Yokogawa GX Ethernet Driver

Table of Contents

Yokogawa GX Ethernet Driver

Help version 1.008

CONTENTS

Overview

The Yokogawa GX Ethernet Driver provides a reliable way to connect Yokogawa GX 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

GX10
GX20
GP10
GP20
GM10

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		Identification	
General		Name	
Write Optimizations		Description	
Advanced		Driver	
		Diagnostics	
		Diagnostics Capture	Disable

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. This property specifies the device driver that was selected during channel creation. It is a disabled setting in the channel properties. The property is required for creating a channel.

⬛ **Note**: With the server's online full-time operation, these properties can be changed at any time. This includes changing the channel name to prevent clients from registering data with the server. If a client has already acquired an item from the server before the channel name is changed, the items are unaffected. If, after the channel name has been changed, the client application releases the item and attempts to re-acquire using the old channel name, the item is not accepted. 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.*

Channel Properties — Ethernet Communications

Ethernet Communication can be used to communicate with devices.

Property Groups	Ethernet Settings	
General	Network Adapter	Default
Ethernet Communications		
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
Write Optimizations	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	Non-Normalized Float Handling	
General	Floating-Point Values	Replace with Zero
Write Optimizations	Inter-Device Delay	
Advanced	Inter-Device Delay (ms)	0

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

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

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

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

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

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

Device Properties — General

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

Property Groups	Identification	
General	Name	
Scan Mode	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. *For more information, refer to the driver's help documentation.*

Operating Mode

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

Simulated: 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 item's memory map is based on the group Update Rate. The data is not saved if the server removes the item (such as when the server is reinitialized). The default is No.

⬤ **Notes:**

1. This System tag (_Simulated) is read only and cannot be written to for runtime protection. The System tag allows this property to be monitored from the client.

2. In Simulation mode, the item's memory map is based on client update rate(s) (Group Update Rate for OPC clients or Scan Rate for native and DDE interfaces). This means that two clients that reference the same item with different update rates return different data.

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

Device Properties — Scan Mode

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

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

- **Respect Client-Specified Scan Rate**: This mode uses the scan rate requested by the client.
- **Request Data No Faster than Scan Rate**: This mode specifies the value set as the maximum scan rate. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
  ⬤ **Note**: When the server has an active client and items for the device and the scan rate value is

increased, the changes take effect immediately. When the scan rate value is decreased, the changes do not take effect until all client applications have been disconnected.

- **Request All Data at Scan Rate**: This mode forces tags to be scanned at the specified rate for subscribed clients. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
- **Do Not Scan, Demand Poll Only**: This mode does not periodically poll tags that belong to the device nor perform a read to get an item's initial value once it becomes active. It is the OPC client's responsibility to poll for updates, either by writing to the _DemandPoll tag or by issuing explicit device reads for individual items. *For more information, refer to "Device Demand Poll" in server help.*
- **Respect Tag-Specified Scan Rate**: This mode forces static tags to be scanned at the rate specified in their static configuration tag properties. Dynamic tags are scanned at the client-specified scan rate.

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

Device Properties — Timing

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

Property Groups	Communication Timeouts	
General	Connect Timeout (s)	3
Scan Mode	Request Timeout (ms)	1000
Timing	Attempts Before Timeout	3
Redundancy	Timing	
	Inter-Request Delay (ms)	0

Communications Timeouts

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

Request Timeout: 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 9,999,999 milliseconds (167.6667 minutes). The default is usually 1000 milliseconds, but can vary depending on the driver. The default timeout for most serial drivers is based on a baud rate of 9600 baud or better. When using a driver at lower baud rates, increase the timeout to compensate for the increased time required to acquire data.

Attempts Before Timeout: 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.

Device Properties — Auto-Demotion

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

Property Groups	Auto-Demotion	
General	Demote on Failure	**Enable**
Scan Mode	Timeouts to Demote	3
Timing	Demotion Period (ms)	10000
Auto-Demotion	Discard Requests when Demoted	Disable

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

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

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

Demotion Period: Indicate how long the device should be placed off-scan when the timeouts value is reached. During this period, no read requests are sent to the device and all data associated with the read requests are set to bad quality. When this period expires, the driver places the device on-scan and allows for another attempt at communications. The valid range is 100 to 3600000 milliseconds. The default is 10000 milliseconds.

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

Device Properties — 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	Tag Generation	
General	On Property Change	Yes
Scan Mode	On Device Startup	Do Not Generate on Startup
Timing	On Duplicate Tag	Delete on Create
Auto-Demotion	Parent Group	
Tag Generation	Allow Automatically Generated Subgroups	Enable
Redundancy	Create	Create tags

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

Port: Specify the port number that the remote device will be configured to use. This driver is currently set to use the Ethernet Exclusive port only (TCP port 34434).

/AS1 Security Option: When enabled, this option changes the login method to three parts: Username, User ID, and Password. When disabled, the driver uses the registered Username and Password login method if the device requires it. The default setting is disabled.

Username: Specify the registered username. If the device is configured with the login function enabled, only users that are registered can log in. The user name is case sensitive.

Password: Specify the username's registered password for when the device is configured with the login function enabled.

User ID: Specify the unique User ID when utilizing the /AS1 Security Option. The default setting is blank.
 Note: All device login properties are client access restricted. Changes can not be made to these properties while an OPC Client is accessing device tags. Reinitialize after changes are made.

Device Properties — Redundancy

Secondary Path: Specifies the explicit, unaliased path to the device serving as backup if the primary device fails. The valid format is <channel>.<Device>.

Operating Mode: Specify how the active device is chosen at runtime. Options include Switch On Failure, Switch On Trigger, Primary Only, and Secondary Only. The default setting is Switch On Failure. Descriptions of the options are as follows:

- **Switch On Failure:** In this mode, communication fails over to the secondary device when the primary device enters an error state (and vice versa). Initial communication begins on the primary device. The options are described below. For detailed information, refer to server help.
- **Switch On Trigger:** In this mode, the server monitors a configured trigger item. When the configured trigger condition becomes true, communication switches over to the secondary device. As long as the condition is false, communication remains on the primary device. Communication is initiated with the primary device until the trigger condition can be evaluated. The options are described below. For detailed information, refer to server help.

  ⬤ **Note**: For Switch On Trigger mode, the communication path is dictated entirely by the trigger state. Unlike the Switch On Failure mode, the communication path is not updated in the case of compromised physical communication.

- **Primary Only:** In this mode, communication is fixed to the primary device regardless of the error state. This option is used for maintenance when the secondary device needs to be taken offline.
- **Secondary Only:** In this mode, communication is fixed to the secondary device regardless of the error state. This option is used for maintenance when the primary device needs to be taken offline.

The following properties are only available when Switch On Trigger operating mode is enabled:

- **Trigger Item:** Specify a fully qualified dynamic or static item reference. This item is used as the source of the trigger and is regularly evaluated at the configured scan rate to determine if a switchover is required. To choose a static item reference defined in the server, use the Tag Browser by pressing the browse (...) button.

  ⬤ **Note**: The Trigger Item can be any server tag, but cannot be an array data type.

- **Scan Rate:** This is the frequency at which the Trigger item is evaluated
- **Trigger type:** This is the type of condition evaluated on the Trigger item. The supported trigger types are Quality Not Good, Value, and No Data Change.
    - **Quality Not Good:** When the trigger type is configured as Quality Not Good, the Trigger item change in quality to bad or uncertain causes communication to switch to the secondary device. When the Trigger item quality returns to good, communication switches back to the primary.
    - **Value:** When the trigger type is configured as Value, the item is compared using an operator from the drop-down list and data in the entry field. The value entered should be compatible with the data type of the enabled trigger item. When the trigger item value meets the condition defined by the configured string and operator, the trigger is considered "set" and communication switches from the primary to the secondary. When the trigger item value no longer meets the criteria, communication switches back to the primary. Available operators are equals (=), does not equal (≠), greater than (>), greater than or equal to (≥), less than (<), and less than or equal to (≤).

      ⬤ **Tips:** This field accepts an arbitrary length string of alpha-numeric characters. When the trigger item is a string type, comparisons are case insensitive ("Hello, World" and "HELLO, WORLD" are considered a match). When the trigger item is a float type (double or float type), care must be exercised because comparisons do not consider precision loss (1.499999 is not considered equal to 1.5).

Monitor Item: This optional property specifies a valid device tag name or address on both the primary and secondary devices that is used to monitor communications on both devices. Valid formats include the device address (such as "400001") or a partially qualified tag name (such as "Tag1") to identify a static tag at the device level (or "Group1.Tag1" to identify a tag in "Group1" and so forth). The default setting is blank.

⬤ **Note**: Only tags that require device communication may be used. *For more information, refer to server help.*

Monitor Interval: Specify the rate at which the Monitor Item is polled on each device in a redundant pair. It is the rate at which the error tag updates on the inactive device in a pair. The valid range is 30 to 999 seconds. The default setting is 300 seconds.

Return to Primary ASAP: This option specifies whether the primary device should become the active device as soon as possible after a failover. When enabled, the primary is re-activated as soon as possible following a transition of its error state from True to False.

Optimizing Communications

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

The server refers to communications protocols 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 m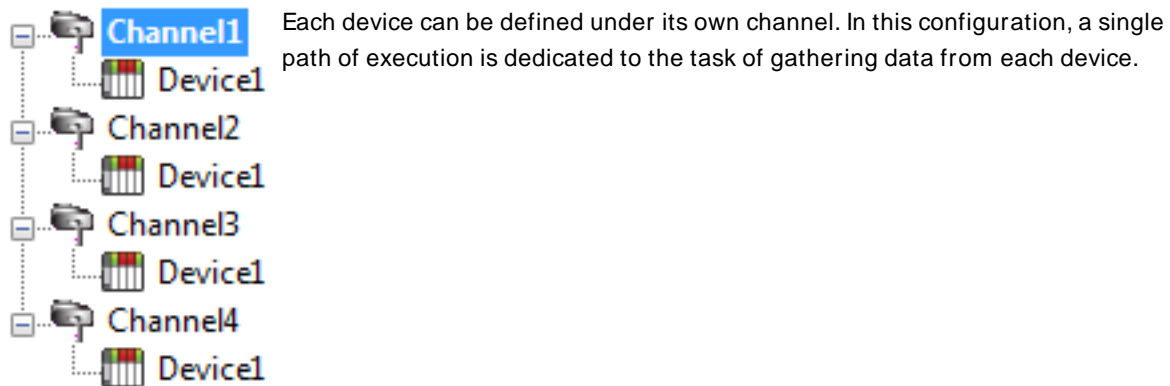ust then be defined under that channel. Each of these devices represents a single Ethernet device from which data will be collected. While this approach to defining the application will provide a high level of performance, it won't take full advantage of the Yokogawa GX Ethernet Driver or the network. An example of how the application may appear when configured using a single channel is shown below.

Each device appears under a single channel. In this configuration, the driver must move from one device to the next as quickly as possible to gather information at an effective rate. As more devices are added or more information is requested from a single device, the overall update rate begins to suffer.
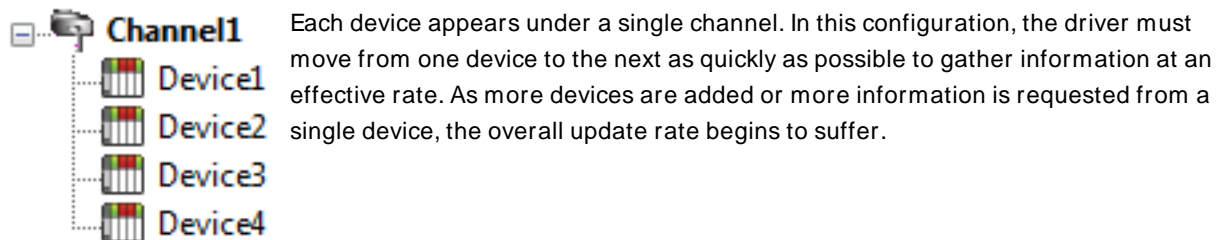
If the Yokogawa GX Ethernet Driver could only define one single channel, this example would be the only option available; however, the Yokogawa GX Ethernet Driver can define multiple channels. Using multiple channels distributes the data collection workload by simultaneously issuing multiple requests to the network.

Each device can be defined under its own channel. In this configuration, a single path of execution is dedicated to the task of gathering data from each device.

Data Types Description

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

Address Descriptions

Most of the tags are associated with configuration data and are only read during the initial communication with the device or when a non-zero value is written to the Refresh tag. All tags below marked with a '*' are read from the device continuously according to the update rate set by the client. Bold values are defaults.

General Device Data

Format	Address Type	Data Types	Access
Firmware	Specifies the firmware version of the main unit	**String**	Read Only
Options	Specifies the codes for the options that have been installed on the device	**String**	Read Only
ProductName	Specifies the product name of the device	**String**	Read Only
Refresh	Writing a non-zero value to this tag causes a refresh of all the configurations values	**Boolean**	Write Only
SerialNumber	Specifies the serial number of the device	**String**	Read Only
UnitStatus	Specifies the status of the device	**String**	Read Only

Communication Channel Tags

All communication tags start with the letter C, followed by the three-digit channel number (shown as xxx below).

Tag Format	Address Type	Range	Data Types	Access
Cxxx.Alarm*	Specifies the alarm summary for this channel. Will be 1 if an alarm is currently active.	xxx = 1-999	**Boolean**	Read Only
Cxxx.Alarmzz.Detection	Specifies whether the alarm detection function will be used	xxx = 1-999 zz = 1-4	**Boolean**	Read Only
Cxxx.Alarmzz.Output	Specifies whether the alarm will output to a relay or internal switch	xxx = 1-999 zz = 1-4	**String**	Read Only
Cxxx.Alarmzz.OutputNumber	Specifies the relay or internal switch number	xxx = 1-999 zz = 1-4	**Word**, Short, DWord, Long	Read Only
Cxxx.Alarmzz.Status*	Specifies the current status of the alarm. Possible values are: 0 = None 1 = High limit 2 = Low limit 3 = Difference high limit 4 = Difference low limit 5 = High limit on rate-of-change 6 = Low limit on rate-of-change	xxx = 1-999 zz = 1-4	**Byte**, Word, Short, DWord, Long	Read Only

Tag Format	Address Type	Range	Data Types	Access
	7 = Delay high limit 8 = Delay low limit			
Cxxx.Alarmzz.Type	Specifies the type of alarm (H, L, TH, TL)	xxx = 1-999 zz = 1-4	**String**	Read Only
Cxxx.Alarmzz.Value	Specifies the value associated with the alarm	xxx = 1-999 zz = 1-4	**Long**, DWord	Read Only
Cxxx.DecimalPlace	Specifies the decimal place for the span and Value values	xxx = 1-999	**Byte**, Word, Short, DWord, Long	Read Only
Cxxx.Enabled	Specifies whether the communication channel is enabled	xxx = 1-999	**Boolean**	Read Only
Cxxx.PV*	Specifies the process value	xxx = 1-999	**String**, Double	Read Only
Cxxx.SpanLower	Specifies the lower span	xxx = 1-999	**Long**, DWord	Read Only
Cxxx.SpanUpper	Specifies the upper span	xxx = 1-999	**Long**, DWord	Read Only
Cxxx.Status*	Specifies the current status of the channel. Possible values are: 0 = No error 1 = Skip 2 = +Over 3 = -OVER 4 = +Burnout 5 = -Burnout 6 = A/D error 7 = Invalid data 16 = Math result is NaN 17 = Communication error	xxx = 1-999	**Byte**, Word, Short, DWord, Long	Read Only
Cxxx.TagName	Specifies the tag name	xxx = 1-999	**String**	Read Only
Cxxx.TagNumber	Specifies the tag number	xxx = 1-999	**String**	Read Only
Cxxx.Unit	Specifies the units	xxx = 1-999	**String**	Read Only

Input / Output Channel Tags

All input / output tags start with the four-digit channel number (shown as xxxx below). The channel number is a representation of the [unit-number][slot-number][channel] as a single value.

Format	Address Type	Range	Data Types	Access
xxxx.Alarm*	Specifies the alarm summary for this channel. Will be 1 if an alarm is currently active	xxxx = 1-9999	**Boolean**	Read Only
xxxx.Alarmzz.Detection	Specifies whether the	xxxx = 1-9999	**Boolean**	Read Only

Format	Address Type	Range	Data Types	Access
	alarm detection function will be used	zz = 1-4		
xxxx.Alarmzz.Output	Specifies whether the alarm will output to a relay or internal switch	xxxx = 1-9999 zz = 1-4	**String**	Read Only
xxxx.Alarmzz.OutputNumber	Specifies the relay or internal switch number	xxxx = 1-9999 zz = 1-4	**Word**, Short, DWord, Long	Read Only
xxxx.Alarmzz.Status*	Specifies the current status of the alarm. Possible values are: 0 = None 1 = High limit 2 = Low limit 3 = Difference high limit 4 = Difference low limit 5 = High limit on rate-of-change 6 = Low limit on rate-of-change 7 = Delay high limit 8 = Delay low limit	xxxx = 1-9999 zz = 1-4	**Byte**, Word, Short, DWord, Long	Read Only
xxxx.Alarmzz.Type	Specifies the type of alarm (H, L, TH, TL)	xxxx = 1-9999 zz = 1-4	**String**	Read Only
xxxx.Alarmzz.Value	Specifies the value associated with the alarm	xxxx = 1-9999 zz = 1-4	**Long**, DWord	Read Only
xxxx.Bias	Specifies the Bias of the channel	xxxx = 1-9999	**Long**, DWord	Read Only
xxxx.CalcType	Specifies the calculation type of the channel	xxxx = 1-9999	**String**	Read Only
xxxx.DecimalPlace	Specifies the decimal place for the span and Value values	xxxx = 1-9999	**Byte**, Word, Short, DWord, Long	Read Only
xxxx.Enabled	Specifies whether the communication channel is enabled	xxxx = 1-9999	**Boolean**	Read Only
xxxx.Energize	Specifies whether the channel has been set to Energize or De-energize	xxxx = 1-9999	**String**	Read Only
xxxx.Hold	Specifies whether the channel has been set to Hold or Non-hold	xxxx = 1-9999	**String**	Read Only
xxxx.IOType	The type of the input or output	xxxx = 1-9999	**String**	Read Only
xxxx.Operation	Specifies when the channel should operate (And, Or)	xxxx = 1-9999	**String**	Read Only

Format	Address Type	Range	Data Types	Access
xxxx.PresetValue	Specifies the Preset value	xxxx = 1-9999	Short	Read Only
xxxx.PV*	Specifies the process values.	xxxx = 1-9999	**Float**, Double	Read Only
xxxx.Range	Specifies the range based on the type of input. See the device's User's Manual for more information and possible values.	xxxx = 1-9999	**String**	Read Only
xxxx.RefChannelNum	Specifies the reference channel number	xxxx = 1-9999	**Word**, Short, DWord, Long	Read Only
xxxx.RefChannelType	Specifies the reference channel type (Input, Math, Com)	xxxx = 1-9999	**String**	Read Only
xxxx.ReflashTime	Specifies the reflash time	xxxx = 1-9999	**String**	Read Only
xxxx.RelayAction	Specifies the relay action on an ACK (Normal, Reset)	xxxx = 1-9999	**String**	Read Only
xxxx.ScalingHigh	Specifies the scaling high value when scaling is enabled	xxxx = 1-9999	**Float**, Double	Read Only
xxxx.ScalingLow	Specifies the scaling low value when scaling is enabled	xxxx = 1-9999	**Float**, Double	Read Only
xxxx.SpanLower	Specifies the lower span	xxxx = 1-9999	**Long**, DWord	Read Only
xxxx.SpanUpper	Specifies the upper span	xxxx = 1-9999	**Long**, DWord	Read Only
xxxx.Status*	Specifies the current status of the channel. Possible values are: 0 = No error 1 = Skip 2 = +Over 3 = -OVER 4 = +Burnout 5 = -Burnout 6 = A/D error 7 = Invalid data 16 = Math result is NaN 17 = Communication error	xxxx = 1-9999	**Byte**, Word, Short, DWord, Long	Read Only
xxxx.TagName	Specifies the tag name	xxxx = 1-9999	**String**	Read Only
xxxx.TagNumber	Specifies the tag number	xxxx = 1-9999	**String**	Read Only
xxxx.Unit	Specifies the units	xxxx = 1-9999	**String**	Read Only

Math Channel Tags

All math tags start with the letter A followed by the three-digit channel number (shown as xxx below).

Format	Address Type	Range	Data Types	Access
Axxx.Alarm*	Specifies the alarm summary for this channel. Will be 1 if an alarm is currently active.	xxx = 1-999	**Boolean**	Read Only
Axxx.Alarmzz.Detection	Specifies whether the alarm detection function will be used	xxx = 1-999 zz = 1-4	**Boolean**	Read Only
Axxx.Alarmzz.Output	Specifies whether the alarm will output to a relay or internal switch	xxx = 1-999 zz = 1-4	**String**	Read Only
Axxx.Alarmzz.OutputNumber	Specifies the relay or internal switch number	xxx = 1-999 zz = 1-4	**Word**, Short, DWord, Long	Read Only
Axxx.Alarmzz.Status*	Specifies the current status of the alarm. Possible values are: 0 = None 1 = High limit 2 = Low limit 3 = Difference high limit 4 = Difference low limit 5 = High limit on rate-of-change 6 = Low limit on rate-of-change 7 = Delay high limit 8 = Delay low limit	xxx = 1-999 zz = 1-4	**Byte**, Word, Short, DWord, Long	Read Only
Axxx.Alarmzz.Type	Specifies the type of alarm (H, L, TH, TL)	xxx = 1-999 zz = 1-4	**String**	Read Only
Axxx.Alarmzz.Value	Specifies the value associated with the alarm	xxx = 1-999 zz = 1-4	**Long**, DWord	Read Only
Axxx.DecimalPlace	Specifies the decimal place for the span and Value values	xxx = 1-999	**Boolean**	Read Only
Axxx.Enabled	Specifies whether the communication channel is enabled	xxx = 1-999	**Byte**, Word, Short, DWord, Long	Read Only
Axxx.PV*	Specifies the process value	xxx = 1-999	**Float**, Double	Read Only
Axxx.SpanLower	Specifies the lower span	xxx = 1-999	**Long**, DWord	Read Only
Axxx.SpanUpper	Specifies the upper span	xxx = 1-999	**Long**, DWord	Read Only
Axxx.Status*	Specifies the current status of the channel. Possible values are: 0 = No error 1 = Skip 2 = +Over	xxx = 1-999	**Byte**, Word, Short, DWord, Long	Read Only

Format	Address Type	Range	Data Types	Access
	3 = -OVER 4 = +Burnout 5 = -Burnout 6 = A/D error 7 = Invalid data 16 = Math result is NaN 17 = Communication error			
Axxx.TagName	Specifies the tag name	xxx = 1-999	**String**	Read Only
Axxx.TagNumber	Specifies the tag number	xxx = 1-999	**String**	Read Only
Axxx.Unit	Specifies the units	xxx = 1-999	**String**	Read Only

Index

A

B

C

D

P

Parent Group 12

R

Redundancy 13
Request Timeout 9
Respect Tag-Specified Scan Rate 9

S

Scan Mode 8
Setup 3
Short 17
Simulated 8

T

Tag Generation 11
Timeouts to Demote 10

W

Word 17