OPC DA Client Driver

Table of Contents

OPC DA Client Driver

CONTENTS

Overview

What is the OPC DA Client Driver?

Channel Setup

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

Device Setup

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

Data Types Description

What data types does this driver support?

Address Descriptions

How do I address a data location with the OPC DA Client Driver?

Error Descriptions

What error messages does the OPC DA Client Driver produce?

Version 1.050
© 2024 PTC Inc. All Rights Reserved.

Overview

The OPC DA Client Driver provides a reliable way to connect OPC DA Client devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It can be used to consolidate data from a number of OPC servers and is made available to server clients through all supported interfaces (such as OPC DA, OPC DX, DDE, SuiteLink, NIO/PDB, and so forth).

The OPC DA Client Driver has the ability to do the following:

- Provide a single, reliable connection point for accessing data from multiple OPC servers on both local and remote machines.
- Optimize OPC server performance through OPC groupings (which may be different than those that are required or allowed by clients).
- Configure connection monitoring and reconnect behavior for each OPC server.
- Provide connectivity to remote OPC servers for clients that do not support DCOM.
- Provide connectivity to multiple OPC servers from clients that do not support multiple connections (or handle them well).
- Allow connectivity to OPC servers that use different interfaces that are supported by clients (such as DDE, SuiteLink, NIO/PDB, and so forth).

Channel and Device Limits

The maximum number of channels supported by this driver is 128. The maximum number of devices supported by this driver is 256 per channel.

Tip: The OPC DA Client Driver supports integration with the Media-Level Redundancy Plug-In.

OPC Compliance

This driver supports OPC DA 1.0, 2.0, and 3.0 connections with underlying servers. It uses 2.0 calls if supported, and automatically falls back to 1.0 if not supported. It does not support 3.0 advanced operations.

Note: By default, the driver requests data updates by exception (active groups and items). It can be configured to poll items (inactive groups and active items) using asynchronous reads. All writes are asynchronous.

Project Architecture

An OPC DA Client Driver channel represents a connection to an OPC server; an OPC DA Client Driver device represents an OPC group. The OPC client can connect to OPC servers provided by one or more vendors. The following diagram shows the OPC client and server relationship.
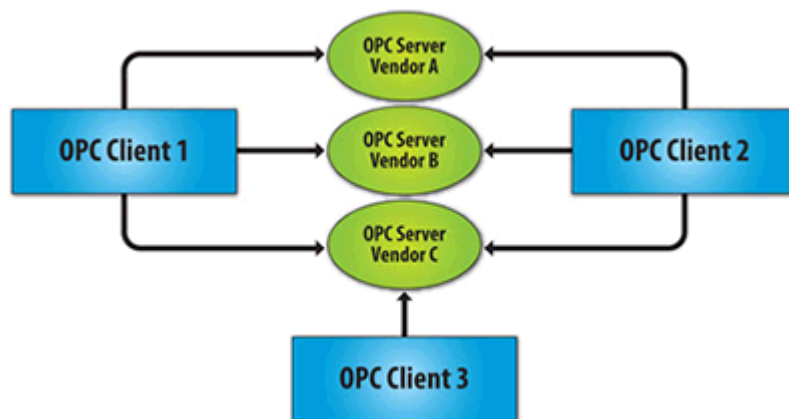
The OPC group object maintains information about itself and provides the mechanism for containing and logically organizing OPC items. The following diagram shows the group and item relationship.
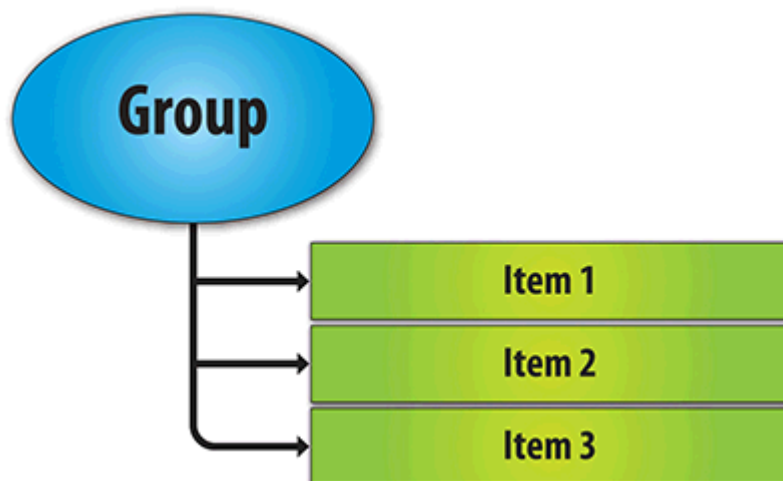
Channel Properties — General

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

Property Groups		Identification	
General		Name	
Write Optimizations		Description	
Advanced		Driver	
		Diagnostics	
		Diagnostics Capture	Disable
		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 does not support diagnostics.
For more information, refer to Communication Diagnostics in the server help.

Diagnostics

Diagnostics Capture: When enabled, this option allows the usage of statistics tags that provide feedback to client applications regarding the operation of the channel. Because the server's diagnostic features require a minimal amount of overhead processing, it is recommended that they be utilized when needed and disabled when not. The default is disabled.

⬭ **Note:** This property is not available if the driver does not support diagnostics.

✤ *For more information, refer to Statistics Tags in the server help.*

Tag Counts

Static Tags: Provides the total number of defined static tags at this level (device or channel). This information can be helpful in troubleshooting and load balancing.

Channel Properties — Write Optimizations

The server must ensure that the data written from the client application gets to the device on time. Given this goal, the server provides optimization properties to meet specific needs or improve application responsiveness.

Property Groups	⊟ **Write Optimizations**	
General	Optimization Method	Write Only Latest Value for All Tags
Write Optimizations	Duty Cycle	10

Write Optimizations

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

- **Write All Values for All Tags**: This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.

- **Write Only Latest Value for Non-Boolean Tags**: Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.
  ⬭ **Note**: This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.

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

Duty Cycle: is used to control the ratio of write to read operations. The ratio is always based on one read for every one to ten writes. The duty cycle is set to ten by default, meaning that ten writes occur for each

read operation. Although the application is performing a large number of continuous writes, it must be ensured that read data is still given time to process. A setting of one results in one read operation for every write operation. If there are no write operations to perform, reads are processed continuously. This allows optimization for applications with continuous writes versus a more balanced back and forth data flow.

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

Channel Properties — Advanced

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

Property Groups	⊟ **Non-Normalized Float Handling**	
General	Floating-Point Values	Replace with Zero
Write Optimizations	⊟ **Inter-Device Delay**	
Advanced	Inter-Device Delay (ms)	0

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

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

⬤ **Note:** This property is 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 — OPC Server

This property specifies the OPC server to which the channel connects.

Property Groups	⊟ **OPC Server**	
General	Server Browse	Select Server...
Write Optimizations	Program ID	
Advanced	Remote Machine Name	
OPC Server	Connection Type	Any
Connection		

Server Browse: Click **Select Server...** to invoke the Select OPC Server dialog to locate and select an OPC server.
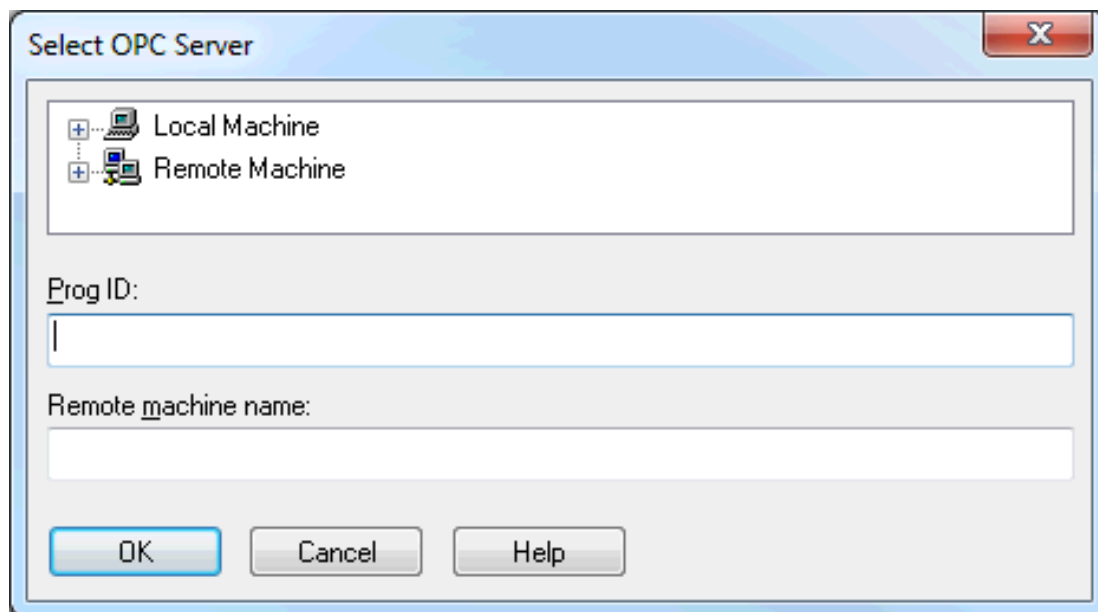
Prog ID: Specify the Program ID of the server to which the driver connects.

Remote Machine Name: Specify the name of the machine in which the server resides, as specified by the Prog ID. This field should be left blank if the server is located on the same machine as the driver.

Connection Type: Specify the type of connection that the driver should establish with the server on the local machine. Options include InProc, Local, and Any. The default is Any.
● **Note**: When the server is registered as InProc, an in process connection is attempted. Otherwise, a Local connection is attempted.

Channel Properties — Connection

This property specifies the time interval between connection retry attempts and integrity polls. Additional integrity checks may be configured at the group level. For more information, refer to **Watchdog**.

Failed Connection Retry Interval (s): Specify the time between connection attempts. The valid range is 5 to 600 seconds. The default is 5 seconds.

Server Status Query Interval (s): Specify how often the driver tests the connection to the underlying server by sending a GetStatus message. If a response is not received, the driver automatically reconnects to the server. The valid range is 5 to 30 seconds. The default is 5 seconds.

Device Properties — General

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

Property Groups	Identification	
General	Name	
Scan Mode	Description	
	Channel Assignment	
	Driver	
	Model	
	ID Format	Decimal
	ID	2

Identification

Name: Specify the name of the device. It is a logical user-defined name that can be up to 256 characters long and may be used on multiple channels.

  Note: Although descriptive names are generally a good idea, some OPC client applications may have a limited display window when browsing the OPC server's tag space. The device name and channel name become part of the browse tree information as well. Within an OPC client, the combination of channel name and device name would appear as "ChannelName.DeviceName".
For more information, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in server help.

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

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

Driver: Selected protocol driver for this device.

Model: Specify the type of device that is associated with this ID. The contents of the drop-down menu depend on the type of communications driver being used. Models that are not supported by a driver are disabled. If the communications driver supports multiple device models, the model selection can only be changed when there are no client applications connected to the device.
  Note: If the communication driver supports multiple models, users should try to match the model selection to the physical device. If the device is not represented in the drop-down menu, select a model that conforms closest to the target device. Some drivers support a model selection called "Open," which allows users to communicate without knowing the specific details of the target device. *For more information, refer to the driver documentation.*

ID: Specify the device's driver-specific station or node. The type of ID entered depends on the communications driver being used. For many communication drivers, the ID is a numeric value. Drivers that support a Numeric ID provide users with the option to enter a numeric value whose format can be changed to suit the needs of the application or the characteristics of the selected communications driver. The format is set by the driver by default. Options include Decimal, Octal, and Hexadecimal.
  Note: If the driver is Ethernet-based or supports an unconventional station or node name, the device's TCP/IP address may be used as the device ID. TCP/IP addresses consist of four values that are separated by periods, with each value in the range of 0 to 255. Some device IDs are string based. There may be additional properties to configure within the ID field, depending on the driver.

Operating Mode

Property Groups	⊞ **Identification**	
General	⊟ **Operating Mode**	
Scan Mode	Data Collection	Enable
	Simulated	No

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

Simulated: Place the device into or out of Simulation Mode. In this mode, the driver does not attempt to communicate with the physical device, but the server continues to return valid OPC data. Simulated stops physical communications with the device, but allows OPC data to be returned to the OPC client as valid data. While in Simulation Mode, the server treats all device data as reflective: whatever is written to the simulated device is read back and each OPC item is treated individually. The data is not saved if the server removes the item (such as when the server is reinitialized). The default is No.

● **Notes:**

1. Updates are not applied until clients disconnect and reconnect.

2. The 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.

3. 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.

4. When a device is simulated, updates may not appear faster than one (1) second in the client.

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

Tag Counts

Property Groups	⊟ **Identification**	
General	⊟ **Operating Mode**	
	⊟ **Tag Counts**	
	Static Tags	130

Static Tags: Provides the total number of defined static tags at this level (device or channel). This information can be helpful in troubleshooting and load balancing.

Device Properties — Scan Mode

The Scan Mode specifies the subscribed-client requested scan rate for tags that require device communications. This setting works differently in this driver than in other drivers. Instead of determining the rate of device communications, it determines the rate that newly cached values are available to subscribed clients. It does not affect the rate of reads sent to the target server. Read frequency is controlled entirely by

static settings configured in **Device Properties — OPC Group**. Generally, Scan Mode should be left in its default state. Synchronous and asynchronous device reads and writes are processed as soon as possible; unaffected by the Scan Mode properties.

Note: In this driver, tag reads occur at the rate defined in **Device Properties — OPC Group** regardless of the Scan Mode setting.

Property Groups	Scan Mode	
General	Scan Mode	Respect Client-Specified Scan Rate
Scan Mode	Initial Updates from Cache	Disable

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

- **Respect Client-Specified Scan Rate**: This mode uses the scan rate requested by the client.
- **Request Data No Faster than Scan Rate**: This mode specifies the value set as the maximum scan rate. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
  Note: When the server has an active client and items for the device and the scan rate value is increased, the changes take effect immediately. When the scan rate value is decreased, the changes do not take effect until all client applications have been disconnected.
- **Request All Data at Scan Rate**: This mode forces tags to be scanned at the specified rate for sub-scribed 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 — OPC Group

This group specifies the properties of the OPC group associated with the device.

Property Groups	OPC Group	
General	Group Name (Optional)	
Scan Mode	Update Mode	Exception
OPC Group	Update/Poll Rate (ms)	1000
Communication Parameters	Percent Deadband	0
Watchdog	Language ID	1033
Import	Write Method	Asynchronous
	Read Method	Asynchronous

Group Name: This property specifies an optional name for identifying the group. If nothing is entered, the underlying OPC server will generate a unique name.

Update Mode: This property specifies the update mode. Options include Exception and Poll. Descriptions of the options are as follows:

- **Exception:** This option is added active, and the server will notify the driver of data value and quality changes. The driver first tries to establish an OPC 2.0 callback for the group. If that is not available, the driver attempts to establish an OPC 1.0 callback.
- **Poll:** This option is added inactive, and the driver reads all items attached to the group / device at the configured update / poll rate (or as close as possible). OPC DA 2.0 asynchronous reads are used if supported by the OPC server. If that is not supported, OPC DA 1.0 asynchronous reads are used.
  ● **Note**: In Poll mode, all tags are added as inactive. The server briefly sets the items to active during each poll cycle; as such, users may notice the active tag count in the server's status bar fluctuate. This is normal.

Update / Poll Rate (ms): When Exception mode is used, this property specifies how often the underlying OPC server should provide updates for changing data. When Poll mode is used, this property specifies how often the driver should read the items attached to this group. The valid range is 0 to 3600000 milliseconds. The default setting is 1000 milliseconds.

Percent Deadband: This property specifies the percent change in data required to notify the client of a data change. This setting is used for Exception mode only.

Language ID: This property specifies the language used by the underlying server when returning values as text for operations on the group. The default setting is 1033 (English).

Write Method: The driver used Asynchronous OPC write operations by default. Select Synchronous for diagnosis of communication issues or to support non-standard OPC server installations. This property should be set to Asynchronous value in most circumstances.

Read Method: The driver uses Asynchronous OPC read operations by default. Select Synchronous for diagnosis of communication issues or to support non-standard OPC server installations. This property should be set to Asynchronous in most circumstances. Synchronous read method is only available when the update mode is set to polling.

Device Properties — Communications Parameters

The Communications Parameters specifies the maximum number of items that can be included in each Read and Write request, as well as the asynchronous Read and Write timeouts.

Property Groups	Request Size	
General	Max. Items per Read	512
Scan Mode	Max. Items per Write	512
OPC Group	Request Timeout	
Communication Parameters	Read Timeout (ms)	1000
Watchdog	Write Timeout (ms)	1000
Import	Read after Write	Enable

Request Size

Max. Items per Read: Specify a limit on the number of items that can be included in a single Read request. The valid range is 1 to 512. The default is 512.

Max. Items per Write: Specify a limit on the number of items that can be included in a single Write request. The valid range is 1 to 512. The default is 512.

Request Timeout

Read Timeout (ms): Specify how long the driver waits for a Read complete notification to be returned from the server before any other Read or Write requests are sent. If the expected notification is not received, the items included in the request are set to Bad quality (and remain Bad until the next successful Read). ⬤ **Note**: Read settings are disabled when the Update Mode is Exception. The driver does not issue Read requests to the server.

Write Timeout (ms): Specify how long the driver waits for a Write complete notification to be returned from the server before sending any other Write or Read request. If the expected notification is not received, the driver logs a Write failed message on timeout.

Read after Write: Choose Enable to force an explicit read after a write command (to confirm the new value). Choose Disable to update after the next publish or poll response. The default is Enable.

Device Properties — Watchdog

When enabled, this property monitors callback integrity when using the Exception Update mode. A watchdog item is required to force callbacks at the requested update rate. Users can select the item to be the watchdog; if one is not chosen, the driver selects an active item on its own.

Watchdog: Enable the watchdog **Watchdog Callback Monitoring** feature.

Item ID of Watchdog Tag: Specify the Item ID of the tag to be automatically added by the driver on connection. This field may be left blank if one or more suitable watchdog tags is active for the connection's duration. 🍀 **Tip**: To locate a watchdog Item ID on the network, click the browse (...) button.

Missed Updates Before Reconnect: Specify the number of update periods that may pass without a data change notification from the underlying server before the driver assumes there is a problem and reconnects. The valid range is 2 to 10. The default is 3.

Watchdog Callback Monitoring

Applying the Exception selection for Update Mode is generally more efficient than polling. Without the Watchdog, users may not know whether a long time between update notifications is because the values

associated with the group have remained the same or whether the underlying OPC server (or COM callback connection) has failed.

Because of this uncertainty, it is recommended that users employ a watchdog. This is an item configured in the underlying OPC server, whose value changes at least once per update period. As long as this item is referenced by the driver, it forces an update notification for every update period. When the watchdog monitoring feature is enabled, the driver tracks the time since the last update. When this time exceeds a predefined limit, the driver assumes there has been a failure and attempts to reconnect to the server.

🔹 *For more information, refer to **Watchdog**.*

Device Properties — Import

This group is used to browse the server for tags.
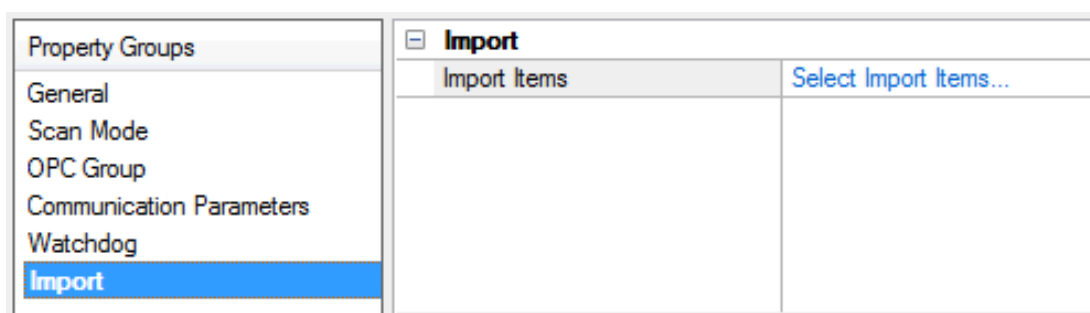
Import Items: Click **Select Import Items...** to invoke the Select Items to Import dialog.

The driver adds all items to the tree and then validates them in the **Import Items** list. To add items to import, select them from the tree and click **Add Items >**. The Import Items list will show the items that will be imported. To remove items from import, select and click **< Remove Items**. Then, select **OK**.

The driver browses the server's address space and displays the results in the tree control on the left. Items selected for import will appear in the list control on the right. To populate the Import Items list, select one or more items in the browse tree and then click **Add Items >**. To add items under a specific branch, first select that branch and then click **Add branch >**. To remove items from the Import Items list, select one of more items in the browse tree and then click **< Remove Items**. To add the selected items to the current tag database, click **OK**.

⚪ **Note**: The Import Item list box is only a list of tags that the user would like to import. The driver validates the tags after the device has completed configuration. A message is posted in the event log if any tag is found to be invalid.
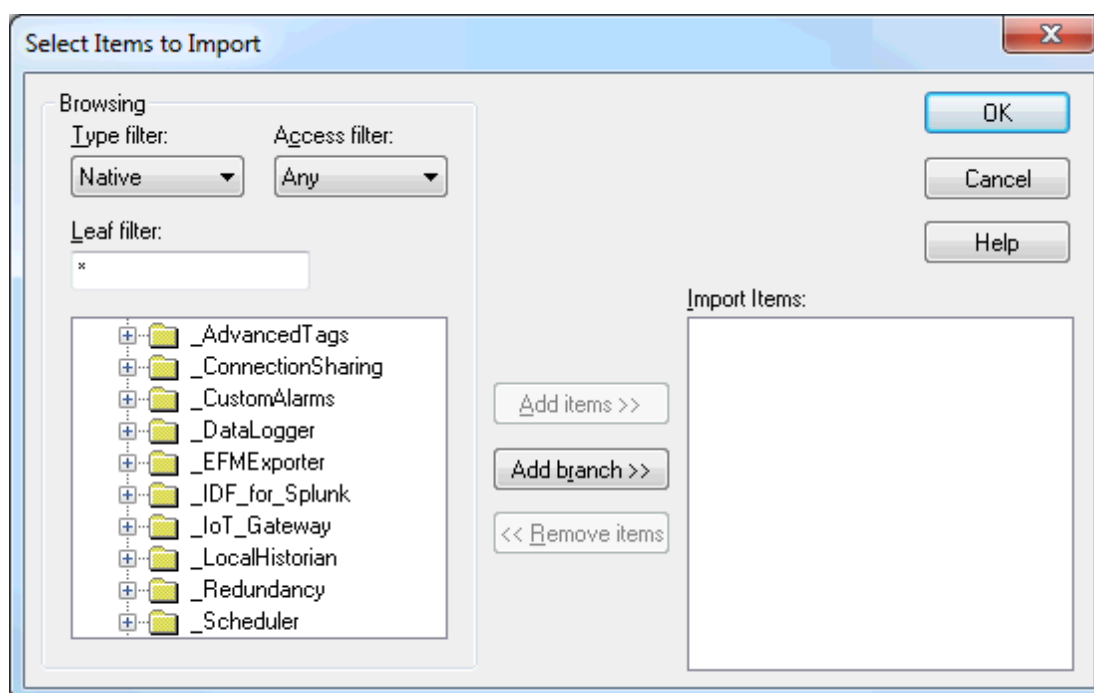
Device Properties — Redundancy

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
Byte	Unsigned 8-bit value bit 0 is the low bit bit 7 is the high bit
Char	Signed 8-bit value bit 0 is the low bit bit 6 is the high bit bit 7 is the sign bit
Word	Unsigned 16-bit value bit 0 is the low bit bit 15 is the high bit
DWord	Unsigned 32-bit value bit 0 is the low bit bit 31 is the high bit
QWord	Unsigned 64-bit value bit 0 is the low bit bit 63 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
Long	Signed 32-bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
LongLong	Signed 64-bit value bit 0 is the low bit bit 62 is the high bit bit 63 is the sign bit
Float	32-bit floating point value. The driver interprets two consecutive 16-bit registers as a floating point value by making the second register the high word and the first register the low word.
Double	64-bit floating point value
String	Null-terminated character array
Date	Date YYYY-MM-DDTHH:MM:SS.MMM

Address Descriptions

The OPC DA Client Driver address descriptions refer to fully qualified item IDs of the server item.

⬤ **Note**: Arrays are supported for all data types.
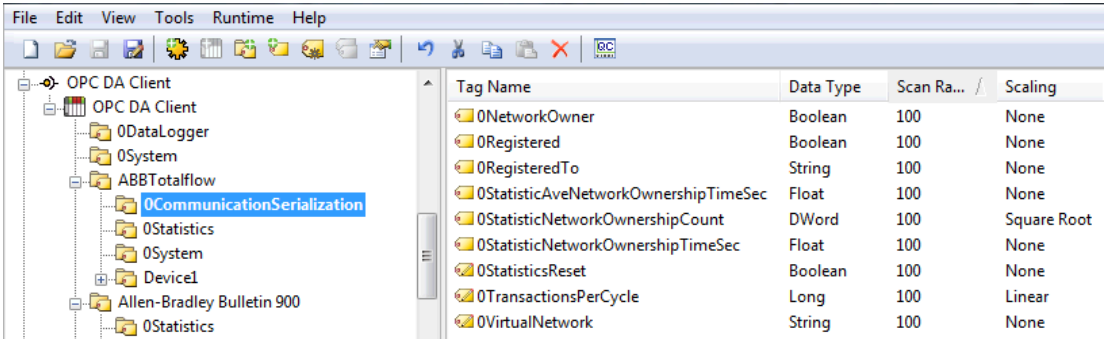
Alternatively, an item's Access Path can be specified by creating a tag and using the following syntax for the item address:
$AccessPath$ItemID.

Field	Description
Access Path	An optional field specified by the client to provide to the server a suggested data path (how to get the data).
Item ID	The value uniquely identifies an OPC data item.

Examples
$Channel1.Device1$Tag1
$Channel1.Device1.Group1.Group2$Tag3
$OpcMatu2.Mat2$COM1.STATION:42.REG:40001;0,4095,-100.0,+1234.0

System Tags

Tag	Description	Data Type	Access
_OPCGroupActive	Specifies whether the OPC group that the driver adds to communicate with the OPC server is active (1) or inactive (0). This can be used to control messages when initially adding items. *See **Appendix** for more information.*	Boolean	Read/Write

Error Descriptions

The following error / warning messages may be generated. They are listed here in alphabetical order. Click on the link for a description of the message.

Add item failed for <item> on device <device>. Function error: <error code>.
Add item failed for <item>on device <device>s. Item error: <error code>.
Add watchdog item failed for <item> on device <device name>. Item error: <error code>. <error description>.
Async Write 1.0 failed for <item> on device <device>. Callback item error: <error code>.
Async Write 1.0 failed for <item> on device <device>. Function error: <error code>.
Async Write 1.0 failed for <item> on device <device>. Item error: <error code>.
Async Write 2.0 failed for <item> on device <device>. Callback item error: <error code>.
Async Write 2.0 failed for <item> on device <device>. Function error: <error code>.
Async Write 2.0 failed for <item> on device <device>. Item error: <error code>.
Attempt to map redundant item to secondary device failed. | Item = 'text', Secondary device = '<channel name>.<device name>'.
Failed to acquire item management interface for device <device>.
Failed to add group for device <device>. Reason: <error code>.
Failed to connect to server on channel <channel>.
Failed to establish callback for device <device >.
Reconnecting server on channel <channel> due to GetStatus failure.
Reconnecting server on channel <channel> due to shutdown notification.
Reconnecting server on channel <channel> due to watchdog failure on device <device>.
Tag <item> on device <device> was not imported. Item failed OPC server validation.
Unknown data type for tag <item> on device <device>. Using default.

Add item failed for <item> on device <device>. Function error: <error code>.

Error Type:
Serious

Possible Cause:
The server rejected an add items request.

Result:
The driver will not be able to read or write to the associated tag(s).

Possible Solution:
The solution depends on the specific error code. There is most likely a problem in an underlying server (such as resource limits).

Add item failed for <item> on device <device>s Item error: <error code>.

Error Type:
Serious

Possible Cause:

The server failed to add the specified item.

Result:

The driver will not be able to read or write to the associated tag(s).

Possible Solution:

Verify that the Item ID set for the tag is valid.

Add watchdog item failed for <item> on device <device>. Item error: <error code>. <error description>.

Error Type:

Warning

Possible Cause:

The driver failed to add the specified watchdog item. The particular reason can be determined from the error code. The driver will include an error description in this message when available from the server. The specified Item ID is not usually valid in the current server configuration.

Possible Solution:

Verify the watchdog's Item ID and change it as needed.

Async Write 1.0 failed for <item> on device <device>. Callback item error: <error code>.

Error Type:

Warning

Possible Cause:

The server processed the write request, but failed to successfully write to the underlying device.

Possible Solution:

Verify that communications to device are good and that the point is writable.

Async Write 1.0 failed for <item> on device <device>. Function error: <error code>.

Error Type:

Warning

Possible Cause:

The server rejected a write request that included the indicated item.

Possible Solution:

The solution depends on the specific error code. There is most likely a problem in the underlying server (such as resource limits).

Async Write 1.0 failed for <item> on device <device>. Item error: <error code>.

Error Type:
Warning

Possible Cause:
This message will be logged if the item was included in a write request that was rejected because of one or more invalid items. The error code here will indicate if this was one of the invalid items.

Possible Solution:
Verify that Item ID is valid and that item is valid.

Async Write 2.0 failed for <item> on device <device>. Callback item error: <error code>.

Error Type:
Warning

Possible Cause:
The server processed the write request, but failed to successfully write to underlying device.

Possible Solution:
Verify that communications to device are good and that point is writable.

Async Write 2.0 failed for <item> on device <device>. Function error: <error code>.

Error Type:
Warning

Possible Cause:
The server rejected a write request that included the indicated item.

Possible Solution:
The solution depends on the specific error code. There is most likely a problem with an underlying server (such as resource limits).

Async Write 2.0 failed for <item> on device <device>. Item error: <error code>.

Error Type:
Warning

Possible Cause:
The server was unable to process a write request for this item. Other items in request may have been accepted.

Possible Solution:
Verify that the Item ID is valid and that the item is valid.

Attempt to map redundant item to secondary device failed. | Item = 'text', Secondary device = '<channel name>.<device name>'.

Error Type:
Warning

Possible Cause:

- The tags are configured differently for the primary and the secondary.
- Tags were created with a client driver import feature with a different folder structure.
- The primary and secondary addressing must match below the device level, which includes any tag groupings created.

Possible Solution:
Correct the differences in the configuration, folder structure, or syntax for the primary and secondary and try again.

Failed to acquire item management interface for device <device>.

Error Type:
Serious

Possible Cause:
The driver was not able to obtain the item management COM interface from the server.

Result:
The driver will not be able to read or write to tags associated with device 0 /group.

Possible Solution:
This is a required interface. Contact Technical Support.

Failed to add group for device <device>. Reason: <error code>.

Error Type:
Serious

Possible Cause:
Server rejected request to add specified group.

Result:
The driver will not be able to read or write to tags associated with device/group.

Possible Solution:
The solution depends on the specific error code. Because the error may have been caused by a duplicate group name, try a different name in the OPC Group device properties page. It may also be due to a problem in an underlying server (such as resource limits).

Failed to connect to server on channel <channel>.

Error Type:
Serious

Possible Cause:
Could not connect to server.

Possible Solution:
Check the Program ID, machine name and network connection.

Failed to establish callback for device <device>.

Error Type:
Serious

Possible Cause:
The driver was not able to establish a COM callback for the device/group.

Result:
The driver will not be able to read, write or receive data change notification for tags associated with the device or group.

Possible Solution:
1. Verify that OPC DA 1.0 or 2.0 is supported.

2. Verify the DCOM settings (if from a remote server).

Reconnecting server on channel <channel> due to GetStatus failure.

Error Type:
Warning

Possible Cause:
The driver did not receive a response to the GetStatus request. A server, network or COM failure is assumed.

Possible Solution:
The driver will automatically reconnect.

Reconnecting server on channel <channel> due to shutdown notification.

Error Type:
Warning

Possible Cause:
The server has sent a notification that it is shutting down.

Possible Solution:
The driver will automatically reconnect.

Reconnecting server on channel <channel> due to watchdog failure on device <device>.

Error Type:
Warning

Possible Cause:
The time that elapsed since the last data change notification was longer than the configured limit. A server, network or COM failure is assumed.

Possible Solution:
The driver will automatically reconnect.

Tag <item> on device <device> was not imported. Item failed OPC server validation.

Error Type:
Warning

Possible Cause:
The item selected for import did not have a valid Item ID. Some servers provide Item IDs that are intended to give the user information, but are not themselves valid Item IDs. For example, this server provides driver address syntax hints to the browser.

Possible Solution:
This error can be avoided by not selecting special purpose Item IDs in the import item browser; however, no harm is done by selecting them.

Unknown data type for tag <item> on device <device>. Using default.

Error Type:
Warning

Possible Cause:

1. The target server did not return a data type.

2. The data type that was entered is not supported by the driver; e.g. QWord or LLong.

Possible Solution:

1. The server determines the data type once a client is connected.

2. Enter a data type that is supported by the driver.

Appendix — Server-Client Communications

The OPC DA client driver does not pass calls from a client connected to the server directly to the connected server. For example, if an OPC DA client (Client 1) adds 10,000 active items to the server (Server 1), the server adds those items to the second OPC DA server (Server 2) through the OPC DA client driver in groups of 128 or less. While those Add calls occur, Server 1 attempts an initial Read on the tags that have been added. Since those items added are active, Server 2 starts sending data changes to Server 1. During the process of adding those 10,000 items, there are Reads and Data Changes being reported. This can overwhelm some OPC servers.

To manage communications:

1. Access Client 1 and create a connection to Server 1.

2. Create a tag group.

3. Add the _OPCGroupActive device system tag in the Add Items dialog.

4. Write a value of 0 to the system tag. This sets the OPC group that is created between Server 1 and Server 2 to be inactive.

5. Add the remaining tag items to that group as inactive. This causes the AddItems OPC calls to occur.

6. Wait a small amount of time. Depending on the number of items and the speed of the communications, this may take several seconds.

7. Set the tags in Client 1 to active to cause initial Reads to occur.

8. Wait a small amount of time. Depending on the number of items and the speed of the communications, this may take several seconds.

9. Write a value of 1 to the _OPCGroupActive system tag. This causes the OPC Group to become active and Data Changes from Server 2 are sent to Server1 and up to Client 1.

🍀 **Tip**: If adding items and doing the initial read at the same time is not a concern, the items can be added as Active in step 5 and jump directly to step 8.

Index

$

_

A

B

C

D

E

F