

Universal Device Driver

© 2025 PTC Inc. All Rights Reserved.

Table of Contents

Universal Device Driver	1
Table of Contents	2
Welcome to the Universal Device Driver Help Center	4
Overview	4
Architecture	5
General Operation	5
Channel Properties – General	6
Tag Counts	6
Channel Properties – Ethernet Communications	6
Channel Properties – Write Optimizations	7
Channel Properties – Advanced	7
Channel Properties – Profile	8
Setup	9
Device Properties – General	9
Operating Mode	9
Tag Counts	10
Device Properties – Scan Mode	10
Device Properties – Communication Transport	11
Device Properties – Timing	13
Device Properties – Auto-Demotion	14
Device Properties – File Mode	15
Device Properties – User-Defined Settings	15
Device Properties – Redundancy	16
Data Types Descriptions	17
Event Log Messages	18
Invalid profile link. Channel = '<channel name>', Profile ID = '<id>'	18
Connection to the service failed. Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'	18
Failed to start service. Channel name = '<name>', service name = '<name>'	18
Profile registration failed because the profile syntax is bad or is missing required functions. Channel name '<name>'	18
Script failure. Result = '<result value>'	19
Failed to convert value from script to '<data type>' for address '<address>'	19
Could not validate address because profile invocation failed. Address = '<address>'	19
State machine unknown (required script function GetDriverInfo is missing or returned a bad result). Channel name '<name>'	19
Could not build a send frame because the script execution failed. Address = '<address>', profile ID = '<id>'	19
Failed to process received frame. Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'	20
Failed to generate write request. Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'	20
Failed to process write response. Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'	20

ValidateAddress() returned an invalid result. Result = '<result value>'.	20
Could not validate address because state machine version is unknown. Address = '<address>'.	20
Failed to process response. Type = '<read/write>', Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'.	21
Script function failed. Channel = '<channel name>'	21
Send data failed. Channel = '<channel name>'	21
Receive data failed. Channel = '<channel name>'	21
Script execution failed. Channel = '<channel name>'	22
Script result invalid. Profile = '<profile name>', Reason = '<reason string>'.	22
Script invocation failed. Channel = '<name>', reason = '<reason string>'.	22
Address validation failed. Channel = '<name>', address = '<tag address>', reason = '<reason string>'. ..	23
Script mode does not match the configured channel profile mode. Channel = '<channel name>'.	23
Profile Inputs failed validation. Channel = '<name>'.	23
Profile link established. Channel = '<name>', profile = '<name>'.	24
Profile link established. Channel = '<name>', profile = '<name>', state machine version = '<version>'. ..	24
Script engine error detected. Temporarily suspending communication. Channel = '<name>'.	24
Profile link established. Channel = '<name>', profile = '<name>', state machine version = '<version>', mode = '<operating mode>'.	24
Connection limit reached. Channel = '<name>'.	24
Sample Profiles	25
Index	26

Welcome to the Universal Device Driver Help Center

This help center is the user documentation for Kepware Universal Device Driver. This help center is updated regularly to reflect the latest functionality and information.

Overview

What is the Universal Device Driver?

Architecture

How does the Universal Device Driver work?

General Operation

How do I start using the Universal Device Driver?

Setup

How do I configure a device for the Universal Device Driver?

Tag Generation

How do I collect data from my devices with the Universal Device Driver?

Data Types Descriptions

What kinds of data can I collect from my devices?

Event Log Messages

What messages are produced by the driver?

Connectivity Guides

Where can I get more information about Configuring Universal Device Driver and Profile Library Plug-in?

Account login required for PTC.com support resources

Sample Profiles

Where can I see example of profiles and code?

Version 1.039

© 2025 PTC Inc. All Rights Reserved.

Overview

The Universal Device Driver provides a reliable way for end users to connect to a wide variety of Ethernet protocols and devices using profiles from the Profile Library Plug-in. The Universal Device Driver also provides the ability for end users to access files when configured with Profile File Mode and using a File Mode profile. The Universal Device Driver provides an interface for mapping existing profiles to devices in the environment through communication channels. The Profile Library Plug-in allows users to create JavaScript scripts that define the functions and parameters necessary for the driver to communicate with a device or operate on files. In addition to the script, the profile contains a universally unique identifier (UUID) used to link the channel to the profile and standard server properties, such as the profile name and description.

 *For more information about profiles, consult the Profile Library Plug-in help file.*

Architecture

The Universal Device Driver works in conjunction with the Profile Library Plug-in to enable users to create custom drivers that communicate with devices or operate on files in the environment.

Service

Installing the Universal Device Driver adds a new server component, the Script Engine. The service is an isolated environment to run the V8 JavaScript engine. The user does not interact with the service; all interaction is handled by the server. The service starts when the driver registers a profile script to execute and stops five minutes after use.

Using an external service for the JavaScript engine protects the server runtime from unexpected JavaScript failures or problems. Measures have been introduced into the service to mitigate the impact of these issues and to prevent the Script Engine from entering an unresponsive state. The Script Engine enters a cool-off state when one of the following circumstances occurs:

- The service itself cannot be started;
- A profile script cannot be registered (i.e., invalid script); or
- A script invocation times out.

The cool-off state terminates after 30 seconds or on a profile change event (such as a different profile ID is used or the profile script is altered).

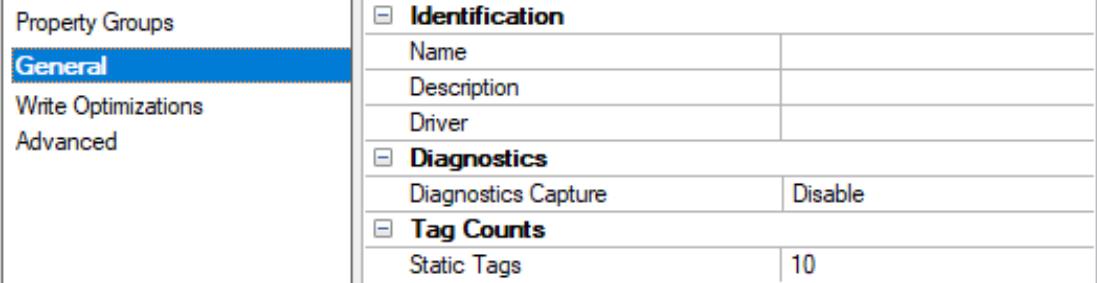
General Operation

The Universal Device Driver can be configured either via the Configuration API Service or through the server configuration user interface. In addition to many of the standard server properties, Universal Device Driver contains a **Profile** section under channel properties.

When building a channel, the ID of the profile for communicating with a device must be specified. This requires that a profile exist before attempting to create a channel.

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	
General	
Write Optimizations	
Advanced	
Identification	
Name	
Description	
Driver	
Diagnostics	
Diagnostics Capture	Disable
Tag Counts	
Static Tags	10

Identification

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

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

Description: Specify user-defined information about this channel.

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

Driver: Specify the protocol / driver for this channel. Specify the device driver that was selected during channel creation. It is a disabled setting in the channel properties. The property is required for creating a channel.

● **Note:** With the server's online full-time operation, these properties can be changed at any time. This includes changing the channel name to prevent clients from registering data with the server. If a client has already acquired an item from the server before the channel name is changed, the items are unaffected. If, after the channel name has been changed, the client application releases the item and attempts to re-acquire using the old channel name, the item is not accepted. Changes to the properties should not be made once a large client application has been developed. Utilize proper user role and privilege management to prevent operators from changing properties or accessing server features.

Diagnostics

Diagnostics Capture: When enabled, this option makes the channel's diagnostic information available to OPC applications. Because the server's diagnostic features require a minimal amount of overhead processing, it is recommended that they be utilized when needed and disabled when not. The default is disabled.

● **Note:** This property is not available if the driver or operating system does not support diagnostics.

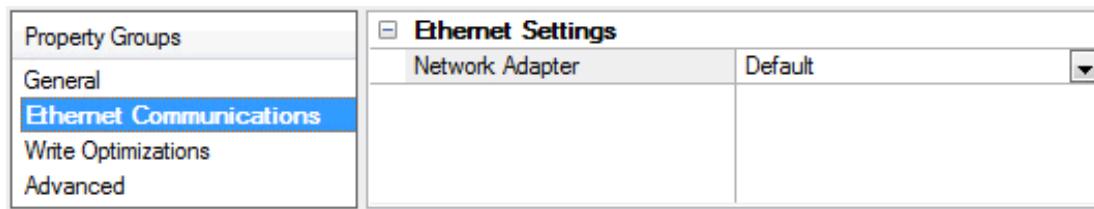
● For more information, refer to *Communication Diagnostics and Statistics Tags* in server help.

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.

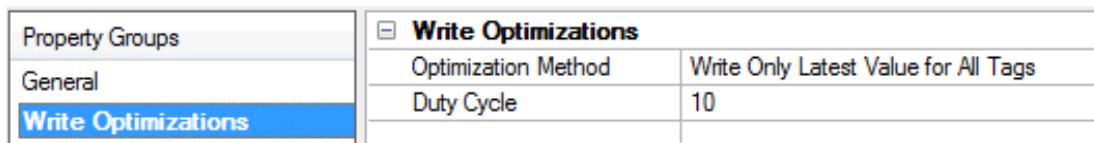


Ethernet Settings

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.



Write Optimizations

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

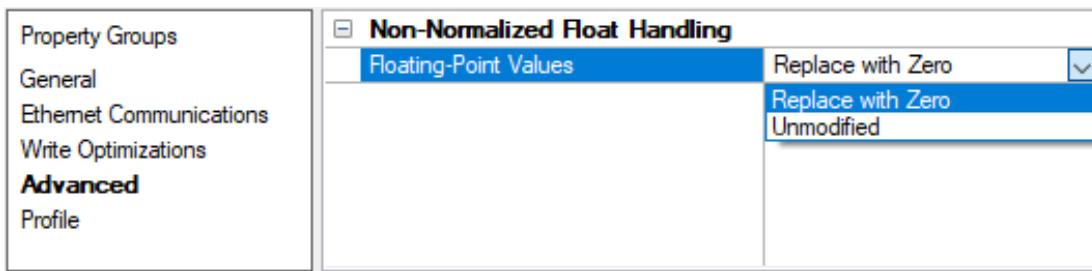
- **Write All Values for All Tags:** This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing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

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

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

Channel Properties – Advanced

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



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

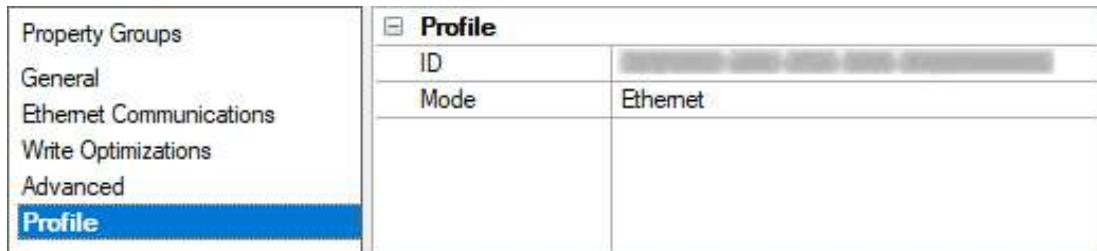
- **Replace with Zero:** This option allows a driver to replace non-normalized IEEE-754 floating point values with zero before being transferred to clients.
- **Unmodified:** This option allows a driver to transfer IEEE-754 denormalized, normalized, non-number, and infinity values to clients without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

● **Note:** This property is 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.

Channel Properties – Profile

Profile properties are specific to Universal Device Driver channels.



ID: Specify a unique identifier to link device channels to profiles defined in the Profile Library Plug-in. The ID must be in a GUID format. Changing the ID updates the associated profile in the Profile Library Plug-in. This property can be changed at any time, including while there are active client references on the channel, but bad quality data is reported while the channel is re-linking to the profile. Once the channel is successfully linked to the profile, the driver runs all the tag addresses through the profile's onValidateTag function. Any that fail validation in the profile's onValidateTag function continue to report bad quality data. All tags that pass validation and successfully parse a response frame from the device return good quality data when the profile re-linking is complete.

● A profile must exist in the Profile Library Plug-in and the GUID generated there before it can be assigned to a channel.

Mode: Specify Ethernet when the mode configured within the profile script is "Client" or "Server", indicating communication with devices is via Ethernet. File Mode is for profile scripts to perform file operations.

● **Notes:**

1. Ethernet Communications channel property group is only available for Ethernet mode; hidden for File Mode.
2. Communication Transport device Properties group is available for Ethernet Mode; hidden for File Mode.
3. File Mode Device Properties group is available for File Mode; hidden for Ethernet Mode.

● **See Also:** See [File Mode](#) for more information.

Setup

A device can be configured through the Configuration API Service or through the server configuration user interface.

The Universal Device Driver allows a maximum of 1024 channels with 1 device per channel.

Green Channel-level settings apply to all devices configured on that channel.

Notes:

- A profile must exist before a channel can be created.
- In [Device Properties](#) the Device ID is set to 1 as this driver has a limit of 1 device per channel.

Device Properties – General

A device represents a single target on a communications channel.

Property Groups																					
General																					
Scan Mode																					
Timing																					
Auto-Demotion																					
Communication Transport Prop...																					
Redundancy																					
<div style="border: 1px solid #ccc; padding: 5px;"> Identification <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Name</td><td>UDD1</td></tr> <tr><td>Description</td><td></td></tr> <tr><td>Driver</td><td>Universal Device</td></tr> <tr><td>Model</td><td>Ethernet</td></tr> <tr><td>Channel Assignment</td><td>UniversalDeviceDriver</td></tr> <tr><td>ID Format</td><td>Decimal</td></tr> <tr><td>ID</td><td>1</td></tr> </table> Operating Mode <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Data Collection</td><td>Enable</td></tr> <tr><td>Simulated</td><td>No</td></tr> </table> Tag Counts <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Static Tags</td><td>0</td></tr> </table> </div>		Name	UDD1	Description		Driver	Universal Device	Model	Ethernet	Channel Assignment	UniversalDeviceDriver	ID Format	Decimal	ID	1	Data Collection	Enable	Simulated	No	Static Tags	0
Name	UDD1																				
Description																					
Driver	Universal Device																				
Model	Ethernet																				
Channel Assignment	UniversalDeviceDriver																				
ID Format	Decimal																				
ID	1																				
Data Collection	Enable																				
Simulated	No																				
Static Tags	0																				

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.

Green 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.

Operating Mode

Property Groups					
General					
Identification <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Data Collection</td><td>Enable</td></tr> <tr><td>Simulated</td><td>No</td></tr> </table>		Data Collection	Enable	Simulated	No
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 <div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 5px;">General</div>	<input type="checkbox"/> Identification
	<input type="checkbox"/> Operating Mode
	<input type="checkbox"/> 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 <div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 5px;">General</div> <div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 5px;">Scan Mode</div>	<input type="checkbox"/> Scan Mode
	Scan Mode Respect Client-Specified Scan Rate
	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 – Communication Transport

The Communication Transport Properties group is available when the Channel Profile Mode is configured for Ethernet Mode. The available device properties then depend on the selected IP Protocol - TCP/IP or UDP.

Communication Transport Properties – TCP/IP

Property Groups	
General	
Scan Mode	
Timing	
Auto-Demotion	
Communication Transport ...	
Redundancy	
Communication Transport Properties	
IP Protocol	TCP/IP
Source IP or Hostname	0.0.0.0
Destination IP or Hostname	255.255.215.255
Port Number	8501
TCP Keep-Alive	
TCP Keep-Alive	Enable
TCP Connection Idle Time (s)	7200
TCP Keep-Alive Count	10
TCP Keep-Alive Interval (s)	1

IP Protocol: Specify the Ethernet communication protocol: **TCP/IP**.

Source IP or Hostname: Verify the IP address of the selected network adapter. This property is configured on the channel in the Ethernet Communications properties.

Destination IP or Hostname: Specify the destination IP or host name of the device that communicates with the driver. ([Client Mode](#) only)

Port Number: Identify the port for the driver to communicate with the device.

TCP Keep-Alive: Specify if a keep-alive probe should be sent to the remote device. The options are Enable or Disable (default). ([Server Mode](#) only)

TCP Connection Idle Time (s): Specify the number of seconds a TCP connection remains idle before sending a keep-alive probe to the remote device. The valid range is 1 to 86400 seconds. The default is 7200 seconds.

TCP Keep-Alive Count: Specify the number of keep-alive probes sent without reply before the connection is closed. The valid range is 1 to 255 probes. The default is 10 probes.

TCP Keep-Alive Interval: Specify the number of seconds a TCP connection waits for a keep-alive probe response before sending. The valid range is 1 to 9999 seconds. The default is 1 second.

Tip: Changes to the communication transport properties are *not* applied if there is an active connection. The Ethernet client (not northbound client) must disconnect and reconnect to use new settings. ([Server Mode](#) only)

Communication Transport Properties – UDP

Property Groups		Communication Transport Properties	
General		IP Protocol	UDP
Scan Mode		Source IP or Hostname	0.0.0.0
Timing		Source Port Number	5000
Auto-Demotion		Destination IP or Hostname	127.0.0.1
Communication Transport ...		Destination Port Number	503
Redundancy			

IP Protocol: Specify the Ethernet communication protocol: **UDP**.

Source IP or Hostname: Verify the IP address of the selected network adapter the driver uses to communicate. This property is configured on the channel in the Ethernet Communications properties.

Source Port Number: Specify the local port the driver binds to, to receive responses from the device (required).

Tip: The Source Port and Destination Port must both be provided to allow the UDP server to accept requests and send responses to the specified destination.

Destination IP or Hostname: Specify the destination IP or host name of the device that communicates with the driver.

Destination Port Number: Specify the port the driver uses to communicate with the device (required).

Tip: The Source Port and Destination Port must both be provided to allow the UDP server to accept requests and send responses to the specified destination.

Note: Once the driver has established a connection to the device to scan one or more tags, that connection is maintained until all clients have unsubscribed or the runtime is shutdown.

Tip: Changes to the communication transport properties are not applied if there is an active connection. The Ethernet client (not northbound client) must disconnect and reconnect to use new settings. ([Server Mode](#) only)

Client / Server Modes

The onProfileLoad function within the profile script provides the mode that defines how the driver operates for this device. The available modes when the channel profile Mode property is configured with Ethernet are “Client” or “Server”. The source and destination properties in the Communication Transport device properties have different meaning when the loaded profile indicates “Client” mode than when it indicates “Server”. The following example illustrates the meaning of these Source and Destination properties based on the ‘Client Server’ communication example, provided in the Examples folder of the install folder, running on the local machine. In this example, two channels are communicating with each other:

- Client is locally bound to port 5000 and communicates with a local server on port 503.
- Server is locally bound to port 503 and communicates with the local client on port 5000.

Example Client Loopback Settings

Property Groups		Communication Transport Properties	
General		IP Protocol	UDP
Scan Mode		Source IP or Hostname	0.0.0.0
Timing		Source Port Number	5000
Auto-Demotion		Destination IP or Hostname	127.0.0.1
Communication Transport ...		Destination Port Number	503
Redundancy			

Example Server Loopback Settings

Property Groups		Communication Transport Properties	
General		IP Protocol	UDP
Scan Mode		Source IP or Hostname	0.0.0.0
Timing		Source Port Number	503
Auto-Demotion		Destination IP or Hostname	127.0.0.1
Communication Transport ...		Destination Port Number	5000
Redundancy			

Note: The IP address 0.0.0.0 refers to the system's default adapter (which on this machine is equivalent with the local loopback adapter). The IP address 127.0.0.1 refers to the local loopback adapter (AKA local host).

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

Communications Timeouts

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

Note: Due to the nature of UDP connections, the connection timeout setting is not applicable when communicating via UDP.

Request Timeout: Specify an interval used by all drivers to determine how long the driver waits for a response from the target device to complete. The valid range is 50 to 9999999 milliseconds (167 minutes). The default is usually 1000 milliseconds, but can vary depending on the driver. The default timeout for most serial drivers is based on a baud rate of 9600 baud or better. When using a driver at lower baud rates, increase the timeout to compensate for the increased time required to acquire data.

Note: When the channel profile mode property is configured for File Mode, the Request Timeout is applied at the transaction level. A transaction is one or more file operations performed while processing a tag. The transaction must be completed within the configured Request Timeout or it is considered a timeout. If Attempts Before Timeout have been exhausted, the script is notified of the timeout for the last file operation performed.

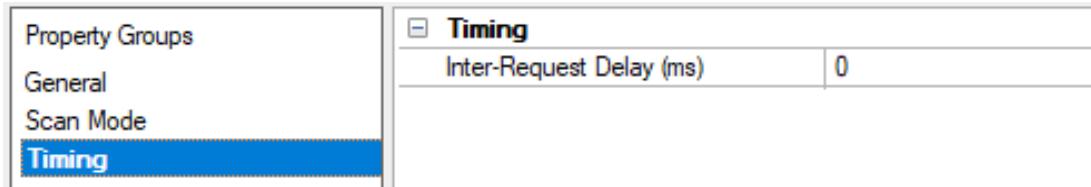
Attempts Before Timeout: Specify how many times the driver issues a communications request before considering the request to have failed and the device to be in error. The valid range is 1 to 10. The default is typically 3, but can vary depending on the driver's specific nature. The number of attempts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

Note: When the channel profile mode property is configured for File Mode, the Attempts Before Timeout is applied at the transaction level. A transaction is one or more file operations performed while processing a tag and limited to a total of Attempts Before Timeout attempts. This means a failure of one operation reduces the total number of Attempts Before Timeout for the entire transaction by one. When all Attempts Before Timeout have been exhausted, the transaction is considered a failure. The script is notified of this failure for the last file operation performed.

Timing

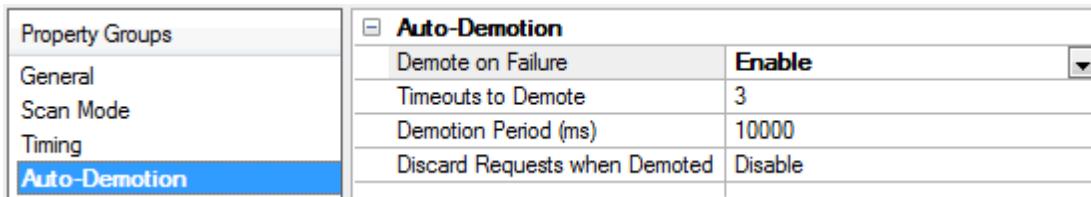
Inter-Request Delay: Specify how long the driver waits before sending the next request to the target device. It overrides the normal polling frequency of tags associated with the device, as well as one-time reads and writes. This delay can be useful when dealing with devices with slow turnaround times 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.



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 – File Mode

When the channel profile Mode property is configured with File Mode, the driver requires a base path that exists to utilize file operations.

Property Groups	
General	
Scan Mode	
Timing	
Auto-Demotion	
File Mode	
Redundancy	

File Mode	
Base Path	
File Name	
Max File Size (kB)	200

Base Path: Specify the directory where the files are located. The default is empty; however, a device cannot be created without populating this property.

 **Tips:**

- The path cannot be empty and the directory must already exist.
- The Base Path supports local Windows file share locations. The configured Base Path must be accessible by the user that is operating the server runtime process.
 - If the process is running interactively, the user must have read / write / delete access to the Base Path.
 - If the process is running as a system service, the log on account for the service must have read / write / delete access to the Base Path.

 **Caution:** The path must comply with the following:

- Must be between 3 and 247 characters
- Must not contain any character or symbols forbidden by the system
- Must be an absolute path; beginning with a drive letter or UNC path
- Must not refer to a system directory
- Must not use directory traversal

File Name: Specify the file name, without the path, to be used in File Mode operations. This property is optional. If not defined, the file name must be provided in file operations request from the script. The File Name may not contain any special characters (< > : " \ | ? *), end with a period, or exceed 244 characters.

 **Tip:** The fully-qualified file path, which includes the Base Path, a separator, and the File Name, must not exceed 258 characters.

Max. File Size : Specify the maximum size for files that the script can access. Attempting to interact with files larger than the specified size results in an error. The valid range for maximum file size is 10 to 200 kB. The default is 200 kilobytes.

Device Properties – User-Defined Settings

Property Groups	
General	
Scan Mode	
Timing	
Auto-Demotion	
Communication Tra...	
User-Defined Set...	
Redundancy	

Profile Inputs	
Input String	UDDString

Input String: Specify the input string the driver passes to the profile through the OnProfileInputsChange API function.

● **Tips:**

- If the Input String remains empty (the default value), OnProfileInputsChange is not called.
- OnProfileInputsChange is called after the Input String changes and immediately before tags are processed.

● **Caution:**

- The Input String cannot be more than 20,480 characters. If a longer input string is entered, it is truncated.
- The property can accept Unicode characters, but they are converted to UTF-8 when passed to the script.

Device Properties – Redundancy

Property Groups	
General	
Scan Mode	
Timing	
Auto-Demotion	
Redundancy	
Redundancy	
Secondary Path	Channel.Device1 ...
Operating Mode	Switch On Failure
Monitor Item	
Monitor Interval (s)	300
Return to Primary ASAP	Yes

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

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

Data Types Descriptions

A tag must be assigned a data type when added to the server project. The driver allows any of the data types supported by the server to be assigned to the tag, including many of the data types as arrays. The tag validation occurs when the driver calls the profile's `onValidateTag` function, which should validate the tag address, data type, and read-only mode. When completing a tag, the profile's `onTagsRequest`, `onData`, `onFileTagsRequest`, and `onFileOperations` functions are responsible for providing the data for the tag in the expected format for the tag data type.

Data Types Supported by the Server

Data Type, Array Data Type (if supported)	Description
Default	The profile assigns the data type.
BCD	Two byte-packed BCD value. Range is 0-9999.
Boolean, Boolean Array	Binary value of true or false
Byte, Byte Array	Unsigned 8-bit integer data
Char, Char Array	Signed 8-bit integer data
Date	8-byte floating point number
Double, Double Array	64-bit real value IEEE-754 standard definition
DWord, DWord Array	Unsigned 32-bit integer data
Float, Float Array	32-bit real value IEEE-754 standard definition
LBCD	Four byte-packed BCD value. Range is 0-99999999.
Long, Long Array	Signed 32-bit integer data
LLong, LLong Array	Signed 64-bit integer data
QWord, QWord Array	Unsigned 64-bit integer data
Short, Short Array	Signed 16-bit integer data
String, String Array	Null-terminated Unicode string. Note: String array elements truncate at 1000 characters each.
Word, Word Array	Unsigned 16-bit integer data

Event Log Messages

The following information concerns messages posted to the Event Log pane in the main user interface. Consult the OPC server help on filtering and sorting the Event Log detail view. Server help contains many common messages, so should also be searched. Generally, the type of message (informational, warning) and troubleshooting information is provided whenever possible.

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

Invalid profile link. | Channel = '<channel name>', Profile ID = '<id>'.

Error Type:

Error

Possible Cause:

There is no profile that matches the identifier currently assigned to the driver.

Possible Solution:

Verify the profile ID is valid for this channel.

Connection to the service failed. | Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'.

Error Type:

Error

Possible Cause:

The service has been disabled or the port is in use.

Possible Solution:

Enable the service or reassign the port using the system administration utility.

Failed to start service. | Channel name = '<name>', service name = '<name>'.

Error Type:

Error

Possible Cause:

The service has been disabled or the binary has been tampered with.

Possible Solution:

Enable the service or reinstall the product.

Profile registration failed because the profile syntax is bad or is missing required functions. | Channel name '<name>'.

Error Type:

Error

Possible Cause:

The profile script syntax is bad or is missing required functions.

Possible Solution:

Correct the syntax errors or implement the missing functions.

Script failure. | Result = '<result value>'.**Error Type:**

Error

Possible Cause:

1. Return type was invalid
2. Returned object was missing fields
3. Field types were invalid
4. Field is an enumeration, but the value does not match the enumerated type

Possible Solution:

Ensure that the script returns a JavaScript object with the required fields.

Failed to convert value from script to '<data type>' for address '<address>'.**Error Type:**

Error

Possible Cause:

The value provided is not convertable to the requested data type

Possible Solution:

Verify that the data type is valid for this address.

Could not validate address because profile invocation failed. | Address = '<address>'**Error Type:**

Error

Possible Cause:

The service is not running or processing the profile timed out.

Possible Solution:

Ensure that the service is enabled and that the profile execution time is not excessive.

State machine unknown (required script function GetDriverInfo is missing or returned a bad result). | Channel name '<name>'.**Error Type:**

Error

Possible Cause:

1. The profile is missing required script function GetDriverInfo.
2. Required script function GetDriverInfo returned a bad result.

Possible Solution:

Correct the syntax errors or implement the missing functions.

Could not build a send frame because the script execution failed. | Address = '<address>', profile ID = '<id>'.**Error Type:**

Error

Possible Cause:

The profile or script is invalid or took too long to process.

Possible Solution:

Ensure that the profile is valid, and that the script execution time is not excessive.

Failed to process received frame. | Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'.

Error Type:

Error

Possible Cause:

The ParseMessage script function failed to execute.

Possible Solution:

Re-examine the runtime logic of this function.

Failed to generate write request. | Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'.

Error Type:

Error

Possible Cause:

The ProcessTags script function failed to execute.

Possible Solution:

Re-examine the runtime logic of this function.

Failed to process write response. | Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'.

Error Type:

Error

Possible Cause:

The ParseMessage script function failed to execute.

Possible Solution:

Re-examine the runtime logic of this function.

ValidateAddress() returned an invalid result. | Result = '<result value>'.

Error Type:

Error

Possible Cause:

The profile script returned an incorrect value from the ValidateAddress() function.

Possible Solution:

Ensure that ValidateAddress() returns a JavaScript object with the required fields.

Could not validate address because state machine version is unknown. | Address = '<address>'.

Error Type:

Error

Possible Cause:

1. The profile is missing required script function GetDriverInfo.
2. Required script function GetDriverInfo returned a bad result.

Possible Solution:

Correct the syntax errors or implement the missing functions.

Failed to process response. | Type = '<read/write>', Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'.

Error Type:

Error

Possible Cause:

The buildWriteRequest script function failed to execute.

Possible Solution:

Re-examine the runtime logic of this function.

Script function failed. | Channel = '<channel name>'

Error Type:

Error

Possible Cause:

The script function returned a status of Failure.

Possible Solution:

Consult profile script to determine possible sources for a Failure return status.

Send data failed. | Channel = '<channel name>'

Error Type:

Error

Possible Cause:

The data could not be sent because the socket connection failed.

Possible Solution:

Ensure that the port is not blocked by another process or application.

Receive data failed. | Channel = '<channel name>'

Error Type:

Error

Possible Cause:

Timeout exceeded while waiting for data to be received.

Possible Solution:

1. Ensure that the device is operating correctly.
2. Check that the timeout configuration is not set too short.

3. Ensure that the profile script does not return status Receive in a case where the device will not be sending any data.

Script execution failed. | Channel = '<channel name>'

Error Type:

Error

Possible Cause:

Failure occurred while invoking script function.

Possible Solution:

Consult event log for a more detailed explanation of script invocation failure.

Script result invalid. | Profile = '<profile name>', Reason = '<reason string>'

Error Type:

Error

Possible Cause:

1. Return type was invalid
2. Returned object was missing fields
3. Field types were invalid
4. Field is an enumeration, but the value does not match the enumerated type
5. Number of tag objects returned does not match the number of tag objects provided
6. Failed to parse JSON
7. Unsupported or invalid state machine version
8. Unexpected tag address
9. Failed to convert value for tag
10. Provided value for tag is out of range
11. Invalid array syntax
12. Multi-dimension arrays not supported
13. Array size is out of range
14. Arrays not supported for provided element values
15. Array size provided but data type isn't an array
16. Data type is an array but no array size provided

Possible Solution:

Ensure that the script returns a JavaScript object with the required fields.

Script invocation failed. | Channel = '<name>', reason = '<reason string>'

Error Type:

Error

Possible Cause:

1. The script engine service is unavailable.
2. The onProfileLoad function is missing or returned an invalid result.
3. Failure occurred while invoking a script function.

Possible Solution:

1. Enable the service or reassign the port using the system administration utility.
2. Check that the profile has a properly formatted onProfileLoad function.
3. Check that the profile has required functions properly formatted.

Address validation failed. | Channel = '<name>', address = '<tag address>', reason = '<reason string>'.

Error Type:

Error

Possible Cause:

1. The script engine service is unavailable.
2. The onProfileLoad function is missing or returned an invalid result.

Possible Solution:

1. Enable the service or reassign the port using the system administration utility.
2. Check that the profile has a properly formatted onProfileLoad function.

Script mode does not match the configured channel profile mode. | Channel = '<channel name>'.

Error Type:

Error

Possible Cause:

The mode returned with onProfileLoad does not match the Channel Profile Mode property.

Possible Solution:

Verify the script used in the channel with the channel property. Select Ethernet for Client and Server script modes, File otherwise.

Profile Inputs failed validation. | Channel = '<name>'.

Error Type:

Error

Possible Cause:

1. The onProfileInputsChange function in the profile returned false due to a Profile Inputs property validation failure.
2. The configured device Profile Inputs properties are not formatted according to the profile expectations.

Possible Solution:

1. Check that the configured device Profile Inputs properties are formatted as expected in the onProfileInputsChange function.
2. Check that the onProfileInputsChange function in the profile is validating the properties properly.

Profile link established. | Channel = '<name>', profile = '<name>'.

Error Type:

Informational

Profile link established. | Channel = '<name>', profile = '<name>', state machine version = '<version>'.

Error Type:

Informational

Script engine error detected. Temporarily suspending communication. | Channel = '<name>'.

Error Type:

Informational

Possible Cause:

1. The script service is disabled.
2. One of the script functions is taking too long to process.
3. The script has a syntax or runtime error.
4. The script function returned status Complete when data was incomplete.
5. Data from the device is invalid.

Possible Solution:

1. Ensure that the script engine service is enabled.
2. Check the script for infinite loops, or long running code.
3. Check the script for syntax errors.
4. Ensure the script function has received all data before returning status Complete.
5. Validate the data from the device.

Profile link established. | Channel = '<name>', profile = '<name>', state machine version = '<version>', mode = '<operating mode>'.

Error Type:

Informational

Connection limit reached. | Channel = '<name>'.

Error Type:

Informational

Sample Profiles

Example profiles for the Universal Device Driver can be added during installation of the product:
<install directory>\Examples\Universal Device Sample Profiles

Other samples are available online:

[Sample Profiles](#)

Index

A

Address validation failed. | Channel = '<name>', address = '<tag address>', reason = '<reason string>'. 23
API 5
Architecture 5
Attempts Before Timeout 13
Auto-Demotion 14

C

Channel Assignment 9
Channel Properties – Advanced 7
Channel Properties – Ethernet Communications 6
Channel Properties – General 6
Channel Properties – Write Optimizations 7
Channel Properties Profile 8
Communication Transport 11
Communications Timeouts 13
Configuration API Service 5, 9
Connect Timeout 13
Connection limit reached. | Channel = '<name>'. 24
Connection to the service failed. | Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'. 18
Cool-off State 5
Could not build a send frame because the script execution failed. | Address = '<address>', profile ID = '<id>'. 19
Could not validate address because profile invocation failed. | Address = '<address>'. 19
Could not validate address because state machine version is unknown. | Address = '<address>'. 20

D

Data Collection 10
Data Types Descriptions 17
Demote on Failure 14
Demotion Period 14
device ID 9
Device Properties – Auto-Demotion 14
Device Properties – General 9
Device Properties – Redundancy 16
Device Properties – Timing 13
Device Properties – User-Defined Settings 15
Diagnostics 6

Discard Requests when Demoted 14
Do Not Scan, Demand Poll Only 10
Driver 9
Duty Cycle 7

E

Ethernet 4
Ethernet Settings 7
Event Log Messages 18

F

Failed to convert value from script to '<data type>' for address '<address>'. 19
Failed to generate write request. | Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'. 20
Failed to process received frame. | Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'. 20
Failed to process response. | Type = '<read/write>', Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'. 21
Failed to process write response. | Channel = '<name>', host = '<host>', port = '<port>', service name = '<name>'. 20
Failed to start service. | Channel name = '<name>', service name = '<name>'. 18

G

General 9
General Operation 5
GUID 8

I

ID 8
Identification 6, 9
Initial Updates from Cache 11
Invalid profile link. | Channel = '<channel name>', Profile ID = '<id>'. 18

J

JavaScript engine 5

M

Mode 8

N

Name 9

Network Adapter 7

Non-Normalized Float Handling 8

O

onValidateTag 8

Operating Mode 9

Optimization Method 7

Overview 4

P

Profile 5, 8

Profile Inputs failed validation. | Channel = '<name>'. 23

Profile link established. | Channel = '<name>', profile = '<name>', state machine version = '<version>', mode = '<operating mode>'. 24

Profile link established. | Channel = '<name>', profile = '<name>', state machine version = '<version>'. 24

Profile link established. | Channel = '<name>', profile = '<name>'. 24

Profile registration failed because the profile syntax is bad or is missing required functions. | Channel name '<name>'. 18

R

Receive data failed. | Channel = '<channel name>'. 21

Redundancy 16

Replace with Zero 8

Request Timeout 13

Respect Tag-Specified Scan Rate 11

S

Scan Mode 10

Script engine error detected. Temporarily suspending communication. | Channel = '<name>'. 24

Script execution failed. | Channel = '<channel name>'. 22

Script failure. | Result = '<result value>'. 19

Script function failed. | Channel = '<channel name>'. 21
Script invocation failed. | Channel = '<name>', reason = '<reason string>'. 22
Script mode does not match the configured channel profile mode. | Channel = '<channel name>'. 23
Script result invalid. | Profile = '<profile name>', Reason = '<reason string>'. 22
Send data failed. | Channel = '<channel name>'. 21
Service 5
Setup 9
Simulated 10
State machine unknown (required script function GetDriverInfo is missing or returned a bad result). | Channel name '<name>'. 19

T

Tag Counts 6, 10
Telnet 4
Timeouts to Demote 14
Timing 13

U

Unmodified 8
User-Defined Settings 15
user interface 5
UUID 4

V

V8 JavaScript 5
ValidateAddress() returned an invalid result. | Result = '<result value>'. 20

W

Write All Values for All Tags 7
Write Only Latest Value for All Tags 7
Write Only Latest Value for Non-Boolean Tags 7