxxx.scale_Lo          | 001-040 | <b>Double</b> , Float     | Read Only  |
| Unit String of Channel*                   | CHxxx.unit              | 001-040 | <b>String</b>             | Read Only  |
| Tagname of Channel*                       | CHxxx.tag               | 001-040 | <b>String</b>             | Read Only  |
| Status of Channel*                        | CHxxx.status            | 001-040 | <b>String</b>             | Read Only  |
| Lowest Measuring Channel*                 | CH.Low                  |         | <b>Short</b> , Word, Byte | Read Only  |
| Highest Measuring Channel*                | CH.High                 |         | <b>Short</b> , Word, Byte | Read Only  |

### Math Channels

| Address Type                        | Format            | Range | Data Types                | Access     |
|-------------------------------------|-------------------|-------|---------------------------|------------|
| Process Value of Math Channel       | CHAxX or CHAxX.PV | 01-30 | <b>Double</b> , Float     | Read Only  |
| Alarm Summary of Math Channel       | CHAxX.Alarm       | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level1 Status of Math Channel | CHAxX.Alarm1      | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level2 Status of Math Channel | CHAxX.Alarm2      | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level3 Status of Math Channel | CHAxX.Alarm3      | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level4 Status of Math Channel | CHAxX.Alarm4      | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Set and Read Level1 Alarm Setpoint  | CHAxX.ASP1        | 01-30 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Setpoint  | CHAxX.ASP2        | 01-30 | <b>Double</b> , Float     | Read/Write |

| Address Type                       | Format          | Range | Data Types        | Access     |
|------------------------------------|-----------------|-------|-------------------|------------|
| Set and Read Level3 Alarm Setpoint | CHAx.x.ASP3     | 01-30 | Double, Float     | Read/Write |
| Set and Read Level4 Alarm Setpoint | CHAx.x.ASP4     | 01-30 | Double, Float     | Read/Write |
| Upper Scale Value of Math Channel* | CHAx.x.scale_Hi | 01-30 | Double, Float     | Read Only  |
| Lower Scale Value of Math Channel* | CHAx.x.scale_Lo | 01-30 | Double, Float     | Read Only  |
| Unit String of Math Channel*       | CHAx.x.unit     | 01-30 | String            | Read Only  |
| Tagname of Math Channel*           | CHAx.x.tag      | 01-30 | String            | Read Only  |
| Status of Math Channel*            | CHAx.x.status   | 01-30 | String            | Read Only  |
| Lowest Math Channel*               | CHA.Low         |       | Short, Word, Byte | Read Only  |
| Highest Math Channel*              | CHA.High        |       | Short, Word, Byte | Read Only  |

\*Data associated with these addresses are only read at the device 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.

### Alarm Setpoints

Data values for Alarm Setpoints that are undefined in the device will be returned as +INF. Data values can only be written to Alarm Setpoints that are defined in the device. Write operations to undefined Alarm Setpoints will return an error.

### Scales

Data values for Scale\_Hi and Scale\_Lo for channels that are skipped will be returned as +INF.

### Tag Names

For devices that do not support tag names and 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 'CH001' will be returned as 'CH001'.

### 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  |
| Model Name of Device*             | Model       |       | String            | Read Only  |
| Math Communication Data*          | CDxx        | 01-30 | Short, Word, Byte | Write Only |
| Control Math Execution            | MathControl |       | Short, Word, Byte | Write Only |
| Reset Alarms                      | AlarmReset  |       | Boolean           | Write Only |
| Control Command and Response      | Command     |       | String            | Read/Write |
| Direct Reloading of Configuration | Reset       |       | Boolean           | Write Only |
| SetTime*                          | Tag         |       | Boolean           | Write Only |

\* The SetTime tag will cause the device time to be updated. Writing 0 or 1 to the tag will update the Device Date and Time which can be verified from the Date tag and the Time tag. The SetTime tag will always display 0 because it is a Write Only tag. After a successful update, the following message will be posted: "Device Clock set to system time [Device <device\_name>]."

● **Note:** The SetTime tag requires that the device's Port property be set to Ethernet Exclusive Port. For more information, refer to [Communications Properties](#).

### Math Communication Data

The CD address type is only valid for devices equipped with the math option and write operations to CD addresses for non-math equipped devices will return an error. The range of valid data for addresses 'CDxx' is -32000 to 32000. CD data values are converted in the device from a whole number to a real number based on the decimal point placement specified for the associated math channel. For example, the device would interpret a data value of 32000 written to a CD address location with a decimal point placement of .000 as 32.000

### Model Name of Device

The Model address type that returns the model name of the device may return 'DR231' for models of type DR241 and 'DR232' for models of type DR242.

### Control Math Execution

The MathControl address type is only available for devices equipped with the math option and write operations to the MathControl tag for non-math equipped devices will return an error.

### Control Command and Response

The Command address allows the user to send a string command and receive a string response to and from the device. This allows the user to send any command to the device, including commands not directly supported by the driver.

● **Caution:** Write operations using the Command address should be performed with care.

#### ● Notes:

1. The actual number of addresses available for of each type depends on the configuration of the Yokogawa device. If at runtime the driver finds that an address is not present in the device, the driver will post an error message and remove the tag from its scan list.
2. 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 and null string for string data types.

## DR232 Addressing

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

### Measured Channels

| Address Type                   | Format            | Range   | Data Types                | Access    |
|--------------------------------|-------------------|---------|---------------------------|-----------|
| Process Value of Channel       | CHxxx or CHxxx.PV | 001-560 | <b>Double</b> , Float     | Read Only |
| Alarm Summary of Channel       | CHxxx.Alarm       | 001-560 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level1 Status of Channel | CHxxx.Alarm 1     | 001-560 | <b>Short</b> , Word, Byte | Read Only |

| Address Type  | Format                  | Range   | Data Types                | Access     |
|---|-------------------------|---------|---------------------------|------------|
| Alarm Level2 Status of Channel                          | CHxxx.Alarm2            | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level3 Status of Channel                          | CHxxx.Alarm3            | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level4 Status of Channel                          | CHxxx.Alarm4            | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level1 of Channel*<br>(numeric, ex. 0 = Off) | CHxxx.AlarmType1.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level2 of Channel*<br>(numeric)              | CHxxx.AlarmType2.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level3 of Channel*<br>(numeric)              | CHxxx.AlarmType3.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level4 of Channel*<br>(numeric)              | CHxxx.AlarmType4.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level1 of Channel*<br>(string, ex. "OFF")    | CHxxx.AlarmType1.String | 001-560 | <b>String</b>             | Read Only  |
| Alarm Type Level2 of Channel*<br>(string)               | CHxxx.AlarmType2.String | 001-560 | <b>String</b>             | Read Only  |
| Alarm Type Level3 of Channel*<br>(string)               | CHxxx.AlarmType3.String | 001-560 | <b>String</b>             | Read Only  |
| Alarm Type Level4 of Channel*<br>(string)               | CHxxx.AlarmType4.String | 001-560 | <b>String</b>             | Read Only  |
| Set and Read Level1 Alarm Set-point                     | CHxxx.ASP1              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Set-point                     | CHxxx.ASP2              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Set-point                     | CHxxx.ASP3              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Set-point                     | CHxxx.ASP4              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Channel*                           | CHxxx.scale_Hi          | 001-560 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Channel*                           | CHxxx.scale_Lo          | 001-560 | <b>Double</b> , Float     | Read Only  |
| Unit String of Channel*                                 | CHxxx.unit              | 001-560 | <b>String</b>             | Read Only  |
| Tagname of Channel*                                     | CHxxx.tag               | 001-    | <b>String</b>             | Read Only  |



| Address Type               | Format       | Range   | Data Types                | Access    |
|----------------------------|--------------|---------|---------------------------|-----------|
|                            |              | 560     |                           |           |
| Status of Channel*         | CHxxx.status | 001-560 | <b>String</b>             | Read Only |
| Lowest Measuring Channel*  | CH.Low       |         | <b>Short</b> , Word, Byte | Read Only |
| Highest Measuring Channel* | CH.High      |         | <b>Short</b> , Word, Byte | Read Only |

## Math Channels

| Address Type                        | Format            | Range | Data Types                | Access     |
|-------------------------------------|-------------------|-------|---------------------------|------------|
| Process Value of Math Channel       | CHAxX or CHAxX.PV | 01-60 | <b>Double</b> , Float     | Read Only  |
| Alarm Summary of Math Channel       | CHAxX.Alarm       | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level1 Status of Math Channel | CHAxX.Alarm1      | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level2 Status of Math Channel | CHAxX.Alarm2      | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level3 Status of Math Channel | CHAxX.Alarm3      | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level4 Status of Math Channel | CHAxX.Alarm4      | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Set and Read Level1 Alarm Setpoint  | CHAxX.ASP1        | 01-60 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Setpoint  | CHAxX.ASP2        | 01-60 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Setpoint  | CHAxX.ASP3        | 01-60 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Setpoint  | CHAxX.ASP4        | 01-60 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Math Channel*  | CHAxX.scale_Hi    | 01-60 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Math Channel*  | CHAxX.scale_Lo    | 01-60 | <b>Double</b> , Float     | Read Only  |
| Unit String of Math Channel*        | CHAxX.unit        | 01-60 | <b>String</b>             | Read Only  |
| Tagname of Math Channel*            | CHAxX.tag         | 01-60 | <b>String</b>             | Read Only  |
| Status of Math Channel*             | CHAxX.status      | 01-60 | <b>String</b>             | Read Only  |
| Lowest Math Channel*                | CHA.Low           |       | <b>Short</b> , Word, Byte | Read Only  |
| Highest Math Channel*               | CHA.High          |       | <b>Short</b> , Word, Byte | Read Only  |

\* Data associated with these addresses are only read at the device 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.

## Alarm Setpoints

Data values for Alarm Setpoints that are undefined in the device will be returned as +INF. Data values can only be written to Alarm Setpoints that are defined in the device. Write operations to undefined Alarm Setpoints will return an error.

## Scales

Data values for Scale\_Hi and Scale\_Lo for channels that are skipped will be returned as +INF.

## Tag Names

For devices that do not support tag names and 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 'CH001' will be returned as 'CH001'.

### 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  |
| Model Name of Device*             | Model       |       | String            | Read Only  |
| Math Communication Data*          | CDxx        | 01-60 | Short, Word, Byte | Write Only |
| Control Math Execution            | MathControl |       | Short, Word, Byte | Write Only |
| Reset Alarms                      | AlarmReset  |       | Boolean           | Write Only |
| Control Command and Response      | Command     |       | String            | Read/Write |
| Direct Reloading of Configuration | Reset       |       | Boolean           | Write Only |
| SetTime*                          | Tag         |       | Boolean           | Write Only |

\*The SetTime tag will cause the device time to be updated. Writing 0 or 1 to the tag will update the Device Date and Time which can be verified from the Date tag and the Time tag. The SetTime tag will always display 0 because it is a Write Only tag. After a successful update, the following message will be posted: "Device Clock set to system time [Device <device\_name>]."

**Note:** The SetTime tag requires that the device's Port property be set to Ethernet Exclusive Port. For more information, refer to [Communications Properties](#).

### Math Communication Data

The CD address type is only valid for devices equipped with the math option and write operations to CD addresses for non-math equipped devices will return an error. The range of valid data for addresses 'CDxx' is -32000 to 32000. CD data values are converted in the device from a whole number to a real number based on the decimal point placement specified for the associated math channel. For example, the device would interpret a data value of 32000 written to a CD address location with a decimal point placement of .000 as 32.000

### Model Name of Device

The Model address type that returns the model name of the device may return 'DR231' for models of type DR241 and 'DR232' for models of type DR242.

### Control Math Execution

The MathControl address type is only available for devices equipped with the math option and write operations to the MathControl tag for non-math equipped devices will return an error.

### Control Command and Response

The Command address allows the user to send a string command and receive a string response to and from the device. This allows the user to send any command to the device, including commands not directly supported by the driver.

**Caution:** Write operations using the Command address should be performed with care.

**Notes:**

1. The actual number of addresses available for of each type depends on the configuration of the Yokogawa device. If at runtime the driver finds that an address is not present in the device, the driver will post an error message and remove the tag from its scan list.
2. 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 and null string for string data types.

## DR241 Addressing

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

### Measured Channels

| Address Type  | Format                  | Range   | Data Types                | Access    |
|---|-------------------------|---------|---------------------------|-----------|
| Process Value of Channel                                | CHxxx or CHxxx.PV       | 001-040 | <b>Double</b> , Float     | Read Only |
| Alarm Summary of Channel                                | CHxxx.Alarm             | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level1 Status of Channel                          | CHxxx.Alarm1            | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level2 Status of Channel                          | CHxxx.Alarm2            | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level3 Status of Channel                          | CHxxx.Alarm3            | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level4 Status of Channel                          | CHxxx.Alarm4            | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level1 of Channel*<br>(numeric, ex. 0 = Off) | CHxxx.AlarmType1.Num    | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level2 of Channel*<br>(numeric)              | CHxxx.AlarmType2.Num    | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level3 of Channel*<br>(numeric)              | CHxxx.AlarmType3.Num    | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level4 of Channel*<br>(numeric)              | CHxxx.AlarmType4.Num    | 001-040 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level1 of Channel*<br>(string, ex. "OFF")    | CHxxx.AlarmType1.String | 001-040 | <b>String</b>             | Read Only |
| Alarm Type Level2 of Channel*<br>(string)               | CHxxx.AlarmType2.String | 001-040 | <b>String</b>             | Read Only |
| Alarm Type Level3 of Channel*<br>(string)               | CHxxx.AlarmType3.String | 001-040 | <b>String</b>             | Read Only |

| Address Type                              | Format                  | Range   | Data Types                | Access     |
|---|-------------------------|---------|---------------------------|------------|
| (string)                                  |                         |         |                           |            |
| Alarm Type Level4 of Channel*<br>(string) | CHxxx.AlarmType4.String | 001-040 | <b>String</b>             | Read Only  |
| Set and Read Level1 Alarm Setpoint        | CHxxx.ASP1              | 001-040 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Setpoint        | CHxxx.ASP2              | 001-040 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Setpoint        | CHxxx.ASP3              | 001-040 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Setpoint        | CHxxx.ASP4              | 001-040 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Channel*             | CHxxx.scale_Hi          | 001-040 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Channel*             | CHxxx.scale_Lo          | 001-040 | <b>Double</b> , Float     | Read Only  |
| Unit String of Channel*                   | CHxxx.unit              | 001-040 | <b>String</b>             | Read Only  |
| Tagname of Channel*                       | CHxxx.tag               | 001-040 | <b>String</b>             | Read Only  |
| Status of Channel*                        | CHxxx.status            | 001-040 | <b>String</b>             | Read Only  |
| Lowest Measuring Channel*                 | CH.Low                  |         | <b>Short</b> , Word, Byte | Read Only  |
| Highest Measuring Channel*                | CH.High                 |         | <b>Short</b> , Word, Byte | Read Only  |

### Math Channels

| Address Type                        | Format          | Range | Data Types                | Access     |
|-------------------------------------|-----------------|-------|---------------------------|------------|
| Process Value of Math Channel       | CHAx or CHAx.PV | 01-30 | <b>Double</b> , Float     | Read Only  |
| Alarm Summary of Math Channel       | CHAx.Alarm      | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level1 Status of Math Channel | CHAx.Alarm1     | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level2 Status of Math Channel | CHAx.Alarm2     | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level3 Status of Math Channel | CHAx.Alarm3     | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level4 Status of Math Channel | CHAx.Alarm4     | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Set and Read Level1 Alarm Setpoint  | CHAx.ASP1       | 01-30 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Setpoint  | CHAx.ASP2       | 01-30 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Setpoint  | CHAx.ASP3       | 01-30 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Setpoint  | CHAx.ASP4       | 01-30 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Math Channel*  | CHAx.scale_Hi   | 01-30 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Math Channel*  | CHAx.scale_Lo   | 01-30 | <b>Double</b> , Float     | Read Only  |
| Unit String of Math Channel*        | CHAx.unit       | 01-30 | <b>String</b>             | Read Only  |
| Tagname of Math Channel*            | CHAx.tag        | 01-30 | <b>String</b>             | Read Only  |

| Address Type            | Format      | Range | Data Types        | Access    |
|-------------------------|-------------|-------|-------------------|-----------|
| Status of Math Channel* | CHAx.status | 01-30 | String            | Read Only |
| Lowest Math Channel*    | CHA.Low     |       | Short, Word, Byte | Read Only |
| Highest Math Channel*   | CHA.High    |       | Short, Word, Byte | Read Only |

\* Data associated with these addresses are only read at the device 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.

### Alarm Setpoints

Data values for Alarm Setpoints that are undefined in the device will be returned as +INF. Data values can only be written to Alarm Setpoints that are defined in the device. Write operations to undefined Alarm Setpoints will return an error.

### Scales

Data values for Scale\_Hi and Scale\_Lo for channels that are skipped will be returned as +INF.

### Tag Names

For devices that do not support tag names and 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 'CH001' will be returned as 'CH001'.

### 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  |
| Model Name of Device*             | Model       |       | String            | Read Only  |
| Math Communication Data*          | CDxx        | 01-30 | Short, Word, Byte | Write Only |
| Control Math Execution            | MathControl |       | Short, Word, Byte | Write Only |
| Reset Alarms                      | AlarmReset  |       | Boolean           | Write Only |
| Control Command and Response      | Command     |       | String            | Read/Write |
| Direct Reloading of Configuration | Reset       |       | Boolean           | Write Only |
| SetTime*                          | Tag         |       | Boolean           | Write Only |

\* The SetTime tag will cause the device time to be updated. Writing 0 or 1 to the tag will update the Device Date and Time which can be verified from the Date tag and the Time tag. The SetTime tag will always display 0 because it is a Write Only tag. After a successful update, the following message will be posted: "Device Clock set to system time [Device <device\_name>]."

**Note:** The SetTime tag requires that the device's Port property be set to Ethernet Exclusive Port. For more information, refer to [Communications Properties](#).

### Math Communication Data

The CD address type is only valid for devices equipped with the math option and write operations to CD addresses for non-math equipped devices will return an error. The range of valid data for addresses 'CDxx' is -32000 to 32000. CD data values are converted in the device from a whole number to a real number

based on the decimal point placement specified for the associated math channel. For example, the device would interpret a data value of 32000 written to a CD address location with a decimal point placement of .000 as 32.000

### Model Name of Device


The Model address type that returns the model name of the device may return 'DR231' for models of type DR241 and 'DR232' for models of type DR242.

### Control Math Execution

The MathControl address type is only available for devices equipped with the math option and write operations to the MathControl tag for non-math equipped devices will return an error.

### Control Command and Response

The Command address allows the user to send a string command and receive a string response to and from the device. This allows the user to send any command to the device, including commands not directly supported by the driver.

 **Caution:** Write operations using the Command address should be performed with care.

#### Notes:

1. The actual number of addresses available for of each type depends on the configuration of the Yokogawa device. If at runtime the driver finds that an address is not present in the device, the driver will post an error message and remove the tag from its scan list.
2. 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 and null string for string data types.

## DR242 Addressing

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

### Measured Channels

| Address Type                   | Format               | Range   | Data Types                | Access    |
|--------------------------------|----------------------|---------|---------------------------|-----------|
| Process Value of Channel       | CHxxx or CHxxx.PV    | 001-560 | <b>Double</b> , Float     | Read Only |
| Alarm Summary of Channel       | CHxxx.Alarm          | 001-560 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level1 Status of Channel | CHxxx.Alarm1         | 001-560 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level2 Status of Channel | CHxxx.Alarm2         | 001-560 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level3 Status of Channel | CHxxx.Alarm3         | 001-560 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level4 Status of Channel | CHxxx.Alarm4         | 001-560 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level1 of Channel*  | CHxxx.AlarmType1.Num | 001-    | <b>Short</b> , Word,      | Read Only |

| Address Type   | Format                  | Range   | Data Types                | Access     |
|--|-------------------------|---------|---------------------------|------------|
| (numeric, ex. 0 = Off)                               |                         | 560     | Byte                      |            |
| Alarm Type Level2 of Channel*<br>(numeric)           | CHxxx.AlarmType2.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level3 of Channel*<br>(numeric)           | CHxxx.AlarmType3.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level4 of Channel*<br>(numeric)           | CHxxx.AlarmType4.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level1 of Channel*<br>(string, ex. "OFF") | CHxxx.AlarmType1.String | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level2 of Channel*<br>(string)            | CHxxx.AlarmType2.String | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level3 of Channel*<br>(string)            | CHxxx.AlarmType3.String | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level4 of Channel*<br>(string)            | CHxxx.AlarmType4.String | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Set and Read Level1 Alarm Set-point                  | CHxxx.ASP1              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Set-point                  | CHxxx.ASP2              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Set-point                  | CHxxx.ASP3              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Set-point                  | CHxxx.ASP4              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Channel*                        | CHxxx.scale_Hi          | 001-560 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Channel*                        | CHxxx.scale_Lo          | 001-560 | <b>Double</b> , Float     | Read Only  |
| Unit String of Channel*                              | CHxxx.unit              | 001-560 | <b>String</b>             | Read Only  |
| Tagname of Channel*                                  | CHxxx.tag               | 001-560 | <b>String</b>             | Read Only  |
| Status of Channel*                                   | CHxxx.status            | 001-560 | <b>String</b>             | Read Only  |
| Lowest Measuring Channel*                            | CH.Low                  |         | <b>Short</b> , Word, Byte | Read Only  |
| Highest Measuring Channel*                           | CH.High                 |         | <b>Short</b> , Word, Byte | Read Only  |

## Math Channels

| Address Type                        | Format          | Range | Data Types                | Access     |
|-------------------------------------|-----------------|-------|---------------------------|------------|
| Process Value of Math Channel       | CHAx or CHAx.PV | 01-60 | <b>Double</b> , Float     | Read Only  |
| Alarm Summary of Math Channel       | CHAx.Alarm      | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level1 Status of Math Channel | CHAx.Alarm1     | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level2 Status of Math Channel | CHAx.Alarm2     | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level3 Status of Math Channel | CHAx.Alarm3     | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level4 Status of Math Channel | CHAx.Alarm4     | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Set and Read Level1 Alarm Setpoint  | CHAx.ASP1       | 01-60 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Setpoint  | CHAx.ASP2       | 01-60 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Setpoint  | CHAx.ASP3       | 01-60 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Setpoint  | CHAx.ASP4       | 01-60 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Math Channel*  | CHAx.scale_Hi   | 01-60 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Math Channel*  | CHAx.scale_Lo   | 01-60 | <b>Double</b> , Float     | Read Only  |
| Unit String of Math Channel*        | CHAx.unit       | 01-60 | <b>String</b>             | Read Only  |
| Tagname of Math Channel*            | CHAx.tag        | 01-60 | <b>String</b>             | Read Only  |
| Status of Math Channel*             | CHAx.status     | 01-60 | <b>String</b>             | Read Only  |
| Lowest Math Channel*                | CHA.Low         |       | <b>Short</b> , Word, Byte | Read Only  |
| Highest Math Channel*               | CHA.High        |       | <b>Short</b> , Word, Byte | Read Only  |

\*Data associated with these addresses are only read at the device 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.

## Alarm Setpoints

Data values for Alarm Setpoints that are undefined in the device will be returned as +INF. Data values can only be written to Alarm Setpoints that are defined in the device. Write operations to undefined Alarm Setpoints will return an error.

## Scales

Data values for Scale\_Hi and Scale\_Lo for channels that are skipped will be returned as +INF.

## Tag Names

For devices that do not support tag names and 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 'CH001' will be returned as 'CH001'.

## General Device Data

| Address Type          | Format | Range | Data Types    | Access    |
|-----------------------|--------|-------|---------------|-----------|
| Date of Last Data     | Date   |       | <b>String</b> | Read Only |
| Time of Last Data     | Time   |       | <b>String</b> | Read Only |
| Model Name of Device* | Model  |       | <b>String</b> | Read Only |



| Address Type                      | Format      | Range | Data Types        | Access     |
|-----------------------------------|-------------|-------|-------------------|------------|
| Math Communication Data*          | CDxx        | 01-60 | Short, Word, Byte | Write Only |
| Control Math Execution            | MathControl |       | Short, Word, Byte | Write Only |
| Reset Alarms                      | AlarmReset  |       | Boolean           | Write Only |
| Control Command and Response      | Command     |       | String            | Read/Write |
| Direct Reloading of Configuration | Reset       |       | Boolean           | Write Only |
| SetTime*                          | Tag         |       | Boolean           | Write Only |

\* The SetTime tag will cause the device time to be updated. Writing 0 or 1 to the tag will update the Device Date and Time which can be verified from the Date tag and the Time tag. The SetTime tag will always display 0, because it is a Write Only tag. After a successful update, the following message will be posted: "Device Clock set to system time [Device <device\_name>]."

● **Note:** The SetTime tag requires that the device's Port property be set to Ethernet Exclusive Port. For more information, refer to [Communications Properties](#).

### Math Communication Data

The CD address type is only valid for devices equipped with the math option and write operations to CD addresses for non-math equipped devices will return an error. The range of valid data for addresses 'CDxx' is -32000 to 32000. CD data values are converted in the device from a whole number to a real number based on the decimal point placement specified for the associated math channel. For example, the device would interpret a data value of 32000 written to a CD address location with a decimal point placement of .000 as 32.000

### Model Name of Device

The Model address type that returns the model name of the device may return 'DR231' for models of type DR241 and 'DR232' for models of type DR242.

### Control Math Execution

The MathControl address type is only available for devices equipped with the math option and write operations to the MathControl tag for non-math equipped devices will return an error.

### Control Command and Response

The Command address allows the user to send a string command and receive a string response to and from the device. This allows the user to send any command to the device, including commands not directly supported by the driver.

● **Caution:** Write operations using the Command address should be performed with care.

● **Note:**

1. The actual number of addresses available for of each type depends on the configuration of the Yokogawa device. If at runtime the driver finds that an address is not present in the device, the driver will post an error message and remove the tag from its scan list.
2. 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 and null string for string data types.

## DR130 Addressing

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

### Measured Channels

| Address Type  | Format                  | Range   | Data Types                | Access     |
|---|-------------------------|---------|---------------------------|------------|
| Process Value of Channel                                | CHxxx or CHxxx.PV       | 001-020 | <b>Double</b> , Float     | Read Only  |
| Alarm Summary of Channel                                | CHxxx.Alarm             | 001-020 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level1 Status of Channel                          | CHxxx.Alarm1            | 001-020 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level2 Status of Channel                          | CHxxx.Alarm2            | 001-020 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level3 Status of Channel                          | CHxxx.Alarm3            | 001-020 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level4 Status of Channel                          | CHxxx.Alarm4            | 001-020 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level1 of Channel*<br>(numeric, ex. 0 = Off) | CHxxx.AlarmType1.Num    | 001-020 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level2 of Channel*<br>(numeric)              | CHxxx.AlarmType2.Num    | 001-020 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level3 of Channel*<br>(numeric)              | CHxxx.AlarmType3.Num    | 001-020 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level4 of Channel*<br>(numeric)              | CHxxx.AlarmType4.Num    | 001-020 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level1 of Channel*<br>(string, ex. "OFF")    | CHxxx.AlarmType1.String | 001-020 | <b>String</b>             | Read Only  |
| Alarm Type Level2 of Channel*<br>(string)               | CHxxx.AlarmType2.String | 001-020 | <b>String</b>             | Read Only  |
| Alarm Type Level3 of Channel*<br>(string)               | CHxxx.AlarmType3.String | 001-020 | <b>String</b>             | Read Only  |
| Alarm Type Level4 of Channel*<br>(string)               | CHxxx.AlarmType4.String | 001-020 | <b>String</b>             | Read Only  |
| Set and Read Level1 Alarm Set-point                     | CHxxx.ASP1              | 001-020 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Set-point                     | CHxxx.ASP2              | 001-020 | <b>Double</b> , Float     | Read/Write |

| Address Type                        | Format         | Range   | Data Types                | Access     |
|-------------------------------------|----------------|---------|---------------------------|------------|
| Set and Read Level3 Alarm Set-point | CHxxx.ASP3     | 001-020 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Set-point | CHxxx.ASP4     | 001-020 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Channel*       | CHxxx.scale_Hi | 001-020 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Channel*       | CHxxx.scale_Lo | 001-020 | <b>Double</b> , Float     | Read Only  |
| Unit String of Channel*             | CHxxx.unit     | 001-020 | <b>String</b>             | Read Only  |
| Tagname of Channel*                 | CHxxx.tag      | 001-020 | <b>String</b>             | Read Only  |
| Status of Channel*                  | CHxxx.status   | 001-020 | <b>String</b>             | Read Only  |
| Lowest Measuring Channel*           | CH.Low         |         | <b>Short</b> , Word, Byte | Read Only  |
| Highest Measuring Channel*          | CH.High        |         | <b>Short</b> , Word, Byte | Read Only  |

### Math Channels

| Address Type                        | Format            | Range | Data Types                | Access     |
|-------------------------------------|-------------------|-------|---------------------------|------------|
| Process Value of Math Channel       | CHAxX or CHAxX.PV | 01-30 | <b>Double</b> , Float     | Read Only  |
| Alarm Summary of Math Channel       | CHAxX.Alarm       | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level1 Status of Math Channel | CHAxX.Alarm1      | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level2 Status of Math Channel | CHAxX.Alarm2      | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level3 Status of Math Channel | CHAxX.Alarm3      | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level4 Status of Math Channel | CHAxX.Alarm4      | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Set and Read Level1 Alarm Setpoint  | CHAxX.ASP1        | 01-30 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Setpoint  | CHAxX.ASP2        | 01-30 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Setpoint  | CHAxX.ASP3        | 01-30 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Setpoint  | CHAxX.ASP4        | 01-30 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Math Channel*  | CHAxX.scale_Hi    | 01-30 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Math Channel*  | CHAxX.scale_Lo    | 01-30 | <b>Double</b> , Float     | Read Only  |
| Unit String of Math Channel*        | CHAxX.unit        | 01-30 | <b>String</b>             | Read Only  |
| Tagname of Math Channel*            | CHAxX.tag         | 01-30 | <b>String</b>             | Read Only  |
| Status of Math Channel*             | CHAxX.status      | 01-30 | <b>String</b>             | Read Only  |
| Lowest Math Channel*                | CHA.Low           |       | <b>Short</b> , Word, Byte | Read Only  |
| Highest Math Channel*               | CHA.High          |       | <b>Short</b> , Word, Byte | Read Only  |

\* Data associated with these addresses are only read at the device 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.

### Alarm Setpoints

Data values for Alarm Setpoints that are undefined in the device will be returned as +INF. Data values can only be written to Alarm Setpoints that are defined in the device. Write operations to undefined Alarm Setpoints will return an error.

### Scales

Data values for Scale\_Hi and Scale\_Lo for channels that are skipped will be returned as +INF.

### Tag Names

For devices that do not support tag names and 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 'CH001' will be returned as 'CH001'.

### 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  |
| Model Name of Device*             | Model       |       | String            | Read Only  |
| Math Communication Data*          | CDxx        | 01-30 | Short, Word, Byte | Write Only |
| Control Math Execution            | MathControl |       | Short, Word, Byte | Write Only |
| Reset Alarms                      | AlarmReset  |       | Boolean           | Write Only |
| Control Command and Response      | Command     |       | String            | Read/Write |
| Direct Reloading of Configuration | Reset       |       | Boolean           | Write Only |
| SetTime*                          | Tag         |       | Boolean           | Write Only |

\* The SetTime tag will cause the device time to be updated. Writing 0 or 1 to the tag will update the Device Date and Time which can be verified from the Date tag and the Time tag. The SetTime tag will always display 0 because it is a Write Only tag. After a successful update, the following message will be posted: "Device Clock set to system time [Device <device\_name>]."

● **Note:** The SetTime tag requires that the device's Port property be set to Ethernet Exclusive Port. For more information, refer to [Communications Properties](#).

### Math Communication Data

The CD address type is only valid for devices equipped with the math option and write operations to CD addresses for non-math equipped devices will return an error. The range of valid data for addresses 'CDxx' is -32000 to 32000. CD data values are converted in the device from a whole number to a real number based on the decimal point placement specified for the associated math channel. For example, the device would interpret a data value of 32000 written to a CD address location with a decimal point placement of .000 as 32.000

### Model Name of Device

The Model address type that returns the model name of the device may return 'DR231' for models of type DR241 and 'DR232' for models of type DR242.

## Control Math Execution

The MathControl address type is only available for devices equipped with the math option and write operations to the MathControl tag for non-math equipped devices will return an error.

## Control Command and Response

The Command address allows the user to send a string command and receive a string response to and from the device. This allows the user to send any command to the device, including commands not directly supported by the driver.

 **Caution:** Write operations using the Command address should be performed with care.

### Notes:

1. The actual number of addresses available for of each type depends on the configuration of the Yokogawa device. If at runtime the driver finds that an address is not present in the device, the driver will post an error message and remove the tag from its scan list.
2. 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 and null string for string data types.

## DC100-1 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                                | CHxxx or CHxxx.PV    | 001-060 | <b>Double</b> , Float     | Read Only |
| Alarm Summary of Channel                                | CHxxx.Alarm          | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level1 Status of Channel                          | CHxxx.Alarm1         | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level2 Status of Channel                          | CHxxx.Alarm2         | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level3 Status of Channel                          | CHxxx.Alarm3         | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level4 Status of Channel                          | CHxxx.Alarm4         | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level1 of Channel*<br>(numeric, ex. 0 = Off) | CHxxx.AlarmType1.Num | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level2 of Channel*<br>(numeric)              | CHxxx.AlarmType2.Num | 001-060 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Type Level3 of Channel*                           | CHxxx.AlarmType3.Num | 001-060 | <b>Short</b> , Word, Byte | Read Only |

| Address Type   | Format                  | Range   | Data Types                | Access     |
|--|-------------------------|---------|---------------------------|------------|
| (numeric)  |                         |         |                           |            |
| Alarm Type Level4 of Channel*<br>(numeric)           | CHxxx.AlarmType4.Num    | 001-060 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level1 of Channel*<br>(string, ex. "OFF") | CHxxx.AlarmType1.String | 001-060 | <b>String</b>             | Read Only  |
| Alarm Type Level2 of Channel*<br>(string)            | CHxxx.AlarmType2.String | 001-060 | <b>String</b>             | Read Only  |
| Alarm Type Level3 of Channel*<br>(string)            | CHxxx.AlarmType3.String | 001-060 | <b>String</b>             | Read Only  |
| Alarm Type Level4 of Channel*<br>(string)            | CHxxx.AlarmType4.String | 001-060 | <b>String</b>             | Read Only  |
| Set and Read Level1 Alarm Set-point                  | CHxxx.ASP1              | 001-060 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Set-point                  | CHxxx.ASP2              | 001-060 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Set-point                  | CHxxx.ASP3              | 001-060 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Set-point                  | CHxxx.ASP4              | 001-060 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Channel*                        | CHxxx.scale_Hi          | 001-060 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Channel*                        | CHxxx.scale_Lo          | 001-060 | <b>Double</b> , Float     | Read Only  |
| Unit String of Channel*                              | CHxxx.unit              | 001-060 | <b>String</b>             | Read Only  |
| Tagname of Channel*                                  | CHxxx.tag               | 001-060 | <b>String</b>             | Read Only  |
| Status of Channel*                                   | CHxxx.status            | 001-060 | <b>String</b>             | Read Only  |
| Lowest Measuring Channel*                            | CH.Low                  |         | <b>Short</b> , Word, Byte | Read Only  |
| Highest Measuring Channel*                           | CH.High                 |         | <b>Short</b> , Word, Byte | Read Only  |

### Math Channels

| Address Type                        | Format          | Range | Data Types                | Access    |
|-------------------------------------|-----------------|-------|---------------------------|-----------|
| Process Value of Math Channel       | CHAx or CHAx.PV | 01-30 | <b>Double</b> , Float     | Read Only |
| Alarm Summary of Math Channel       | CHAx.Alarm      | 01-30 | <b>Short</b> , Word, Byte | Read Only |
| Alarm Level1 Status of Math Channel | CHAx.Alarm 1    | 01-30 | <b>Short</b> , Word, Byte | Read Only |

| Address Type                        | Format          | Range | Data Types                | Access     |
|-------------------------------------|-----------------|-------|---------------------------|------------|
| Alarm Level2 Status of Math Channel | CHAx.x.Alarm2   | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level3 Status of Math Channel | CHAx.x.Alarm3   | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level4 Status of Math Channel | CHAx.x.Alarm4   | 01-30 | <b>Short</b> , Word, Byte | Read Only  |
| Set and Read Level1 Alarm Setpoint  | CHAx.x.ASP1     | 01-30 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Setpoint  | CHAx.x.ASP2     | 01-30 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Setpoint  | CHAx.x.ASP3     | 01-30 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Setpoint  | CHAx.x.ASP4     | 01-30 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Math Channel*  | CHAx.x.scale_Hi | 01-30 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Math Channel*  | CHAx.x.scale_Lo | 01-30 | <b>Double</b> , Float     | Read Only  |
| Unit String of Math Channel*        | CHAx.x.unit     | 01-30 | <b>String</b>             | Read Only  |
| Tagname of Math Channel*            | CHAx.x.tag      | 01-30 | <b>String</b>             | Read Only  |
| Status of Math Channel*             | CHAx.x.status   | 01-30 | <b>String</b>             | Read Only  |
| Lowest Math Channel*                | CHA.Low         |       | <b>Short</b> , Word, Byte | Read Only  |
| Highest Math Channel*               | CHA.High        |       | <b>Short</b> , Word, Byte | Read Only  |

\* Data associated with these addresses are only read at the device 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.

### Alarm Setpoints

Data values for Alarm Setpoints that are undefined in the device will be returned as +INF. Data values can only be written to Alarm Setpoints that are defined in the device. Write operations to undefined Alarm Setpoints will return an error.

### Scales

Data values for Scale\_Hi and Scale\_Lo for channels that are skipped will be returned as +INF.

### Tag Names

For devices that do not support tag names and 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 'CH001' will be returned as 'CH001'.

### General Device Data

| Address Type                 | Format      | Range | Data Types                | Access     |
|------------------------------|-------------|-------|---------------------------|------------|
| Date of Last Data            | Date        |       | <b>String</b>             | Read Only  |
| Time of Last Data            | Time        |       | <b>String</b>             | Read Only  |
| Model Name of Device*        | Model       |       | <b>String</b>             | Read Only  |
| Math Communication Data*     | CDxx        | 01-30 | <b>Short</b> , Word, Byte | Write Only |
| Control Math Execution       | MathControl |       | <b>Short</b> , Word, Byte | Write Only |
| Reset Alarms                 | AlarmReset  |       | <b>Boolean</b>            | Write Only |
| Control Command and Response | Command     |       | <b>String</b>             | Read/Write |

| Address Type                      | Format | Range | Data Types     | Access     |
|-----------------------------------|--------|-------|----------------|------------|
| Direct Reloading of Configuration | Reset  |       | <b>Boolean</b> | Write Only |
| SetTime*                          | Tag    |       | <b>Boolean</b> | Write Only |

\* The SetTime tag will cause the device time to be updated. Writing 0 or 1 to the tag will update the Device Date and Time which can be verified from the Date tag and the Time tag. The SetTime tag will always display 0 because it is a Write Only tag. After a successful update, the following message will be posted: "Device Clock set to system time [Device <device\_name>]."

● **Note:** The SetTime tag requires that the device's Port property be set to Ethernet Exclusive Port. For more information, refer to [Communications Properties](#).

### Math Communication Data

The CD address type is only valid for devices equipped with the math option and write operations to CD addresses for non-math equipped devices will return an error. The range of valid data for addresses 'CDxx' is -32000 to 32000. CD data values are converted in the device from a whole number to a real number based on the decimal point placement specified for the associated math channel. For example, a data value of 32000 written to a CD address location with a decimal point placement of .000 would be interpreted by the device as 32.000

### Model Name of Device

The Model address type that returns the model name of the device may return 'DR231' for models of type DR241 and 'DR232' for models of type DR242.

### Control Math Execution

The MathControl address type is only available for devices equipped with the math option and write operations to the MathControl tag for non-math equipped devices will return an error.

### Control Command and Response

The Command address allows the user to send a string command and receive a string response to and from the device. This allows the user to send any command to the device, including commands not directly supported by the driver.

● **Caution:** Write operations using the Command address should be performed with care.

#### ● Notes:

1. The actual number of addresses available for of each type depends on the configuration of the Yokogawa device. If at runtime the driver finds that an address is not present in the device, the driver will post an error message and remove the tag from its scan list.
2. 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 and null string for string data types.

### DC100-2 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                                | CHxxx or CHxxx.PV       | 001-560 | <b>Double</b> , Float     | Read Only  |
| Alarm Summary of Channel                                | CHxxx.Alarm             | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level1 Status of Channel                          | CHxxx.Alarm 1           | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level2 Status of Channel                          | CHxxx.Alarm2            | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level3 Status of Channel                          | CHxxx.Alarm 3           | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level4 Status of Channel                          | CHxxx.Alarm 4           | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level1 of Channel*<br>(numeric, ex. 0 = Off) | CHxxx.AlarmType1.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level2 of Channel*<br>(numeric)              | CHxxx.AlarmType2.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level3 of Channel*<br>(numeric)              | CHxxx.AlarmType3.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level4 of Channel*<br>(numeric)              | CHxxx.AlarmType4.Num    | 001-560 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Type Level1 of Channel*<br>(string, ex. "OFF")    | CHxxx.AlarmType1.String | 001-560 | <b>String</b>             | Read Only  |
| Alarm Type Level2 of Channel*<br>(string)               | CHxxx.AlarmType2.String | 001-560 | <b>String</b>             | Read Only  |
| Alarm Type Level3 of Channel*<br>(string)               | CHxxx.AlarmType3.String | 001-560 | <b>String</b>             | Read Only  |
| Alarm Type Level4 of Channel*<br>(string)               | CHxxx.AlarmType4.String | 001-560 | <b>String</b>             | Read Only  |
| Set and Read Level1 Alarm Set-point                     | CHxxx.ASP1              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Set-point                     | CHxxx.ASP2              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Set-point                     | CHxxx.ASP3              | 001-560 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Set-point                     | CHxxx.ASP4              | 001-560 | <b>Double</b> , Float     | Read/Write |

| Address Type                  | Format         | Range   | Data Types                | Access    |
|-------------------------------|----------------|---------|---------------------------|-----------|
| Upper Scale Value of Channel* | CHxxx.scale_Hi | 001-560 | <b>Double</b> , Float     | Read Only |
| Lower Scale Value of Channel* | CHxxx.scale_Lo | 001-560 | <b>Double</b> , Float     | Read Only |
| Unit String of Channel*       | CHxxx.unit     | 001-560 | <b>String</b>             | Read Only |
| Tagname of Channel*           | CHxxx.tag      | 001-560 | <b>String</b>             | Read Only |
| Status of Channel*            | CHxxx.status   | 001-560 | <b>String</b>             | Read Only |
| Lowest Measuring Channel*     | CH.Low         |         | <b>Short</b> , Word, Byte | Read Only |
| Highest Measuring Channel*    | CH.High        |         | <b>Short</b> , Word, Byte | Read Only |

### Math Channels

| Address Type                        | Format            | Range | Data Types                | Access     |
|-------------------------------------|-------------------|-------|---------------------------|------------|
| Process Value of Math Channel       | CHAxX or CHAxX.PV | 01-60 | <b>Double</b> , Float     | Read Only  |
| Alarm Summary of Math Channel       | CHAxX.Alarm       | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level1 Status of Math Channel | CHAxX.Alarm1      | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level2 Status of Math Channel | CHAxX.Alarm2      | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level3 Status of Math Channel | CHAxX.Alarm3      | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Alarm Level4 Status of Math Channel | CHAxX.Alarm4      | 01-60 | <b>Short</b> , Word, Byte | Read Only  |
| Set and Read Level1 Alarm Setpoint  | CHAxX.ASP1        | 01-60 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level2 Alarm Setpoint  | CHAxX.ASP2        | 01-60 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level3 Alarm Setpoint  | CHAxX.ASP3        | 01-60 | <b>Double</b> , Float     | Read/Write |
| Set and Read Level4 Alarm Setpoint  | CHAxX.ASP4        | 01-60 | <b>Double</b> , Float     | Read/Write |
| Upper Scale Value of Math Channel*  | CHAxX.scale_Hi    | 01-60 | <b>Double</b> , Float     | Read Only  |
| Lower Scale Value of Math Channel*  | CHAxX.scale_Lo    | 01-60 | <b>Double</b> , Float     | Read Only  |
| Unit String of Math Channel*        | CHAxX.unit        | 01-60 | <b>String</b>             | Read Only  |
| Tagname of Math Channel*            | CHAxX.tag         | 01-60 | <b>String</b>             | Read Only  |
| Status of Math Channel*             | CHAxX.status      | 01-60 | <b>String</b>             | Read Only  |
| Lowest Math Channel*                | CHA.Low           |       | <b>Short</b> , Word, Byte | Read Only  |
| Highest Math Channel*               | CHA.High          |       | <b>Short</b> , Word, Byte | Read Only  |

\* Data associated with these addresses are only read at the device 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.

### Alarm Setpoints

Data values for Alarm Setpoints that are undefined in the device will be returned as +INF. Data values can only be written to Alarm Setpoints that are defined in the device. Write operations to undefined Alarm Setpoints will return an error.

## Scales

Data values for Scale\_Hi and Scale\_Lo for channels that are skipped will be returned as +INF.

## Tag Names

For devices that do not support tag names and 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 'CH001' will be returned as 'CH001'.

## 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  |
| Model Name of Device*             | Model       |       | String            | Read Only  |
| Math Communication Data*          | CDxx        | 01-60 | Short, Word, Byte | Write Only |
| Control Math Execution            | MathControl |       | Short, Word, Byte | Write Only |
| Reset Alarms                      | AlarmReset  |       | Boolean           | Write Only |
| Control Command and Response      | Command     |       | String            | Read/Write |
| Direct Reloading of Configuration | Reset       |       | Boolean           | Write Only |
| SetTime*                          | Tag         |       | Boolean           | Write Only |

\* The SetTime tag will cause the device time to be updated. Writing 0 or 1 to the tag will update the Device Date and Time which can be verified from the Date tag and the Time tag. The SetTime tag will always display 0 because it is a Write Only tag. After a successful update, the following message will be posted: "Device Clock set to system time [Device <device\_name>]."

● **Note:** The SetTime tag requires that the device's Port property be set to Ethernet Exclusive Port. For more information, refer to [Communications Properties](#).

## Math Communication Data

The CD address type is only valid for devices equipped with the math option and write operations to CD addresses for non-math equipped devices will return an error. The range of valid data for addresses 'CDxx' is -32000 to 32000. CD data values are converted in the device from a whole number to a real number based on the decimal point placement specified for the associated math channel. For example, the device would interpret a data value of 32000 written to a CD address location with a decimal point placement of .000 as 32.000

## Model Name of Device


The Model address type that returns the model name of the device may return 'DR231' for models of type DR241 and 'DR232' for models of type DR242.

## Control Math Execution

The MathControl address type is only available for devices equipped with the math option and write operations to the MathControl tag for non-math equipped devices will return an error.

## Control Command and Response

The Command address allows the user to send a string command and receive a string response to and from the device. This allows the user to send any command to the device, including commands not directly supported by the driver.

 **Caution:** Write operations using the Command address should be performed with care.

 **Note:**

1. The actual number of addresses available for of each type depends on the configuration of the Yokogawa device. If at runtime the driver finds that an address is not present in the device, the driver will post an error message and remove the tag from its scan list.
2. 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 and null string for string data types.

## Automatic Tag Database Generation

This driver's Automatic OPC Tag Database Generation features have been designed to make configuring 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 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 then 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 driver-specific information. An example of these two conditions is as follows:

1. 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.
2. An Ethernet I/O system that supports detection of I/O module type. The driver 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.

Automatic Tag Database Generation is completely configurable. The following property group is used to define how the OPC Server and the associated communications driver will handle Automatic OPC Tag Database Generation:

| Tag Generation                          |                            |
|---|----------------------------|
| On Device Startup                       | Do Not Generate on Startup |
| On Duplicate Tag                        | Delete on Create           |
| Parent Group                            |                            |
| Allow Automatically Generated Subgroups | Enable                     |
| Create                                  | Create tags                |

The **On Device Startup** property is used to configure when OPC tags will be automatically generated. Descriptions of the properties are as follows:

- **Do not generate on startup:** This option prevents the driver from adding any OPC tags to the OPC Server's tag space. This is the default condition.
- **Always generate on startup:** This option causes the driver to always evaluate the device for tag information. OPC tags will be added to the tag space of the server each time the server is launched.
- **Generate on first startup:** This option causes the driver to evaluate the target device for tag information the first time the OPC Server project is run. OPC tags will be added to the server tag space as needed.

**Note:** 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.

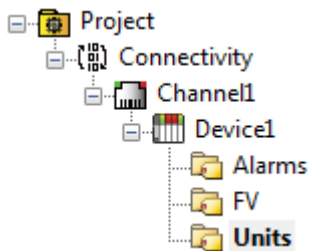
When automatic tag generation is enabled, the server needs to know what to do with the OPC tags that were added from previous runs (or with the OPC tags that have been added or modified after being added by the communications driver originally). The selection **On Duplicate Tag property** controls how 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. In the Ethernet I/O example above, this would occur if users continued to change the I/O modules in the rack while the OPC Server was configured to always generate new OPC tags on startup. Under this condition, tags would be added to the server every time the communications driver detected a new I/O module. If the old tags are not removed, they will accumulate in the server's tag space. Descriptions of the selections are as follows:

- **Delete on create:** This option allows the server to remove any tags that have previously been added to the tag space before any new tags can be added by the communications driver. This is the default setting.
- **Overwrite as necessary:** This option allows the server to only remove tags that the communications driver will replace with new tags. Any tags that are not being overwritten will remain in the server's tag space.
- **Do not overwrite:** This option prevents the server from removing any tags that had previously been generated or that already exist in the server. With this selection, the communications driver can only add tags that are completely new.
- **Do not overwrite, log error:** This option has the same effect as the third option, but also posts an error message to the OPC Server's Event Log when a tag overwrite would have occurred.

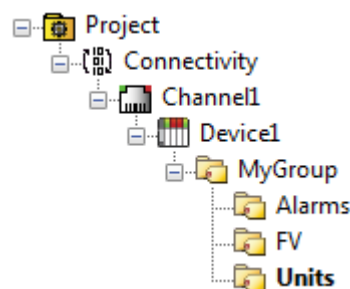
● **Note:** The removal of OPC tags not only affects tags that have been automatically generated by the communications driver, but also any tags that have been added using names that match generated tags. It is recommended that users avoid adding tags to the server using names that match tags that may be automatically generated by the driver.

**Parent Group:** can be used to keep automatically-generated tags from mixing with tags that have been entered manually. This property specifies 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 image displays demonstrate how this property affects where automatically generated tags are placed in the server's tag space. It provides a root branch to which all automatically-generated tags will be added.

#### No sub group specified



#### Sub group named MyGroup specified



**Allow Sub Groups:** can be used to indicate that the server create sub groups matching object hierarchy for the device. When enabled this property causes the server to create the tags in subgroups conforming to the device object hierarchy. When disabled the tags will be created in a flat structure. The default value is enabled.

**Create:** manually initiates the creation of automatically-generated OPC tags, and also forces the communications driver to reevaluate the device for possible tag changes. It can be accessed from the System Tags, thus allowing the OPC client application to initiate tag database creation.

## Error Descriptions

---

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

### Address Validation

[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>' contains a syntax error](#)

[Device address '<address>' is Read Only](#)

[Missing address](#)

### 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 Yokogawa Darwin Ethernet Driver](#)

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

---

### Error Type:

Warning

### Possible Cause:

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

### Solution:

Verify 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 statically has been assigned an invalid data type.

### Solution:

Modify the requested data type in the client application.

## Device address '<address>' contains a syntax error

---

### Error Type:

Warning

### Possible Cause:

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

**Solution:**

Re-enter the address in the client application.

**Device address '<address>' is Read Only**

---

**Error Type:**

Warning

**Possible Cause:**

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

**Solution:**

Change the access mode in the client application.

**Missing address**

---

**Error Type:**

Warning

**Possible Cause:**

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

**Solution:**

Re-enter the address in the client application.

**Device '<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 connection cannot be established in the specified timeout period.
4. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device property.

**Solution:**

1. Verify the cabling between the PC and the PLC device.
2. Verify the IP address given to the named device matches that of the actual device.
3. Increase the Connect Timeout value in the Timeout property group of Device Properties.
4. Increase the Request Timeout property so that the entire response can be handled.



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

---

### Error Type:

Serious

### Possible Cause:

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

## 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 Yokogawa Darwin 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

## A

Address '<address>' is out of range for the specified device or register 55  
Address Descriptions 21  
Allow Sub Groups 15  
Attempts Before Timeout 13  
Auto-Demotion 13  
Automatic Tag Database Generation 53

## B

Boolean 20

## C

Channel Assignment 10  
Channel Properties — Advanced 9  
Channel Properties — Ethernet Communications 8  
Channel Properties — General 6  
Channel Properties — Write Optimizations 8  
Communications Timeouts 12  
Connect Timeout 12  
Create 16

## D

Data Collection 11  
Data Types Description 20  
DC100-1 Addressing 45  
DC100-1\_Addressing 21  
DC100-2 Addressing 48  
DC100-2\_Addressing 24  
Delete 15  
Demote on Failure 13  
Demotion Period 14  
Device '<device name>' is not responding 56

Device address '<address>' contains a syntax error 55  
Device address '<address>' is Read Only 56  
Device Configuration 17  
Device ID 5  
Device Properties — Auto-Demotion 13  
Device Properties — General 10  
Device Properties — Redundancy 18  
Device Properties — Tag Generation 14  
Device Properties — Timing 12  
Diagnostics 6  
Discard Requests when Demoted 14  
Do Not Scan, Demand Poll Only 12  
DR130 Addressing 42  
DR231 Addressing 28  
DR232 Addressing 31  
DR241 Addressing 35  
DR242 Addressing 38  
Driver 10  
Duty Cycle 9

## **E**

Error Descriptions 55  
Ethernet Settings 8

## **F**

Float 20

## **G**

General 10  
Generate 15

## **I**

ID 10  
Identification 6, 10

Initial Updates from Cache 12

Inter-Device Delay 9

## **M**

Missing address 56

Model 10

## **N**

Name 10

Network 5

Network Adapter 8

Non-Normalized Float Handling 9

## **O**

On Device Startup 14

On Duplicate Tag 15

On Property Change 14

Operating Mode 11

Optimization Method 8

Optimizing Ethernet Communications 19

Overview 4

Overwrite 15

## **P**

Parent Group 15

## **R**

Redundancy 18

Replace with Zero 9

Request Timeout 13

Respect Tag-Specified Scan Rate 12

**S**

Scan Mode 12

Setup 5

Short 20

Simulated 11

**T**

Tag Counts 7, 11

Tag Generation 14

Timeouts to Demote 13

Timing 12

**U**

Unable to write tag '<address>' on device '<device name>' 57

Unmodified 9

**W**

Winsock initialization failed (OSError = n) 57

Winsock V1.1 or higher must be installed to use the Yokogawa Darwin Ethernet device driver 57

Word 20

Write All Values for All Tags 8

Write Only Latest Value for All Tags 9

Write Only Latest Value for Non-Boolean Tags 8