Telemecanique Uni-Telway Driver

Table of Contents

Telemecanique Uni-Telway Driver

Help version 1.020

CONTENTS

Overview

The Telemecanique Uni-Telway Driver provides a reliable way to connect Telemecanique Uni-Telway devices to OPC client applications; including HMI, SCADA, Historian, MES, ERP, and countless custom applications. It is intended for use with Telemecanique devices that communicate using the Uni-Telway protocol.

Setup

Supported Devices

Large Frame and Small Frame. Telemecanique Devices that do not support the full 128-byte data field provided by the Telemecanique Uni-Telway Slave protocol are considered Small Frame. One example of a Small Frame device is the TSX17-20.

Communication Protocol

Uni-Telway

Supported Communication Properties

Baud Rate: 1200, 2400, 9600, 19200 or 38400
Parity: None, Even or Odd
Data Bits: 5, 6, 7 or 8
Stop Bits: 1 or 2

Note: Due to the nature of the Uni-Telway protocol, users may be required to disable FIFO buffer usage on the COM port(s) that have been selected for use with this driver.

Converters

Depending on type of PLC being used, the TSX SCA 72 converter may not work. The TSX SCA 72 requires that the PC control the switching of the RS-485 transmitter/receiver on and off. This is done using the RTS line from the serial port. PLCs like the TSX37 respond quickly to the Telemecanique Uni-Telway Driver: the Windows communication driver is not fast enough to effectively control the TSX SCA 72s RS-485 line converter.

Schneider Automation (Electric) makes a modem cable that should work. The cable was designed for devices that do not manage the RTS signal. The part # is:TSX PCX 1031.

Note: Converters are to be used with our 32 bit Telemecanique Uni-Telway Driver.

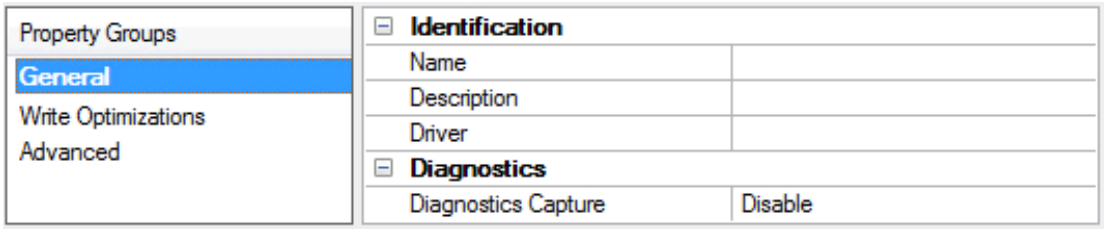
Flow Control

When using an RS232/RS485 converter, the type of flow control that is required will depend upon the needs of the converter. Some converters do not require any flow control and others will require RTS flow.To determine the converter's flow requirements, refer to its documentation. We recommend using an RS485 converted that provides automatic flow control.

Note: When using the manufacturer's supplied communications cable, it is sometimes necessary to choose a flow control property of **RTS** or **RTS Always** under the Channel Properties.

Channel Properties — General

This server supports the use of simultaneous multiple communications drivers. Each protocol or driver used in a server project is called a channel. A server project may consist of many channels with the same communications driver or with unique communications drivers. A channel acts as the basic building block of an OPC link. This group is used to specify general channel properties, such as the identification attributes and operating mode.



Identification

Name: User-defined identity of this channel. In each server project, each channel name must be unique. Although names can be up to 256 characters, some client applications have a limited display window when browsing the OPC server's tag space. The channel name is part of the OPC browser information.
For information on reserved characters, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in the server help.

Description: User-defined information about this channel.
Many of these properties, including Description, have an associated system tag.

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

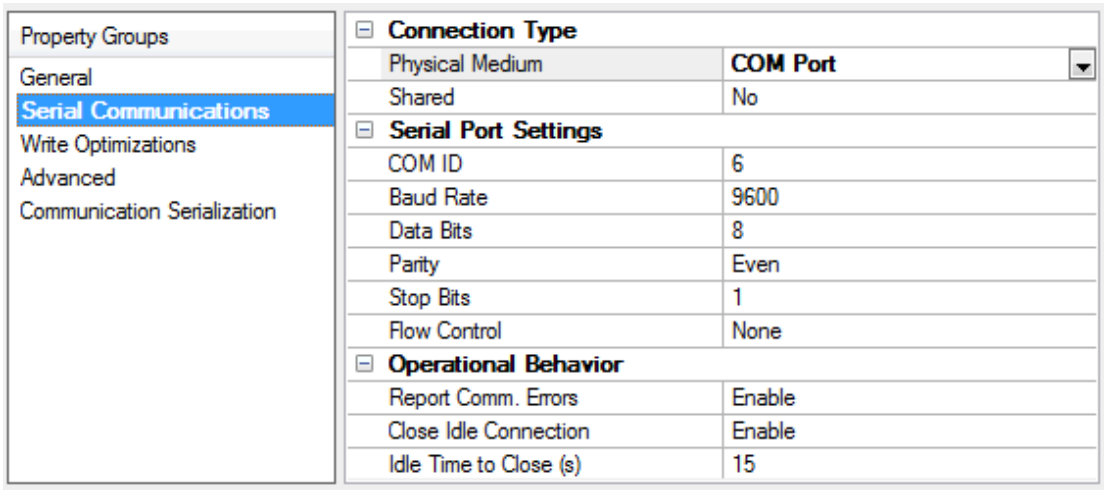
Diagnostics

Diagnostics Capture: When enabled, this option makes the channel's diagnostic information available to OPC applications. 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 disabled if the driver does not support diagnostics.
For more information, refer to "Communication Diagnostics" in the server help.

Channel Properties — Serial Communications

Serial communication properties are available to serial drivers and vary depending on the driver, connection type, and options selected. Below is a superset of the possible properties.
Click to jump to one of the sections: **Connection Type**, **Serial Port Settings** or **Ethernet Settings**, and **Operational Behavior**.

Note: With the server's online full-time operation, these properties can be changed at any time. Utilize the User Manager to restrict access rights to server features, as changes made to these properties can temporarily disrupt communications.



Connection Type

Physical Medium: Choose the type of hardware device for data communications. Options include COM Port, None, Modem, and Ethernet Encapsulation. The default is COM Port.

- **None**: Select None to indicate there is no physical connection, which displays the **Operation with no Communications** section.
- **COM Port**: Select Com Port to display and configure the **Serial Port Settings** section.
- **Modem**: Select Modem if phone lines are used for communications, which are configured in the **Modem Settings** section.
- **Ethernet Encap.**: Select if Ethernet Encapsulation is used for communications, which displays the **Ethernet Settings** section.
- **Shared**: Verify the connection is correctly identified as sharing the current configuration with another channel. This is a read-only property.

Serial Port Settings

COM ID: Specify the Communications ID to be used when communicating with devices assigned to the channel. The valid range is 1 to 9991 to 16. The default is 1.

Baud Rate: Specify the baud rate to be used to configure the selected communications port.

Data Bits: Specify the number of data bits per data word. Options include 5, 6, 7, or 8.

Parity: Specify the type of parity for the data. Options include Odd, Even, or None.

Stop Bits: Specify the number of stop bits per data word. Options include 1 or 2.

Flow Control: Select how the RTS and DTR control lines are utilized. Flow control is required to communicate with some serial devices. Options are:

- **None**: This option does not toggle or assert control lines.
- **DTR**: This option asserts the DTR line when the communications port is opened and remains on.
- **RTS**: This option specifies that the RTS line is high if bytes are available for transmission. After all buffered bytes have been sent, the RTS line is low. This is normally used with RS232/RS485 converter hardware.
- **RTS, DTR**: This option is a combination of DTR and RTS.
- **RTS Always**: This option asserts the RTS line when the communication port is opened and remains on.
- **RTS Manual**: This option asserts the RTS line based on the timing properties entered for RTS Line Control. It is only available when the driver supports manual RTS line control (or when the properties are shared and at least one of the channels belongs to a driver that provides this support).
  RTS Manual adds an **RTS Line Control** property with options as follows:
  - **Raise**: This property specifies the amount of time that the RTS line is raised prior to data transmission. The valid range is 0 to 9999 milliseconds. The default is 10 milliseconds.
  - **Drop**: This property specifies the amount of time that the RTS line remains high after data transmission. The valid range is 0 to 9999 milliseconds. The default is 10 milliseconds.
  - **Poll Delay**: This property specifies the amount of time that polling for communications is delayed. The valid range is 0 to 9999. The default is 10 milliseconds.

Tip: When using two-wire RS-485, "echoes" may occur on the communication lines. Since this communication does not support echo suppression, it is recommended that echoes be disabled or a RS-485 converter be used.

Operational Behavior

- **Report Comm. Errors**: Enable or disable reporting of low-level communications errors. When enabled, low-level errors are posted to the Event Log as they occur. When disabled, these same errors are not posted even though normal request failures are. The default is Enable.
- **Close Idle Connection**: Choose to close the connection when there are no longer any tags being referenced by a client on the channel. The default is Enable.
- **Idle Time to Close**: Specify the amount of time that the server waits once all tags have been removed before closing the COM port. The default is 15 seconds.

Ethernet Settings

Note: Not all serial drivers support Ethernet Encapsulation. If this group does not appear, the functionality is not supported.

Ethernet Encapsulation provides communication with serial devices connected to terminal servers on the Ethernet network. A terminal server is essentially a virtual serial port that converts TCP/IP messages on the Ethernet network to serial data. Once the message has been converted, users can connect standard devices that support serial communications to the terminal server. The terminal server's serial port must be properly configured to match the requirements of the serial device to which it is attached. *For more information, refer to "How To... Use Ethernet Encapsulation" in the server help.*

- **Network Adapter**: Indicate a network adapter to bind for Ethernet devices in this channel. Choose a network adapter to bind to or allow the OS to select the default.
  Specific drivers may display additional Ethernet Encapsulation properties. For more information, refer to Channel Properties — Ethernet Encapsulation.

Modem Settings

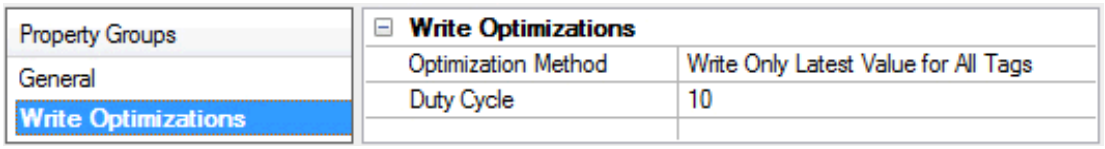
- **Modem**: Specify the installed modem to be used for communications.
- **Connect Timeout**: Specify the amount of time to wait for connections to be established before failing a read or write. The default is 60 seconds.
- **Modem Properties**: Configure the modem hardware. When clicked, it opens vendor-specific modem properties.
- **Auto-Dial**: Enables the automatic dialing of entries in the Phonebook. The default is Disable. *For more information, refer to "Modem Auto-Dial" in the server help.*
- **Report Comm. Errors**: Enable or disable reporting of low-level communications errors. When enabled, low-level errors are posted to the Event Log as they occur. When disabled, these same errors are not posted even though normal request failures are. The default is Enable.
- **Close Idle Connection**: Choose to close the modem connection when there are no longer any tags being referenced by a client on the channel. The default is Enable.
- **Idle Time to Close**: Specify the amount of time that the server waits once all tags have been removed before closing the modem connection. The default is 15 seconds.

Operation with no Communications

- **Read Processing**: Select the action to be taken when an explicit device read is requested. Options include Ignore and Fail. Ignore does nothing; Fail provides the client with an update that indicates failure. The default setting is Ignore.

Channel Properties — Write Optimizations

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



Write Optimizations

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

- **Write All Values for All Tags**: This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.
- **Write Only Latest Value for Non-Boolean Tags**: Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are

needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.

Note: This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.

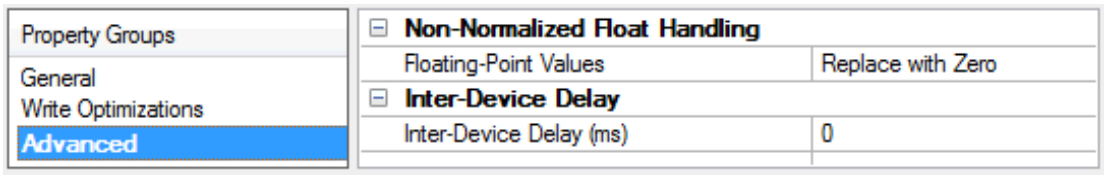
- **Write Only Latest Value for All Tags**: This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write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

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

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

Channel Properties — Advanced

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



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

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

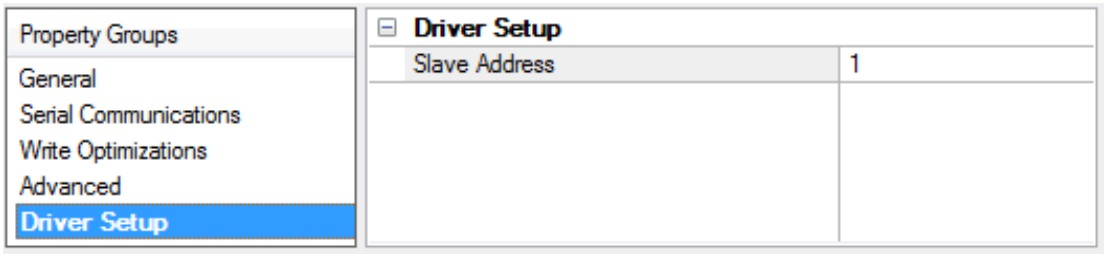
Note: This property is disabled if the driver does not support floating point values or if it only supports the option that is displayed. According to the channel's float normalization setting, only real-time driver tags (such as values and arrays) are subject to float normalization. For example, EFM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

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

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

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

Channel Properties - Driver Setup



Slave Address: If this driver is seen as a slave on the Uni-Telway network; it must respond to the master as it is polled. The Slave Address determines which poll request the driver responds to in the programmed sequence. Valid range is from 1-253.

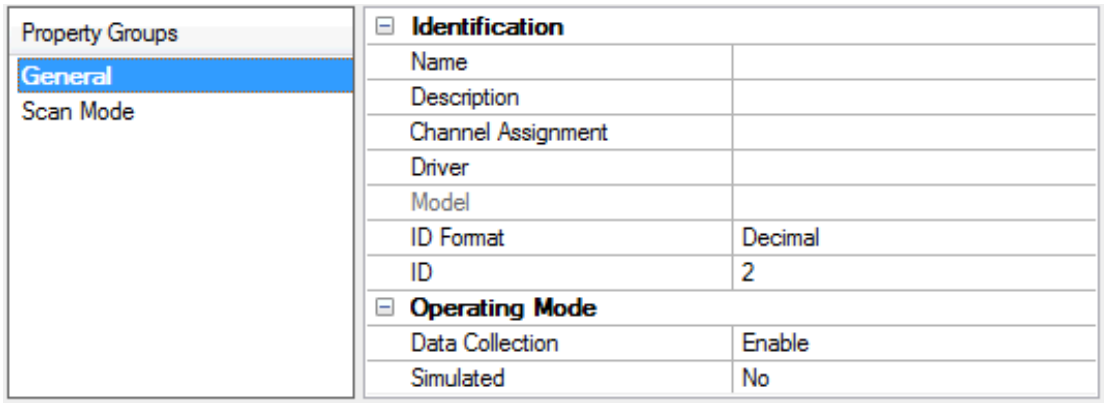
Tips:

1. Make sure that the assigned Slave ID is valid. Since the driver is seen as a slave on the Uni-Telway network, make sure that the PLC (master) is polling the range that the Slave falls into. For example, if the master is polling devices 1-31, then a Slave ID of 32 would not be valid.
2. Make sure that the 5-layer Device ID is correct. When trying to communicate with one Master PLC, the Device ID for the master should be as follows. Addresses are in hex.

   Network - 0
   Station - FE
   Gate - 0
   Module - 0
   Channel - 0
3. Select either Sm_Frame or Lg_Frame depending on the Telemecanique PLC model with which communication is being established. For the Telemecanique PLC model TSX17, select the Sm_ Frame option. For all other Telemecanique PLC models, select the Lg_Frame option.
4. Because an RS232/485 converter (which is capable of automatic direction control) will be used, we recommend that the Device Setup section of the Telemecanique Uni-Telway Slave help file be consulted. The specs of the converter we use may be faxed.
5. In some Telemecanique PLCs, the **Wait Time** can be configured. The Wait Time is the amount of time the PLC acting as the Uni-Telway master will wait when it issues poll messages to a slave device before timing out on the slave. The default in most cases is 10 milliseconds. If this time is increased to 20 milliseconds or more, the driver's response time will dramatically improve due to the reduced number of missed poll messages. This property is available in the PLC configuration.

Device Properties — General

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



Identification

Name: This property specifies the name of the device. It is a logical user-defined name that can be up to 256 characters long, and may be used on multiple channels.
Note: Although descriptive names are generally a good idea, some OPC client applications may have a limited display window when browsing the OPC server's tag space. The device name and channel name become part of the browse tree information as well. Within an OPC client, the combination of channel name and device name would appear as "ChannelName.DeviceName".
For more information, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in server help.

Description: User-defined information about this device.
Many of these properties, including Description, have an associated system tag.

Channel Assignment: User-defined name of the channel to which this device currently belongs.

Driver: Selected protocol driver for this device. This property specifies the driver selected during channel creation. It is disabled in the channel properties.

Model: This property specifies the specific type of device that is associated with this ID. The contents of the drop-down menu depends on the type of communications driver being used. Models that are not supported by a driver are disabled. If the communications driver supports multiple device models, the model selection can only be changed when there are no client applications connected to the device.

Note: If the communication driver supports multiple models, users should try to match the model selection to the physical device. If the device is not represented in the drop-down menu, select a model that conforms closest to the target device. Some drivers support a model selection called "Open," which allows users to communicate without knowing the specific details of the target device. For more information, refer to the driver help documentation.

ID: This property specifies the device's station / node / identity / address. The type of ID entered depends on the communications driver being used. For many 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 ID format can be Decimal, Octal, and Hexadecimal. If the driver is Ethernet-based or supports an unconventional station or node name, the device's TCP/IP address may be used as the device ID. TCP/IP addresses consist of four values that are separated by periods, with each value in the range of 0 to 255. Some device IDs are string based. There may be additional properties to configure within the ID field, depending on the driver.

Operating Mode

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

Simulated: This option places the device into Simulation Mode. In this mode, the driver does not attempt to communicate with the physical device, but the server continues to return valid OPC data. Simulated stops physical communications with the device, but allows OPC data to be returned to the OPC client as valid data. While in Simulation Mode, the server treats all device data as reflective: whatever is written to the simulated device is read back and each OPC item is treated individually. The item's memory map is based on the group Update Rate. The data is not saved if the server removes the item (such as when the server is reinitialized). The default is No.

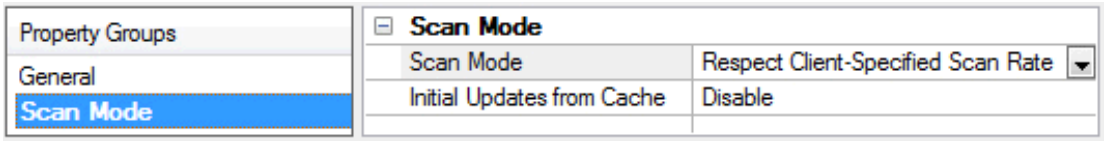
Notes:

1. This System tag (_Simulated) is read only and cannot be written to for runtime protection. The System tag allows this property to be monitored from the client.
2. In Simulation mode, the item's memory map is based on client update rate(s) (Group Update Rate for OPC clients or Scan Rate for native and DDE interfaces). This means that two clients that reference the same item with different update rates return different data.

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

Device Properties — Scan Mode

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



Scan Mode: specifies how tags in the device are scanned for updates sent to subscribed 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 maximum scan rate to be used. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
  Note: When the server has an active client and items for the device and the scan rate value is increased, the changes take effect immediately. When the scan rate value is decreased, the changes do not take effect until all client applications have been disconnected.
- **Request All Data at Scan Rate**: This mode forces tags to be scanned at the specified rate for subscribed clients. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
- **Do Not Scan, Demand Poll Only**: This mode does not periodically poll tags that belong to the device nor perform a read to get an item's initial value once it becomes active. It is the client's

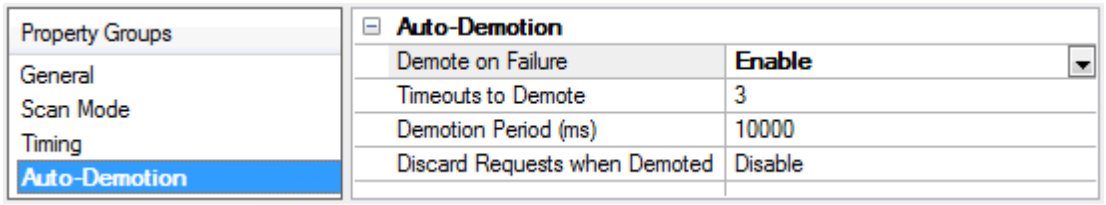
responsibility to poll for updates, either by writing to the _DemandPoll tag or by issuing explicit device reads for individual items. *For more information, refer to "Device Demand Poll" in server help*.

- **Respect Tag-Specified Scan Rate**: This mode forces static tags to be scanned at the rate specified in their static configuration tag properties. Dynamic tags are scanned at the client-specified scan rate.

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

Device Properties — 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.



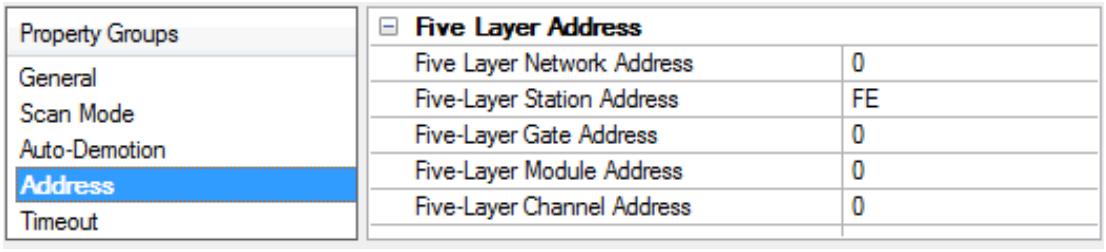
Demote on Failure: When enabled, the device is automatically taken off-scan until it is responding again.
Tip: Determine when a device is off-scan by monitoring its demoted state using the _AutoDemoted system tag.

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 - Address

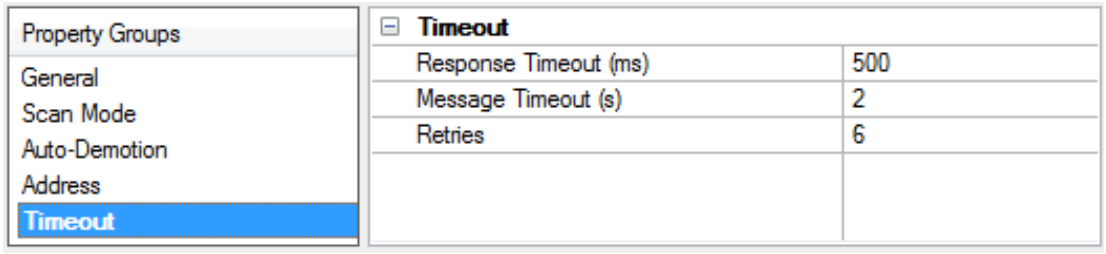


Five-Layer Address

This property group specifies the five-layer address information to identify the device connected to the Telemecanique network with the following components:

Network 00 - FE (0 - 254)
Station 00 - FE (0 - 254)
Gate 00 - FE (0 - 254)
Module Rack 0 - F (0 - 15) / Module 0 - 7
Channel 00 - FE (0 - 254)

Device Properties - Timeout

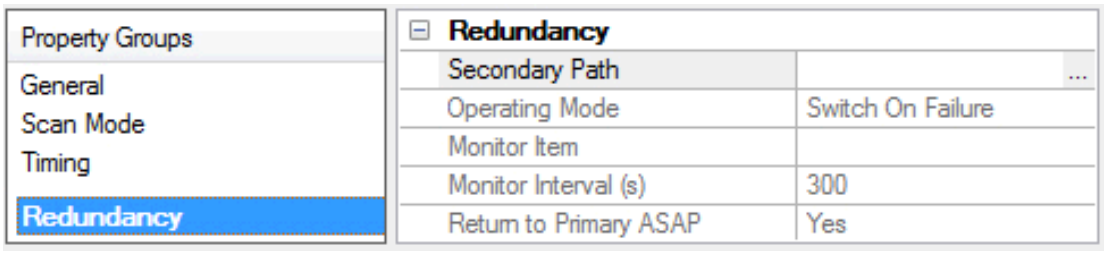


Response Timeout: Specify how long the driver should wait for intermediate responses from the master before timing out. An intermediate response is any single portion of a response frame or poll request from the master. Valid response timeout values range from 20 to 3000 ms. The default is 500 ms.

Message Timeout: Specify how long the driver should wait for the master to satisfy a request for data. Valid message timeout values range from 1 to 999 seconds. The default is 2 seconds.

Retries: Specify how many times the driver attempt a communications request before considering the request to have failed. The valid range is 1 to 10 retries. The default is 6 retries. The number of retries configured for the application depends largely on the communications environment.

Device Properties — Redundancy



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

- *Consult the website, a sales representative, or the user manual for more information.*

Data Types Description

Data Type	Description
Boolean	Single bit
Word	Unsigned 16-bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16-bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32-bit value bit 0 is the least significant bit bit 31 the most significant bit
Long	Signed 32-bit value bit 0 is the least significant bit bit 30 the most significant bit bit 31 is the sign bit

Address Descriptions

Select a model type below to see address information for PLCs of that type. Telemecanique Devices that do not support the full 128-byte data field provided by the Telemecanique Uni-Telway Slave protocol are considered Small Frame. One example of a Small Frame device is the TSX17-20.

Small Frame
Large Frame

Small Frame Addresses

The default data types for dynamically defined tags are shown in **bold** .

Device Type	Range	Data Type	Access
Internal Word Memory	MW00000-MW32767* MWxxxxx.00 - MWxxxxx.15 MWxxxxx.L L = Number of words used for String Range: 1-15	Short, **Word**, Long, DWord Boolean String	Read/Write Read Only Read/Write
Internal Bit Memory	M0000-M4095	Boolean	Read/Write
Constant Memory	KW00000-KW32767* KWxxxxx.00-KWxxxxx.15	Short, **Word**, Long, DWord Boolean	Read Only Read Only
Timer Preset	T000.P-T127.P	Short, **Word**	Read/Write
Timer Value	T000.V-T127.V	Short, **Word**	Read Only
Timer Done Bit	T000.Q-T127.Q	Boolean	Read Only
Counter Preset	C000.P-C255.P	Short, **Word**	Read/Write
Counter Value	C000.V-C255.V	Short, **Word**	Read Only
Counter Done Bit	C000.D-C255.D	Boolean	Read Only
PL7 Timer Preset	TP000.P-TP127.P	Short, **Word**	Read/Write
PL7 Timer Value	TP000.V-TP127.V	Short, **Word**	Read Only
PL7 Timer Done Bit	TP000.Q-TP127.Q	Boolean	Read Only
IEC Timer Preset	TM000.P-TM127.P	Short, **Word**	Read/Write
IEC Timer Value	TM000.V-TM127.V	Short, **Word**	Read Only
IEC Timer Done Bit	TM000.Q-TM127.Q	Boolean	Read Only
PL7 Counter Preset	CP000.P-CP255.P	Short, **Word**	Read/Write
PL7 Counter Value	CP000.V-CP255.V	Short, **Word**	Read Only
PL7 Counter Done Bit	CP000.D-CP255.D	Boolean	Read Only

*Supports array notation. To specify an array, append the array size to the address specification as follows: address[array size] or address[rows][cols]. Small Frame arrays can have a maximum size of 15 word elements.

Notes:

1. Use PL7/IEC Timers (TP, TM Device Types) and PL7 Counter (CP Device Type) with TSX Micro or Premium PLCs. These types are not supported when using TSX Nano PLC.
2. Maximum value for TP, TM and CP Device Types Preset is 9999. A write of a value greater than Maximum will fail with an error '06' (invalid write value).
3. Timer and Counter types 'T' and 'C' are not supported when using TSX Nano, Micro or Premium PLCs and will fail with a "Negative Response Returned" error message.

Large Frame Addresses

The default data types for dynamically defined tags are shown in **bold**.

Device Type	Range	Data Type	Access
Internal Word Memory	MW00000-MW32767* MWxxxxx.00 - MWxxxxx.15 MWxxxxx.L L = Number of words used for String Range: 1-60	Short, **Word**, Long, DWord Boolean String	Read/Write Read Only Read/Write
Internal Bit Memory	M0000-M4095	Boolean	Read/Write
Constant Memory	KW00000-KW32767* KWxxxxx.00-KWxxxxx.15	Short, **Word**, Long, DWord Boolean	Read Only Read Only
Timer Preset	T000.P-T127.P	Short, **Word**	Read/Write
Timer Value	T000.V-T127.V	Short, **Word**	Read Only
Timer Done Bit	T000.Q-T127.Q	Boolean	Read Only
Counter Preset	C000.P-C255.P	Short, **Word**	Read/Write
Counter Value	C000.V-C255.V	Short, **Word**	Read Only
Counter Done Bit	C000.D-C255.D	Boolean	Read Only
PL7 Timer Preset	TP000.P-TP127.P	Short, **Word**	Read/Write
PL7 Timer Value	TP000.V-TP127.V	Short, **Word**	Read Only
PL7 Timer Done Bit	TP000.Q-TP127.Q	Boolean	Read Only
IEC Timer Preset	TM000.P-TM127.P	Short, **Word**	Read/Write
IEC Timer Value	TM000.V-TM127.V	Short, **Word**	Read Only
IEC Timer Done Bit	TM000.Q-TM127.Q	Boolean	Read Only
PL7 Counter Preset	CP000.P-CP255.P	Short, **Word**	Read/Write
PL7 Counter Value	CP000.V-CP255.V	Short, **Word**	Read Only
PL7 Counter Done Bit	CP000.D-CP255.D	Boolean	Read Only

*Supports array notation. To specify an array, append the array size to the address specification as follows: address [array size] or address[rows][cols]. Large Frame arrays can have a maximum size of 60 word elements.

Notes:

1. Use PL7/IEC Timers (TP, TM Device Types) and PL7 Counter (CP Device Type) with TSX Micro or Premium PLCs. These types are not supported when using TSX Nano PLC.
2. Maximum value for TP, TM and CP Device Types Preset is 9999. A write of a value greater than Maximum will fail with an error '06' (invalid write value).
3. Timer and Counter types 'T' and 'C' are not supported when using TSX Nano, Micro or Premium PLCs and will fail with a "Negative Response Returned" error message.

Error Descriptions

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

Address Validation

Missing address
Device address <address> contains a syntax error
Address <address> is out of range for the specified device or register
Data type <type> is not valid for device address <address>
Device address <address> is read only

Serial Communications

COMn does not exist
Error opening COMn
COMn is in use by another application
Unable to set comm properties on COMn
Communications error on <channel name> [<error mask>]

Device Status Messages

Device <device name> is not responding
Unable to write to <address> on device <device name>
Device <device name> responded with error <error byte> (Tag <address>, Size <size>)
Device <device name> negative response returned (Tag <address>, Size <size>)
Device <device name> response invalid (Tag <address>, Size <size>)

Missing address

Error Type:

Warning

Possible Cause:

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

Solution:

Re-enter the address in the client application.

Device address <address> contains a syntax error

Error Type:

Warning

Possible Cause:

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

Solution:

Re-enter the address in the client application.

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

Error Type:
Warning

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

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

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

Error Type:
Warning

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

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

Device address <address> is read only

Error Type:
Warning

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

Solution:
Change the access mode in the client application.

COMn does not exist

Error Type:
Fatal

Possible Cause:
The specified COM port is not present on the target computer.

Solution:
Verify that the proper COM port has been selected.

Error opening COMn

Error Type:
Fatal

Possible Cause:
The specified COM port could not be opened due to an internal hardware or software problem on the target computer.

Solution:
Verify that the COM port is functional and may be accessed by other Windows applications.

COMn is in use by another application

Error Type:
Fatal

Possible Cause:
The serial port assigned to a device is being used by another application.

Solution:
Verify that the correct port has been assigned to the channel.

Unable to set comm properties on COMn

Error Type:
Fatal

Possible Cause:
The serial properties for the specified COM port are not valid.

Solution:
Verify the serial properties and make any necessary changes.

Communications error on <channel name> [<error mask>]

Error Type:
Serious

Error Mask Definitions:
B = Hardware break detected.
F = Framing error.
E = I/O error.
O = Character buffer overrun.
R = RX buffer overrun.
P = Received byte parity error.
T = TX buffer full.

Possible Cause:

1. The serial connection between the device and the host PC is bad.
2. The communications properties for the serial connection are incorrect.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications properties match those of the device.

Device <device name> is not responding

Error Type:
Serious

Possible Cause:

1. The serial connection between the device and the host PC is broken.
2. The communications properties for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.
4. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device property.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications properties match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.
4. Increase the Request Timeout property so that the entire response can be handled.

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

Error Type:
Serious

Possible Cause:

1. The serial connection between the device and the host PC is broken.
2. The communications properties for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications properties match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.

Device <device name> responded with error <error byte> (Tag <address>, Size <size>)

Error Type:
Warning

Possible Cause:
See the Solution.

Solution:
Use following tables to decode the error byte.

Read Error Byte Possible Values

Error Byte	Description
00h	Read performed
02h	Instance number outside configuration limits
03h	Number of bits to read not a multiple of 8
04h	Object not configured in PL7 application
05h	Response buffer overflow
07h	Access property, object label or element number incorrect
08h	Access temporarily not possible
09h	Number of reads requested equal to zero

Write Error Byte Possible Values

Error Byte	Description
00h	Write operation performed
01h	Element write-protected
02h	Instance number outside configuration limits
04h	Object not configured in PL7 application
06h	Invalid write value
08h	Access temporarily not possible
09h	Number of write operations requested equal to zero

Device <device name> Negative response returned (Tag <address>, Size <size>)

Error Type:
Serious

Possible Cause:
An unsupported type was used.

Solution:
Use a supported type.

Device <device name> response invalid (Tag <address>, Size <size>)

Error Type:
Warning

Possible Cause:
Either an unsupported type was used or the PLC does not support the type.

Solution:
Use a supported type.

Resources

In addition to this user manual, there are a variety of resources available to assist customers, answer questions, provide more detail about specific implementations, or help with troubleshooting specific issues.

Knowledge Base
Whitepapers
Connectivity Guides
Technical Notes
Training Programs
Training Videos
Kepware Technical Support
PTC Technical Support

Index

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W