

# InTouch Client Driver

© 2017 PTC Inc. All Rights Reserved.

# Table of Contents

<b>InTouch Client Driver</b> .....	<b>1</b>
<b>Table of Contents</b> .....	<b>2</b>
InTouch Client Driver .....	4
Overview .....	4
External Dependencies .....	5
System Configuration .....	5
<b>Driver Setup</b> .....	<b>6</b>
Channel Properties — General .....	6
Channel Properties — Write Optimizations .....	7
Channel Properties — Advanced .....	8
Device Properties — General .....	8
Device Properties — Scan Mode .....	10
Device Properties — Tag Generation .....	11
Device Properties - Tag Import Settings .....	13
Device Properties - Mode .....	17
<b>Automatic Tag Database Generation</b> .....	<b>19</b>
<b>Optimizing Communications</b> .....	<b>20</b>
<b>Data Types Description</b> .....	<b>22</b>
<b>Address Descriptions</b> .....	<b>23</b>
<b>Error Descriptions</b> .....	<b>26</b>
Missing address .....	26
Device address <address> contains a syntax error .....	26
Address <address> is out of range for the specified device or register .....	27
Data Type <type> is not valid for device address <address> .....	27
Device address <address> is read only .....	27
Array size is out of range for address <address> .....	27
Array Support is not available for the specified address: <address> .....	28
Device <Device name> is not responding .....	28
Unable to write to <address> on device <device name> .....	28
Low resources (OS error: <error code>) .....	29
Connection to InTouch failed. Server runtime process must be configured for interactive mode ...	29
Tag import failed (OS error: <error code>) .....	29
Tag <tag name> could not be imported due to unresolved datatype .....	29
Tag <tag name> could not be imported due to name conflict .....	30
Tag <tag name> could not be imported due to address length limitation .....	30
Imported tag name changed from <original name> to <new name> .....	30

Tag browser error .....	30
<b>Index</b> .....	<b>31</b>

## InTouch Client Driver

---

Help version 1.029

### CONTENTS

#### Overview

What is the InTouch Client Driver?

#### Driver Setup

How do I configure this driver to communicate with InTouch?

#### Automatic Tag Database Generation

How can I easily configure tags for this driver?

#### Optimizing Communications

How do I get the best performance from this driver?

#### Data Types Descriptions

What data types does this driver support?

#### Address Descriptions

How do I address a tag in the InTouch tag database?

#### Error Descriptions

What error messages does this driver produce?

### Overview

---

The InTouch Client Driver provides a reliable way to connect InTouch applications to OPC client applications; including HMI, SCADA, Historian, MES, ERP, and countless custom applications.

This driver can be used to expose InTouch tag databases to OPC clients. This includes all I/O and memory tags in the InTouch Tag Database, including those configured as local and remote tags. This driver can import tags defined in InTouch applications, greatly reducing the project development effort.

 **See Also:** [External Dependencies](#)

---

## External Dependencies

---

The InTouch Client Driver has external dependencies. Wonderware InTouch™ 7.1 or higher is required.

● **Note:** The server must exist on the same machine as an InTouch node. Data from remote InTouch nodes can be accessed via the local InTouch node or with multiple OPC servers.

---

## System Configuration

---

### System Path

To access live data with this driver, the system path environment variable must include the location of the "wwheap.dll" and "ptacc.dll" files that were installed with InTouch. When InTouch is installed before the driver, the server installation program automatically adds the file location. When InTouch is installed after the driver, it is recommended that the server installation program be run again to update the path variable. The path may also be updated manually. For more information, follow the instructions below.

1. To start, locate wwheap.dll and ptacc.dll on the machine.
  - For InTouch 7.1, both files can typically be found in **C:\Program Files\FactorySuite\InTouch**.
  - For InTouch 8.0 and higher, wwheap.dll can typically be found in **C:\Program Files\Common Files\Archestra** and ptacc.dll can typically be found in **C:\Program Files\Wonderware\InTouch**.
2. Next, invoke the **Control Panel** dialog and then select **System**.
3. Click **Advanced | Environment Variable** and then select **Path**.
4. Next, click **Edit**. Append the current setting with the location(s) of wwheap.dll and ptacc.dll, separating each with a semicolon.

## Driver Setup

Before live data can be accessed with this driver, InTouch® must be added to the system path. Server projects may be created before this is done, however. For more information, refer to [System Configuration](#).

The first step in creating a server project is to create a channel that uses this driver. Next, add a device to that channel. This device acts as a client of InTouch WindowViewer™, which must be running on the same machine. Data in remote InTouch nodes can be accessed only if it is linked to the local InTouch node with remote tags. For more information, refer to [Optimizing Communications](#).

The device object can be configured to import tags from an InTouch project, which greatly reduces the project development time. For more information, refer to [Tag Import Settings](#) and [Automatic Tag Database Generation](#).

The device object can obtain data from InTouch by polling, data change notifications or a combination method. For more information, refer to [Mode](#).

## 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.

Property Groups	[-] <b>Identification</b>	
General	Name	
Write Optimizations	Description	
Advanced	Driver	
	[-] <b>Diagnostics</b>	
	Diagnostics Capture	Disable

### 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 — 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.

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

**Non-Normalized Float Handling:** Non-normalized float handling allows users to specify how a driver handles non-normalized IEEE-754 floating point data. 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. Descriptions of the options are as follows:

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

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

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

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

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

## Device Properties — General

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

Property Groups	Identification	
General	Name	
Scan Mode	Description	
Auto-Demotion	Channel Assignment	
Redundancy	Driver	
	Model	
	ID Format	Decimal
	ID	2
	Operating Mode	
	Data Collection	Enable
	Simulated	No

## 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.

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

**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 — Tag Generation

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

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

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

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

Property Groups	<input type="checkbox"/> <b>Tag Generation</b>	
General	On Property Change	Yes
Scan Mode	On Device Startup	Do Not Generate on Startup
Timing	On Duplicate Tag	Delete on Create
Auto-Demotion	Parent Group	
<b>Tag Generation</b>	Allow Automatically Generated Subgroups	Enable
Tag Import	Create	Create tags
Redundancy		

**On Property Change:** If the device supports automatic tag generation when certain properties change, the **On Property Change** option is shown. It is set to **Yes** by default, but it can be set to **No** to control over when tag generation is performed. In this case, the **Create tags** action must be manually invoked to perform tag generation.

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

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

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

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

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

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

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

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

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

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

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

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

## Device Properties - Tag Import Settings

The InTouch Client Driver can import tags from the InTouch project, reducing project development time. To access these settings after the Device Wizard has completed, click **Edit | Device Properties** and select **Tag Import Settings**.

Property Groups	[-] <b>Tag Import Settings</b>	
General	Import Method	Import from InTouch Project
Scan Mode	InTouch Project Folder	
Tag Generation	InTouch CSV File	*.csv
<b>Tag Import Settings</b>	[-] <b>Tag Import Options</b>	
Mode	Include Tag Descriptions	Enable
	Import System Tags	Enable
	Tag Naming	Enhanced

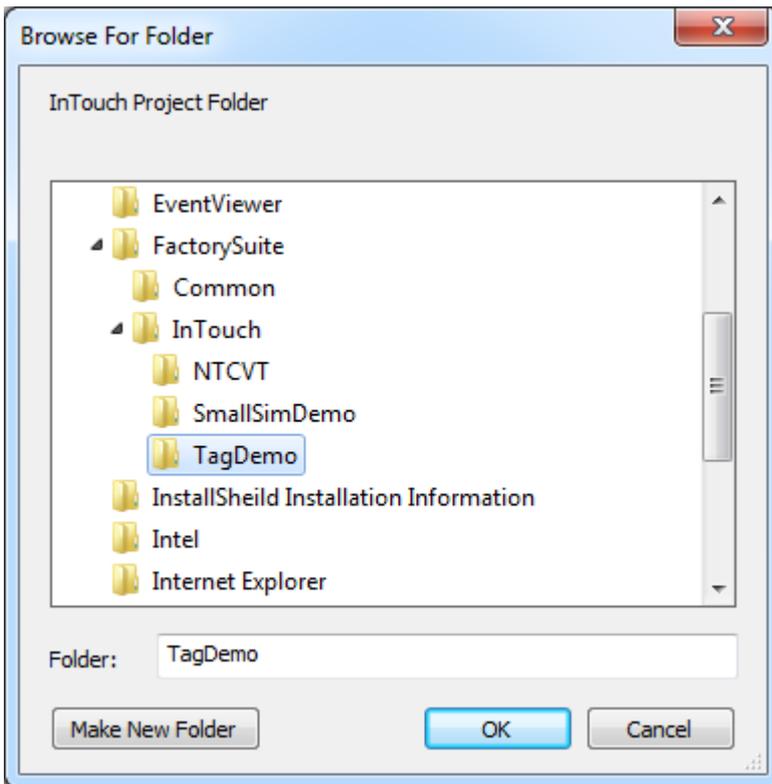
### Import Method

There are two methods that may be used to import tags. Descriptions of the options are as follows:

- **Import from InTouch Project:** This option allows tags to be imported directly from an InTouch project. This is the default setting.
- **Import from InTouch CSV File:** This option requires that the tag database be exported to a CSV file first through the use of the InTouch DBDump utility. This method is recommended for larger projects.

### InTouch Project Folder

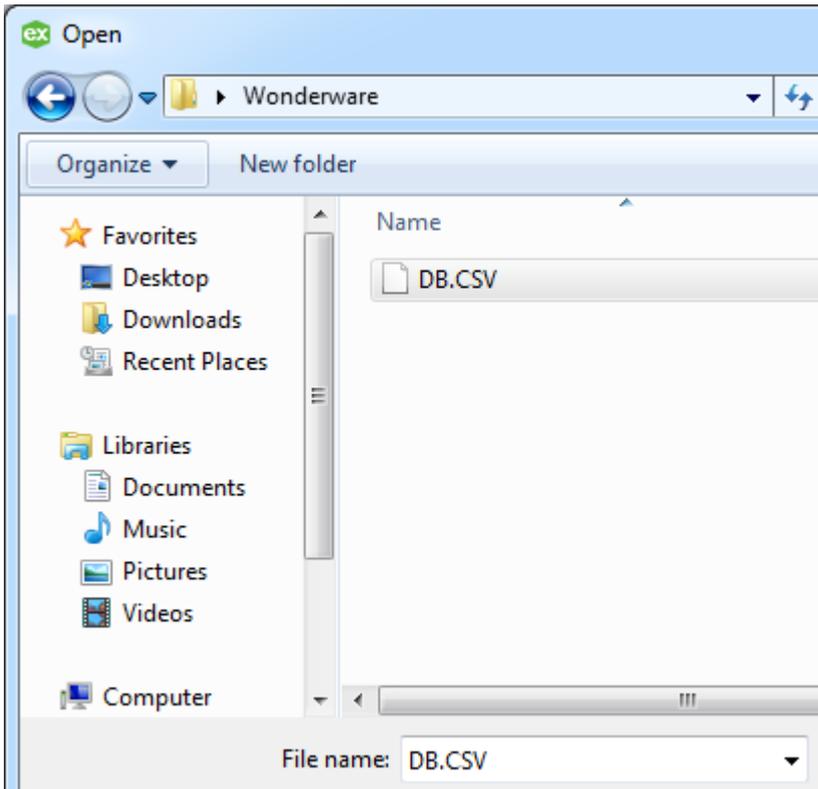
This setting specifies the root folder of the InTouch project from which tags are imported, and is used in conjunction with the **Import from InTouch project** method. The file path can be typed manually or browsed for using the Browse button. If no folder is specified, the most recent InTouch project is used. The folder browser should appear as shown below.



In this example, the root folder of an InTouch project called "ThousandTagDemo" has been selected. Once **OK** is clicked, the full path to this folder is automatically placed in the InTouch project folder box.

### InTouch CSV file

This setting specifies the name and path of the InTouch CSV file from which tags are imported and is used in conjunction with the **Import from InTouch CSV file** method. The file path and name can be typed manually or browsed for using the Browse button. The file browser should appear as shown below.



In this example, the file in an InTouch project folder called "DB.CSV" has been selected. Once **OK** is clicked, the full path and file name is automatically placed in the InTouch CSV file box.

## Options

To view more options, expand the **Tag Import Options** section in the **Tag Import Settings** properties group. The window should appear as shown below.

Tag Import Options	
Include Tag Descriptions	Enable
Import System Tags	Enable
Tag Naming	Enhanced

Descriptions of the properties are as follows:

- **Include Tag Descriptions:** When enabled, this option includes the descriptions attached to each InTouch tag included with tags generated in the OPC server.
- **Import System Tags:** When enabled, this option imports the InTouch system tags (such as \$ApplicationVersion, \$Date, \$Time, and so forth). Remember that since tag names in the OPC server must start with an alphanumeric character, the dollar sign in each imported system tag name is changed to a zero (0).
- **Tag Naming:** This property specifies one of the two tag naming options: Enhanced or Legacy. Enhanced has fewer naming constraints and is consistent with the naming requirements of the current OPC server. Tag names cannot have a period, double quotes or start with an underscore. Legacy enforces the stricter naming requirements of previous versions of this driver. Tag names must start with a letter, and the name must consist of letters and digits only.

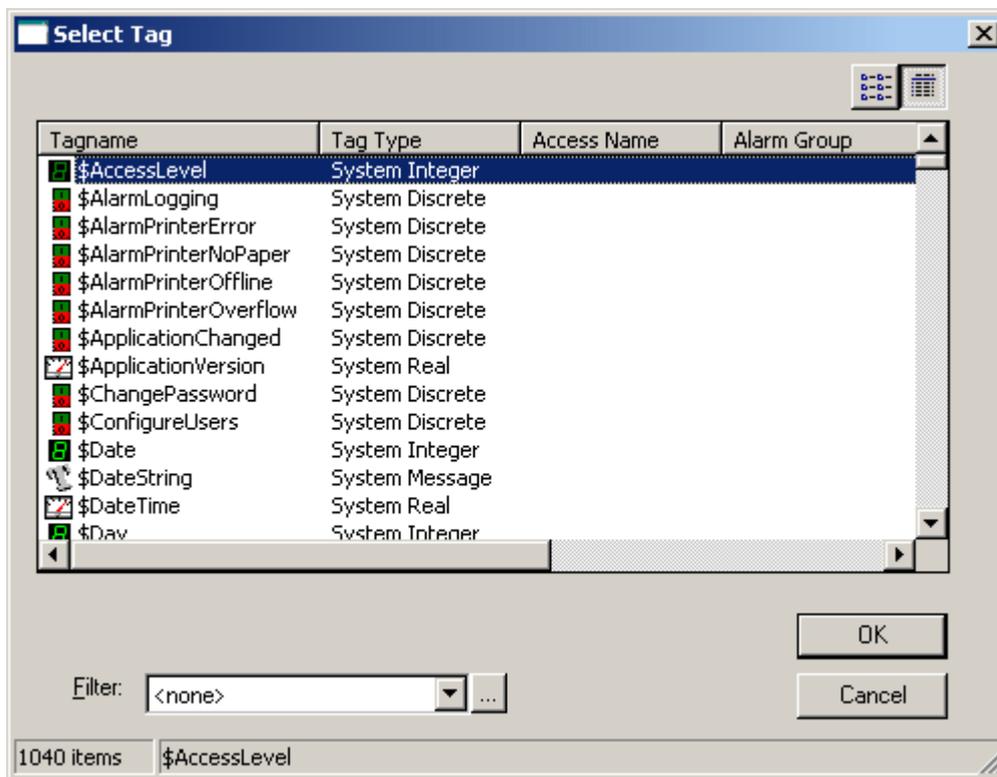
● **Note:** When changing any of the settings in the Tag Import Settings tab, click **Apply** to apply the changes before proceeding.

## Tag Generation

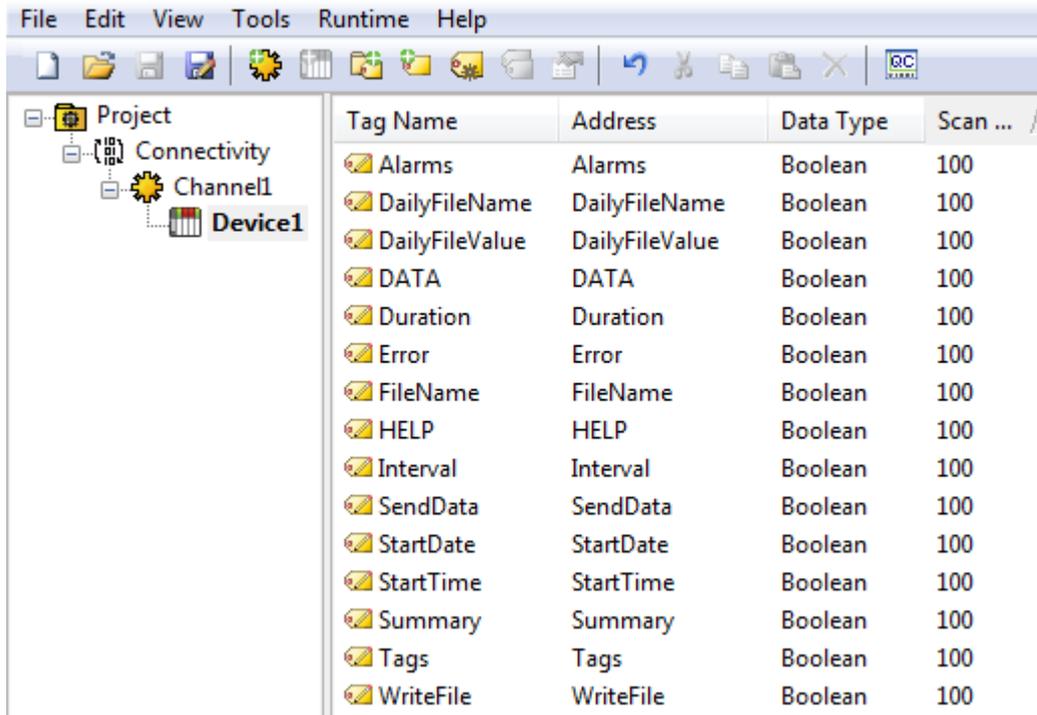
Once the settings described above are configured, the OPC server's automatic tag database generation feature may be invoked. It may be configured to run automatically or be triggered manually. For more information, refer to [Automatic Tag Database Generation](#).

### Example

1. To start, specify the **Tag Import Settings** as described above.
2. Next, double click **Create** to trigger automatic tag database generation manually.
3. The tags defined in InTouch should appear as shown below.



4. After the tags have been imported into the OPC server, the dialog should appear as shown below.



**Note:** Generated tag names are essentially the same as the tag names in InTouch. Minor name modifications may be required and are indicated in the Event Log when they occur. The addresses in the OPC server are the same as the tag names in InTouch.

### Supported InTouch Tag Types

The following InTouch tag types can be imported into to the server with this driver.

InTouch Type	Data Type	Access
I/O Discrete	Boolean	Read Only or Read/Write*
I/O Integer	Long	Read Only or Read/Write*
I/O Real	Float	Read Only or Read/Write*
I/O Message	String	Read Only or Read/Write*
Memory Discrete	Boolean	Read/Write
Memory Integer	Long	Read/Write
Memory Real	Float	Read/Write
Memory Message	String	Read/Write
Indirect Discrete	Boolean	Read/Write
Indirect Analog	Float	Read/Write
Indirect Message	String	Read/Write

\*This depends on the access configured in InTouch.

### Device Properties - Mode

Servicing data requests from an external application (such as this driver) can place a significant work load on the InTouch application. In some cases, it even causes WindowViewer to slow down. Therefore, three modes of data access are offered that can be used to optimize communications with InTouch. These settings

affect how data is acquired for all tags associated with the device. It is possible to override the mode for individual tags using addressing options. For information on choosing the settings in Device Properties (and other project design considerations) refer to [Optimizing Communications](#).

● **See Also:** [Address Descriptions](#)

Property Groups	Mode	
General	Mode	Driver Polls InTouch
Scan Mode	Maximum Active Time (ms)	60000
Tag Generation	Delete Inactive Tags	Enable
Tag Import Settings		
<b>Mode</b>		

**Mode:** Select how the driver communicates with Wonderware InTouch.

- **Driver Polls InTouch:** When enabled, the driver issues a read request to InTouch once per update interval for each tag associated with the device. This option is recommended for rapidly changing data because it has the greatest impact on InTouch performance. If the scan period of a tag is greater than the **Maximum active time** property, the driver deactivates the tag between reads to reduce InTouch's processing load.
- **InTouch Notifies Driver:** When enabled, InTouch sends the driver a notification when a data value has changed. The driver does not interrupt InTouch with repeated read requests as it does in **Polled Mode**. This option is recommended for slowly changing data.
- **Combination:** When enabled, the driver sets the update mode for each tag depending on its scan rate (as defined in the OPC client) and the **Maximum active time**. If the time between scans is less than or equal to the maximum active time, the tag is placed in Polled Mode. Otherwise, the tag is placed in **Notified Mode**.

**Maximum Active Time:** This property specifies how long the driver should keep tags active. Since many active tags can burden the InTouch memory manager, tags associated with slowly changing data should not be kept active. Care should be taken not to make this value too low, however, since repeated activation/re-activation requires processing time.

**Delete Inactive Tags:** When enabled, tags are completely deleted in between reads. This reduces the work load on InTouch. Users should use this option with care, however, since the repeated creation and destruction of tags requires processing time.

● **Notes:**

1. Each of these settings can be changed on-the-fly.
2. InTouch stops polling a tag's data source if all of its clients (including the driver) have deactivated the tag and if that tag's value is not currently displayed in the active WindowViewer window. As a result, the most current value in the data source may not be returned through InTouch the next time this tag is reactivated and read. This should be kept in mind when adjusting the Maximum active time property.

---

## Automatic Tag Database Generation

---

The InTouch Client Driver tag database generation feature is designed to make the OPC application setup a plug and play operation. This drivers can be configured to automatically build a list of OPC tags within the server that correspond to application specific data.

### Server Configuration

The automatic tag database generation features can be customized to fit specific needs. The primary control options may be set during the Database Creation step of the Device Wizard or later by selecting **Edit Properties** | **Tag Import Settings**. For more information, refer to the OPC server's help documentation.

Many basic settings are common to all drivers that support automatic tag database generation. This driver requires additional settings, including the location of the InTouch project from which tags are imported. This information may be specified during the Tag Import step of the Device Wizard or later by clicking **Device Properties** | **Tag Import Settings**. For more information, refer to [Tag Import Settings](#).

### Operation

Depending on the configuration, tag generation may start automatically when the OPC server project starts or be initiated manually at some other time. The OPC server's Event Log shows when the tag generation process started, any errors that occurred while importing the tag information, and when the process completed. Imported tag names may be altered for compatibility. For example, an InTouch tag named "\$Date" appears as "0Date" in the OPC server. Such name changes are noted in the server's Event Log.

## Optimizing Communications

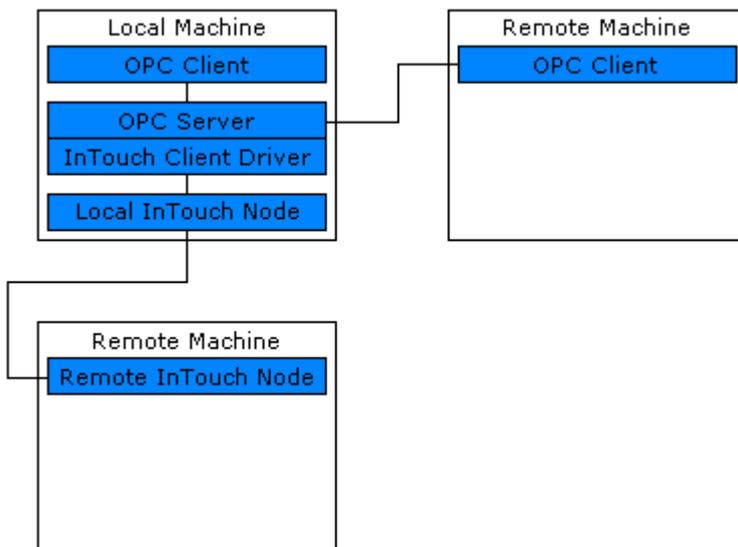
Servicing data requests to an external application (such as this driver) can add a significant work load on WindowViewer. For suggestions on optimizing InTouch Client Driver communications, refer to the notes below.

- Unless the tag's data changes slowly, the Polled Mode should be used. When doing so, the OPC scan rate should be kept down. Furthermore, the **Maximum active time** device mode property should be adjusted so that tags with slow update rates are deactivated in InTouch between reads.
- Users should not add tags to the InTouch Client Driver project unless they are needed. A practical limit on the number of tags is on the order of 10,000. If the InTouch application is busy and/or the computer is heavily burdened, the number of tags should be much lower.
- When there is a variety of slowly and rapidly changing data, it may be beneficial to separate the tags into two devices that operate in notified and polled mode, respectively. Overall server performance may be improved by placing these devices on separate channels, as well.

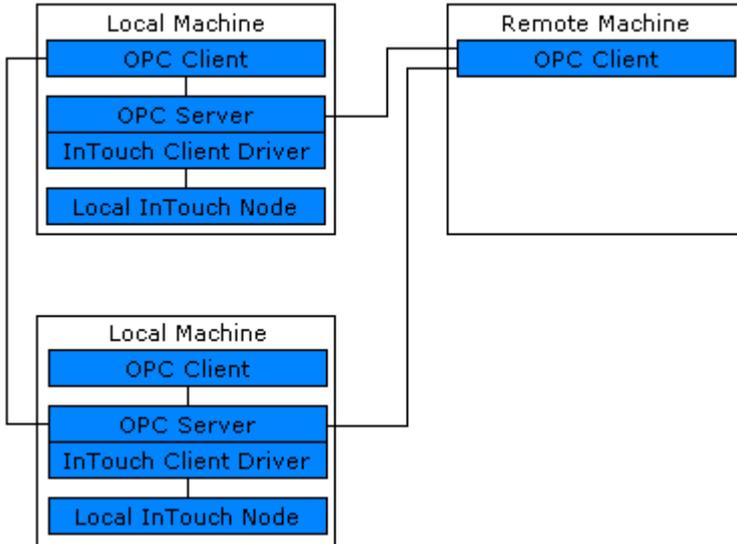
● **Note:** The InTouch Client Driver supports 16 channels.

- Instead of passing large amounts of remote data through the local InTouch node, users should consider using multiple OPC servers. For more information, refer to the images below.

### Channeling Remote Data Through the Local InTouch Node



### Channeling Remote Data Through Multiple OPC Servers



## 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
Short	Signed 16-bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
BCD	Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.
DWord	Unsigned 32-bit value bit 0 is the low bit bit 31 is the high bit
Long	Signed 32-bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
LBCD	Four byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.
Float	32-bit floating point value
Double	64-bit floating point value
String	Null terminated ASCII string

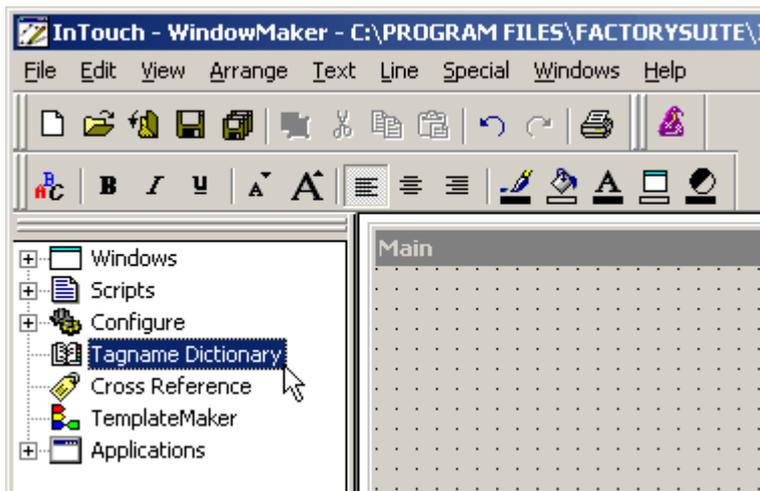
## Address Descriptions

### Basic Addresses

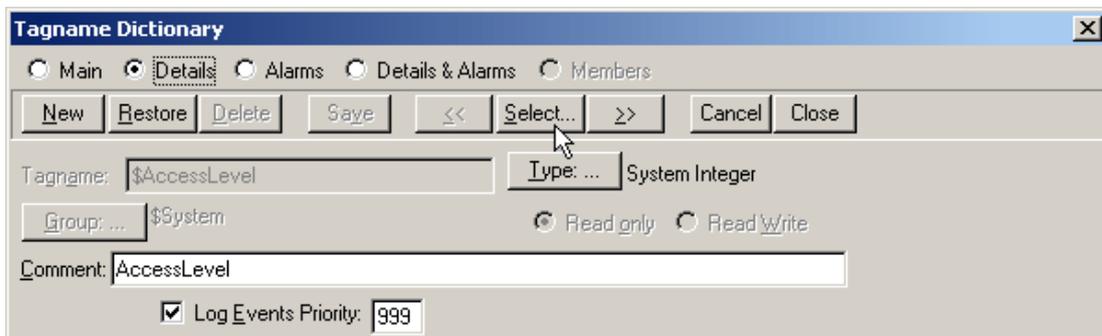
The address of a tag in the server must be the name of that tag in the InTouch Tagname Dictionary. The driver is not able to validate addresses during configuration. During run time, a tag with an invalid address shows bad quality and displays an unknown value.

The easiest way to create tags in the OPC server is to import them directly from InTouch. For more information, refer to [Automatic Tag Database Generation](#). To create the tags manually, browse the names of all defined tags by following the instructions below.

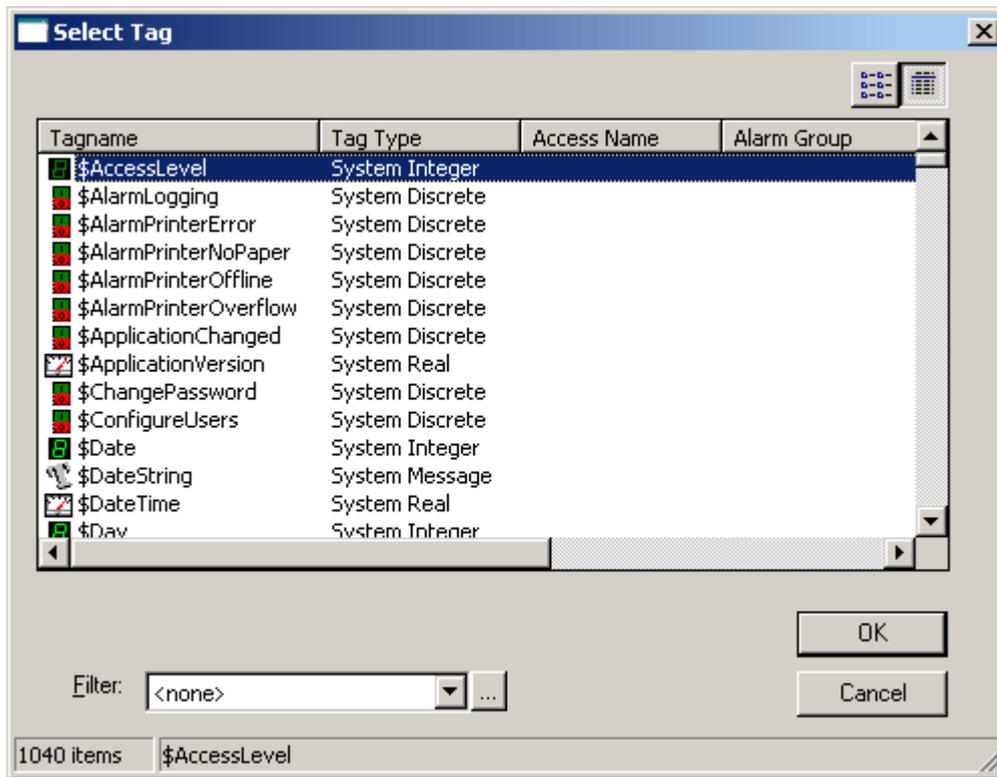
1. To start, load the project into WindowMaker and then select **Tagname Dictionary**.



2. Next, click **Select**.



- The tags should be displayed as shown below.

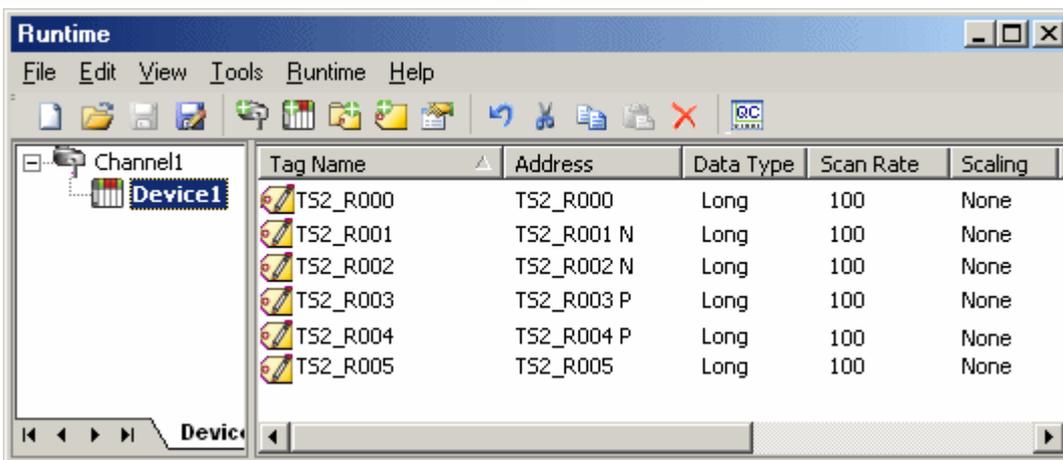


- Name the tags in the OPC server as desired.

**Note:** The addresses in the OPC server must be the InTouch tag names shown above.

### Addresses with Mode Override

A basic address may be appended with a mode override character. An address followed by a space and "N" places that tag in Notified Mode, regardless of the device mode settings. An address followed by a space and "P" places that tag in Polled Mode, regardless of the device mode settings. For more information, refer to [Mode](#).



In this example, tags TS2\_R001 and TS2\_R002 are placed in Notified Mode regardless of the device-level settings. Tags TS2\_R003 and TS2\_R004 are placed in Polled Mode regardless of the device-level settings. All other tags shown above use the default mode specified in Device Properties.

## 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](#)

#### [Array size is out of range for address <address>](#)

#### [Array support is not available for the specified address: <address>](#)

### Device Status Messages

#### [Device <device name> is not responding](#)

#### [Unable to write to <address> on device <device name>](#)

### Driver Error Messages

#### [Low resources \(OS error: <error code>\)](#)

#### [Connection to InTouch failed. Server runtime process must be configured for interactive mode](#)

### Automatic Tag Database Generation Messages

#### [Tag import failed \(OS error: <error code>\)](#)

#### [Tag <tag name> could not be imported due to unresolved datatype](#)

#### [Tag <tag name> could not be imported due to name conflict](#)

#### [Tag <tag name> could not be imported due to address length limitation](#)

#### [Imported tag name changed from <original name> to <new name>](#)

#### [Tag browser error](#)

## Missing address

---

### Error Type:

Warning

### Possible Cause:

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

### Solution:

Re-enter the address in the client application.

## Device address <address> contains a syntax error

---

### Error Type:

Warning

### Possible Cause:

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

**Solution:**

Re-enter the address in the client application.

---

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

---

**Error Type:**

Warning

**Possible Cause:**

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

**Solution:**

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

---

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

---

**Error Type:**

Warning

**Possible Cause:**

A tag address that has been specified statically has been assigned an invalid data type.

**Solution:**

Modify the requested data type in the client application.

---

**Device address <address> is read only**

---

**Error Type:**

Warning

**Possible Cause:**

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

**Solution:**

Change the access mode in the client application.

---

**Array size is out of range for address <address>**

---

**Error Type:**

Warning

**Possible Cause:**

A tag address that has been specified statically is requesting an array size that is too large for the address type or block size of the driver.

**Solution:**

Re-enter the address in the client application to specify a smaller value for the array or a different starting point.

---

**Array Support is not available for the specified address: <address>**

---

**Error Type:**

Warning

**Possible Cause:**

A tag address that has been specified statically contains an array reference for an address type that doesn't support arrays.

**Solution:**

Re-enter the address in the client application to remove the array reference or correct the address type.

---

**Device <Device name> is not responding**

---

**Error Type:**

Serious

**Possible Cause:**

1. InTouch WindowViewer is not running.
2. The system is not configured correctly.

**Solution:**

1. Verify that InTouch WindowViewer is running.
2. Verify that the system is configured correctly.

**See Also:**

[System Configuration](#)

---

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

---

**Error Type:**

Serious

**Possible Cause:**

1. InTouch WindowViewer is not running.
2. The system is not configured correctly.

**Solution:**

Verify that WindowViewer is running.

**See Also:**

[System Configuration](#)

---

**Low resources (OS error: <error code>)**

---

**Error Type:**

Serious

**Possible Cause:**

The driver could not access necessary system resources.

**Solution:**

Shut down unnecessary applications and retry. The returned error code provides specific information.

---

**Connection to InTouch failed. Server runtime process must be configured for interactive mode**

---

**Error Type:**

Fatal

**Possible Cause:**

The server runtime process is configured to run as a service. The inter-process communication mechanism used by this driver and InTouch requires that the server run as an interactive process.

**Solution:**

Use the server administration tool to configure the runtime process for interactive mode.

**Note:**

For more information, refer to the OPC server's help documentation section on Runtime Process.

---

**Tag import failed (OS error: <error code>)**

---

**Error Type:**

Serious

**Possible Cause:**

The driver could not perform tag import due to a system level error condition, such as an attempt to read a corrupted file.

**Solution:**

The returned error code provides specific information.

---

**Tag <tag name> could not be imported due to unresolved datatype**

---

**Error Type:**

Warning

**Possible Cause:**

The data type assigned to the tag in InTouch is not compatible with any of the data types supported by the server.

**Solution:**

None.

---

### **Tag <tag name> could not be imported due to name conflict**

---

**Error Type:**

Warning

**Possible Cause:**

The imported tag name could not be coerced into a valid server tag name.

**Solution:**

1. Rename the tag in InTouch if possible.
2. Define the tag in the OPC server manually.

### **Tag <tag name> could not be imported due to address length limitation**

---

**Error Type:**

Warning

**Possible Cause:**

The server limits the address length to 128 characters. InTouch tag names are used as the address, and therefore must not exceed this limit.

**Solution:**

This error is not expected with the present version of InTouch, 7.1, where names are limited to 32 characters. If this error occurs with later version of InTouch, rename the tag in InTouch if possible.

### **Imported tag name changed from <original name> to <new name>**

---

**Error Type:**

Warning

**Possible Cause:**

The InTouch tag name is not a valid OPC server tag name.

**Solution:**

The driver automatically modifies the name to make it compatible with the OPC server.

### **Tag browser error**

---

**Error Type:**

Warning

**Possible Cause:**

The driver's InTouch tag browser utility failed.

**Solution:**

Reinstall the driver to ensure that all necessary components are properly installed.

# Index

## A

Address <address> is out of range for the specified device or register 27

Address Descriptions 23

Advanced Channel Properties 8

Allow Sub Groups 12

Array size is out of range for address <address> 27

Array support is not available for the specified address 28

ASCII string 22

Automatic Tag Database Generation 19

## B

BCD 22

Boolean 22

## C

Channel Assignment 9

Channel Properties - General 6

Channel Properties — Write Optimizations 7

Connection to InTouch failed. Server runtime process must be configured for interactive mode 29

Create 13

## D

Data Collection 10

Data Type <type> is not valid for device address <address> 27

Data Types Description 22

Delete 12

Description 9

Device <device name> is not responding 28

Device address <address> contains a syntax error 26

Device address <address> is read only 27

Device Properties — General 8

Device Properties — Tag Generation 11

Diagnostics 7

Do Not Scan, Demand Poll Only 11

Driver 6, 9

Driver Setup 6

Duty Cycle 8

DWord 22

## **E**

Error Descriptions 26

External Dependencies 5

## **F**

Float 22

## **G**

Generate 12

## **I**

ID 9

IEEE-754 floating point 8

Imported tag name changed from <original name> to <new name> 30

Initial Updates from Cache 11

## **L**

LBCD 22

Long 22

Low resources (OS error: <error code>) 29

## **M**

Missing address 26

Mode 17

Model 9

**N**

Name 9

Non-Normalized Float Handling 8

**O**

On Device Startup 11

On Duplicate Tag 12

On Property Change 11

Optimization Method 7

Optimizing InTouch Client Driver Communications 20

Overview 4

Overwrite 12

**P**

Parent Group 12

**R**

Request All Data at Scan Rate 10

Request Data No Faster than Scan Rate 10

Respect Client-Specified Scan Rate 10

Respect Tag-Specified Scan Rate 11

**S**

Scan Mode 10

Short 22

Simulated 10

System Configuration 5

System Path 5

**T**

Tag <tag name> could not be imported due to address length limitation 30

Tag <tag name> could not be imported due to name conflict 30

Tag <tag name> could not be imported due to unresolved datatype 29

Tag browser error 30

Tag Generation 11

Tag import failed (OS error: <error code>) 29

Tag Import Settings 13

## **U**

Unable to write tag <address> on device <device name> 28

## **W**

Word 22

Write All Values for All Tags 7

Write Only Latest Value for All Tags 7

Write Only Latest Value for Non-Boolean Tags 7

Write Optimizations 7