C H A P T E R 1 3

I/O Communications

InTouch uses the Microsoft Dynamic Data Exchange (DDE), FastDDE, NetDDE and Wonderware SuiteLink protocols to communicate with other Windows programs, Wonderware I/O Servers and third-party I/O Server programs that are communicating with the real world.

Contents

Supported Communication Protocols

Dynamic Data Exchange (DDE) is a communication protocol developed by Microsoft to allow applications in the Windows environment to send/receive data and instructions to/from each other. It implements a client-server relationship between two concurrently running applications. The *server* application provides the data and accepts requests from any other application interested in its data. Requesting applications are called *clients*. Some applications such as InTouch and Microsoft Excel can simultaneously be both a *client* and a *server*.

FastDDE provides a means of packing many proprietary Wonderware DDE messages into a single Microsoft DDE message. This packing improves efficiency and performance by reducing the total number of DDE transactions required between *client* and *server*. Although Wonderware's FastDDE has extended the usefulness of DDE for our industry, this extension is being pushed to its performance constraints in distributed environments.

NetDDE extends the standard Windows DDE functionality to include communication over local area networks and through serial ports. Network extensions are available to allow DDE links between applications running on different computers connected via networks or modems. For example, NetDDE supports DDE between applications running on IBM PCs connected via LAN or modem and DDE-aware applications running on non-PC based platforms under operating environments such as VMS and UNIX.

Wonderware SuiteLink uses a TCP/IP based protocol. SuiteLink is designed specifically to meet industrial needs, such as data integrity, high-throughput, and easier diagnostics. This protocol standard is supported for both Microsoft Windows NT and Windows 2000.

SuiteLink is not a replacement for DDE, FastDDE, or NetDDE. Each connection between a client and a server depends on your network situation. SuiteLink was designed specifically for high speed industrial applications and provides the following features:

Value Time Quality (VTQ) places a time stamp and quality indicator on all data values delivered to VTQ-aware clients.

Extensive diagnostics of the data throughput, the server loading, computer resource consumption, and network transport are made accessible through the Microsoft Windows NT operating system performance monitor. This feature is critical for the scheme and maintenance of distributed industrial networks.

Consistent high data volumes can be maintained between applications regardless if the applications are on a single node or distributed over a large node count.

The network transport protocol is TCP/IP using Microsoft's standard Winsock interface.

Wonderware SuiteLink

Wonderware SuiteLink uses a TCP/IP based protocol. SuiteLink is designed specifically to meet industrial needs, such as data integrity, high-throughput, and easier diagnostics. This protocol standard is supported for both Microsoft Windows NT and Windows 2000.

To use the SuiteLink communication protocol

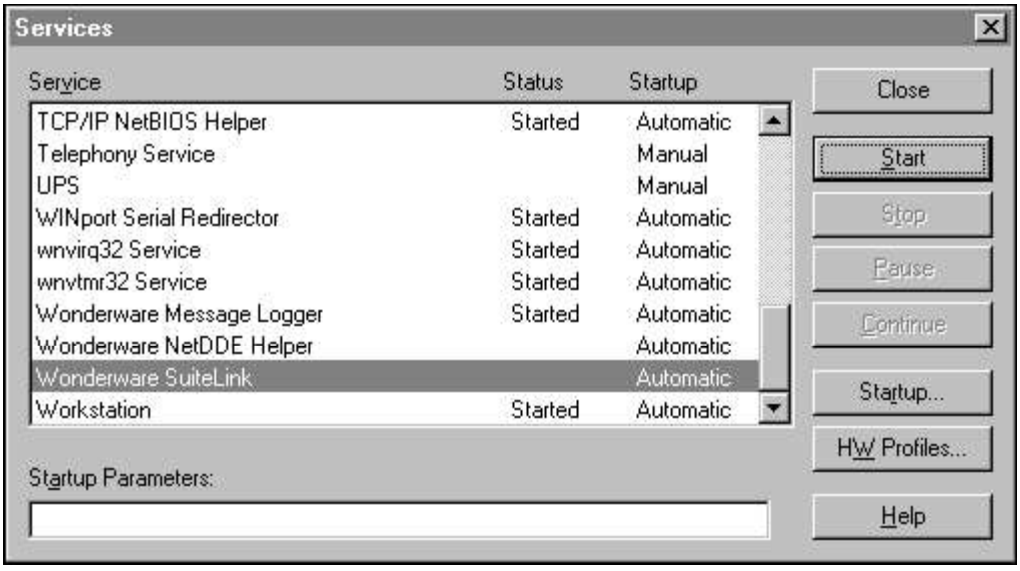
1. You must have Microsoft TCP/IP configured and working properly.
2. You must use computer names (**Node Names**) of no more than 15 characters.

   For more information on installing and configuring Microsoft TCP/IP, see your Microsoft Windows operating system's documentation.
3. Wonderware SuiteLink must be running as a service. If for some reason SuiteLink has been stopped, you will need to start it again. (SuiteLink is automatically installed when you install InTouch and by default, it is configured to startup automatically as an NT Service.)

   For more information on Windows NT services, see Appendix A, "Overview of the InTouch Windows NT Services."

To start SuiteLink as an NT Service

1. Open the Windows Control Panel.
2. Double-click **Services**. The **Services** dialog box appears.



3. Select **Wonderware SuiteLink**, and then click **Start**.
4. Click **Close**.

The InTouch I/O Address Convention

InTouch identifies an element of data in an I/O Server program by using a three-part naming convention that includes the *application name, topic name* and *item name*. To obtain data from another application the *client* program (InTouch) opens a channel to the *server* program by specifying these three items.

In order for InTouch to acquire a data value from another application, it must also know the name of the *application* providing the data value, the name of the *topic* within the application that contains the data value, and the name of the specific *item* within the *topic*. In addition, InTouch needs to know the data's type; discrete, integer, real (floating point), or message (string). This information determines the I/O type for the tagname when it is defined in the InTouch database. Now, when WindowViewer is running, it will automatically perform all of the actions required to acquire and maintain the value of this *item*.

For example, in the case of Excel, the *application name* is "Excel," the *topic name* is the name of the specific spreadsheet that contains the data and the *item name* is the identification of the cell on the spreadsheet to/from which the data is to be read/written.

The InTouch I/O Address

When another Windows application requests a data value from InTouch, it also must know the three I/O address items. The following describes the I/O address convention for InTouch:

VIEW *(application name)* identifies the InTouch runtime program that contains the data element.

TAGNAME *(topic name)* is the word always used when reading/writing to a tagname in the InTouch database.

ActualTagname *(item name)* is the actual tagname defined for the item in the InTouch Tagname Dictionary.

For example, to access a data value in InTouch from Excel, a DDE Remote Reference formula would be entered in the cell into which the data value is to be written:

```
=VIEW|TAGNAME!'ActualTagname'
```

Note If you are networking using Wonderware NetDDE, the *application name* portion of the I/O address must be prefixed with the remote node's name preceded by two backslashes and followed by one backslash. For example:

```
\\NodeName\VIEW|TAGNAME!'ActualTagname'
```

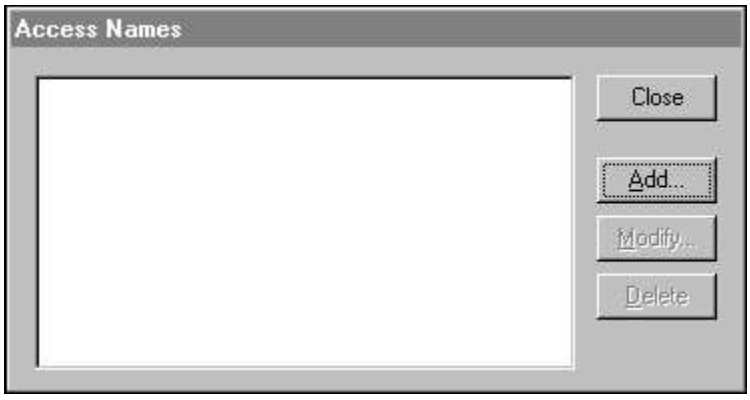
InTouch Access Names

When you create I/O type tagnames or remote tagname references, they must be associated with an Access Name. Access Names contain the information that is used to communicate with other I/O data sources including the node name, application name and topic name.

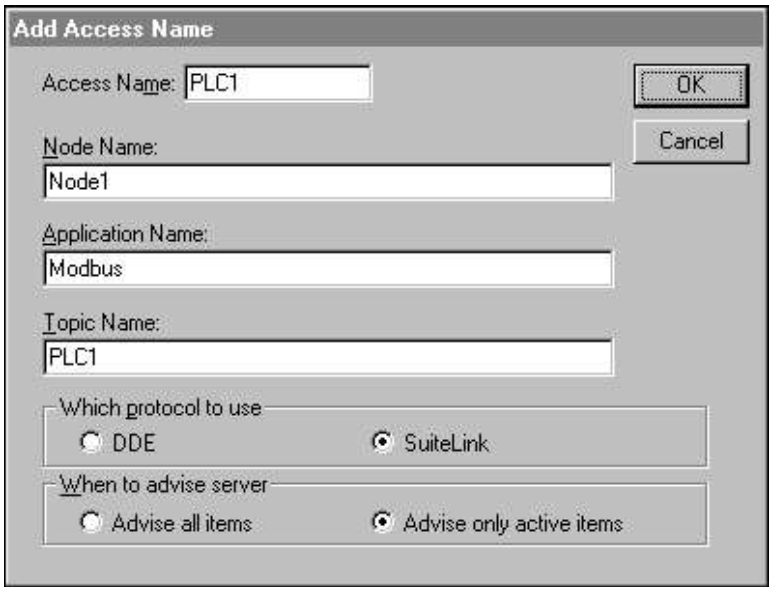
To create an Access Name

1. On the **Special** menu, click **Access Names**, or in the Application Explorer under **Configure**, double-click **Access Names**. The **Access Names** dialog box appears.

2. In the Application Explorer, you can right-click **Access Names**, and then click **Open**. You can also create Access Names while you are defining an I/O type tagname in the Tagname dictionary.



3. Click **Add**. The **Add Access Name** dialog box appears.



4. In the **Access Name** box, type the name you want InTouch to use to this Access Name. (For simplicity, use the same name that you will use for the *topic name* here.)

   InTouch uses Access Names to reference real-time I/O data. Each Access Name equates to an I/O address, which can contain a Node, Application, and Topic. In a distributed application, I/O references can be set up as global addresses to a network I/O Server or local addresses to a local I/O Server.

5. If the data resides in a network I/O Server, in the **Node Name** box, type the remote node's name.

6. In the **Application Name** box, type the actual program name for the I/O Server program from which the data value will be acquired. In this case the value is coming from the Wonderware Modbus I/O Server, therefore **MODBUS** is used. **Do not** enter the **.exe** extension portion of the program name.
7. In the **Topic Name** box, type the *topic name* you want to access. The **Topic Name** is an application-specific sub-group of data elements. In the case of data coming from a Wonderware I/O Server program, the *topic name* is the **exact** same name configured for the *topic* in the I/O Server program. When communicating with Microsoft Excel, the *topic name* must be the name given to the spreadsheet when it was saved. For example, Book1.xls.
8. Select the protocol that you are using.

   For more information, see "Supported Communication Protocols:"
9. Select the option you want to use to advise the server:

Option	Definition
Advise all items	Polls for all data whether or not it is in visible windows, alarmed, logged, trended or used in a script. Selecting this option will impact performance, therefore its use is not recommended.
Advise only active items	Polls only points in visible windows and points that are alarmed, logged, trended or used in any script. **Note** A touch pushbutton action script will not be polled unless it appears in a visible window.

10. Click **OK** to accept the new Access Name and close the dialog box. The **Access Names** dialog box reappears displaying the new Access Name selected in the list.

11. Click **Close** to close the dialog box and return to your tagname definition.

To modify or delete an Access Name

1. On the **Special** menu, click **Access Names**, or in the Application Explorer under **Configure**, double-click **Access Names**. The **Access Names** dialog box appears.
2. In the Application Explorer, you can right-click **Access Names**, and then click **Open**.

3. To change an Access Name's definition, select it in the list, and then click **Modify**. The **Modify Access Name** dialog box appear. Make your required changes, and then click **OK**. The **Access Names** dialog box reappears. Click **Close** or repeat this procedure if you need to modify other defined Access Names.
4. To delete an Access Name, select it in the list, and then click **Delete**. A message box will appear asking you to confirm the deletion of the selected Access Name. Click **Yes** to delete it or click **No** to cancel the deletion. Click **Close** or repeat this procedure if you need to delete other defined Access Names.

Note Access Names that are associated with tagnames cannot be deleted.

Defining an I/O Item in InTouch

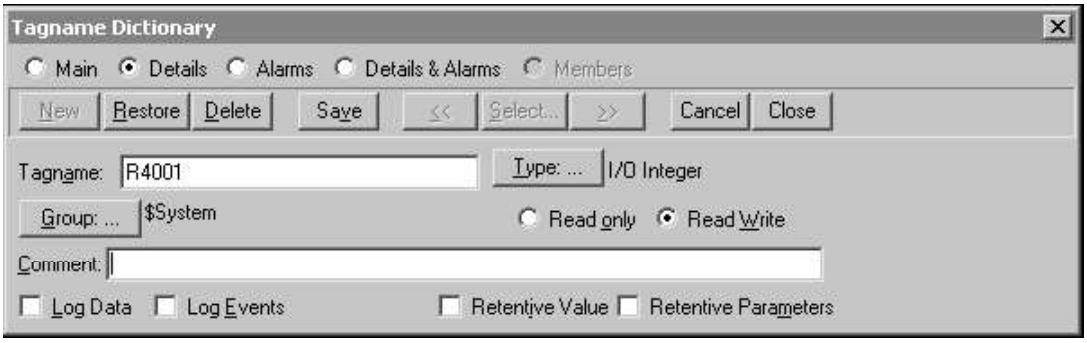
InTouch can receive data from other local or remote Windows applications when you define I/O type tagnames in the Tagname Dictionary. Each I/O type tagname references a valid *item* in the I/O Server program.

For more information on distributed applications see, Chapter 5, "Building a Distributed Application."

To define an I/O type tagname

1. On the **Special** menu, click **Tagname Dictionary**, or in the Application Explorer double-click **Tagname Dictionary**. The **Tagname Dictionary** dialog box appears.
2. Click **New**. The **Tagname** box clears.

Tip If you right-click any of the text entry boxes in any of the Tagname Dictionary dialog boxes, a menu will appear displaying the commands that you can apply to the selected text.



The first time you access the Tagname Dictionary, the definition for the internal system tagname **$AccessLevel** is displayed. Once you define tagnames in the Tagname Dictionary, when you access it again, the last edited tagname's definition is displayed.

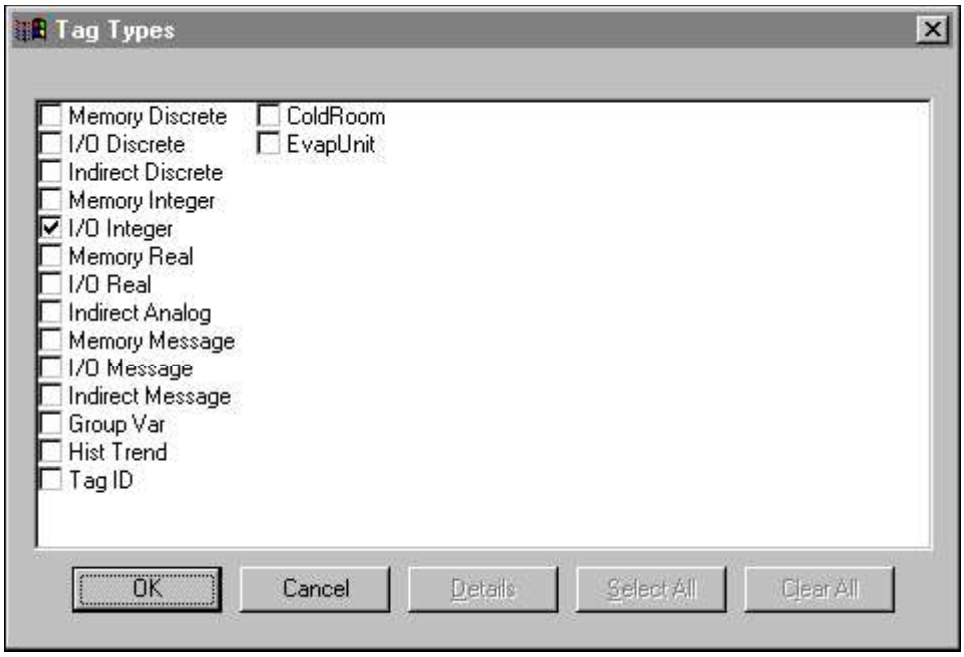
3. In the **Tagname** box, type the name you want to use for the new tagname.

   Tagnames can be up to 32 characters long and must begin with an alpha character (A-Z or a-z). The remaining characters can be A-Z, a-z, 0-9, !, @, -, ?, #, $, %, _, \ and &.

   Tagnames are also auto-indexed. For example, if you enter and save tagname R4001, and then click **New**, the tagname will automatically be indexed to R4002. If an tagname contains a character separating numbers, it is auto-indexed by the first whole number InTouch finds. For example, N7-0 would be indexed as N7-1. Positive changes only are permitted. For example, R4002 to R4003, R4003 to R4004 and so on.

   You cannot use the word **RetVal** for a tagname. This is a reserved word. If you attempt to use this word, and then try to edit a QuickFunction you will receive the error message "A variable cannot have that name. Tag exists."

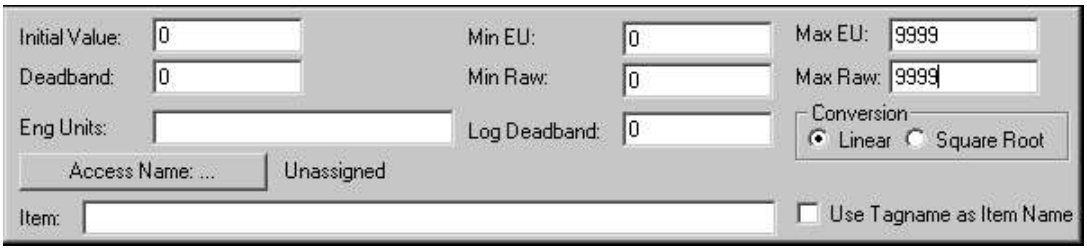
4. Click **Type**. The **Choose tagname type** dialog box appears.



5. Select the I/O type for the tagname as follows:

Tag Type	Input/Output Value
I/O Discrete	True, On, Yes (1) or False, Off, No (0)
I/O Integer	Whole number
I/O Real	Floating decimal point
I/O Message	String

6. Once you select the I/O type, click **OK**. The respective "details" dialog box for the selected I/O type appear. For example, if you select I/O Integer, the following dialog box appears.



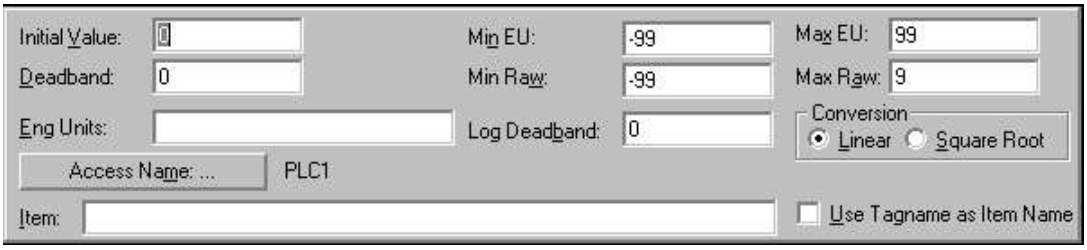
Tip If the "Details" dialog box does not appear, click **Details** at the top of the screen.

7. Specify all the required details for defining the *item*.

8. Click **Access Name**. The **Access Names** dialog box appears.

9. Double-click the Access Name that you want to use in the list or select it, and then click **Close**.

   The Access Name that you selected (now associated with this tagname definition) appears adjacent to the **Access Name** button in the details dialog box. For example:



10. In the **Item** box, type the *item name* for the data value in the I/O Server program.

> **Note** It is important to understand that the "tagname" is the name used within InTouch to refer to a data value. The **Item** is the name used by a remote Windows application to refer to the same value. These names do not have to be the same but, it is recommended when applicable to use the same names. Also, if the **Item** is a cell in Excel, it must be specified either by its defined name in Excel, or by its row/column identification. For example, R1C1.

11. Click **Close**.

    For more information on defining I/O tagnames, see Chapter 6, "Tagname Dictionary."

Monitoring the Status of an I/O Conversation

WindowViewer supports a built-in *topic name* called **IOStatus** (**DDEStatus** in versions prior to InTouch 7.0) that can be used to monitor the status of specific I/O conversations.

Using IOStatus Topic Name

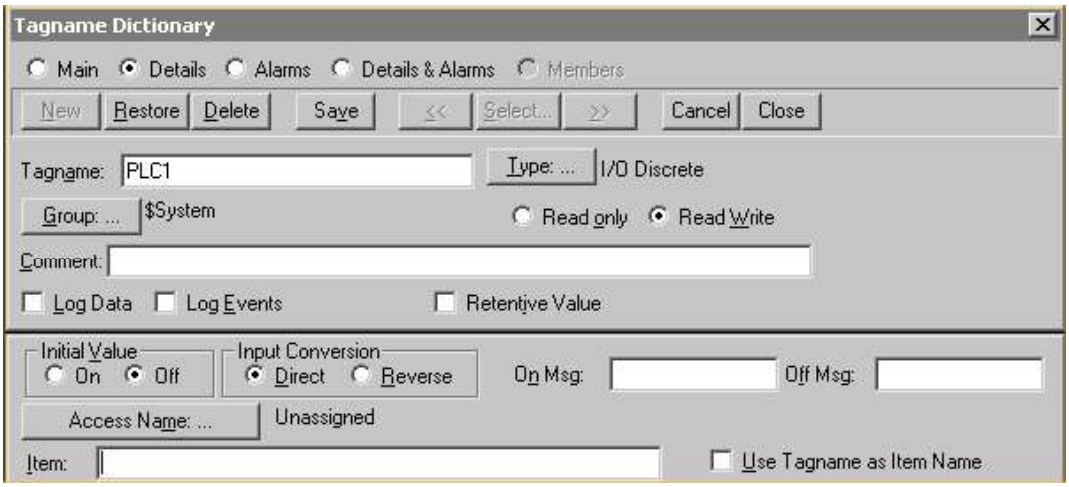
Let's assume that WindowViewer (View) is communicating with the Wonderware Simulate I/O Server to a PLC that has been defined in the I/O Server with **PLC1** for its *topic name*.

(Simulate is a generic Wonderware I/O Server that is intended to be used as a training tool. Simulate is included with FactorySuite.)

To monitor the status of I/O communications

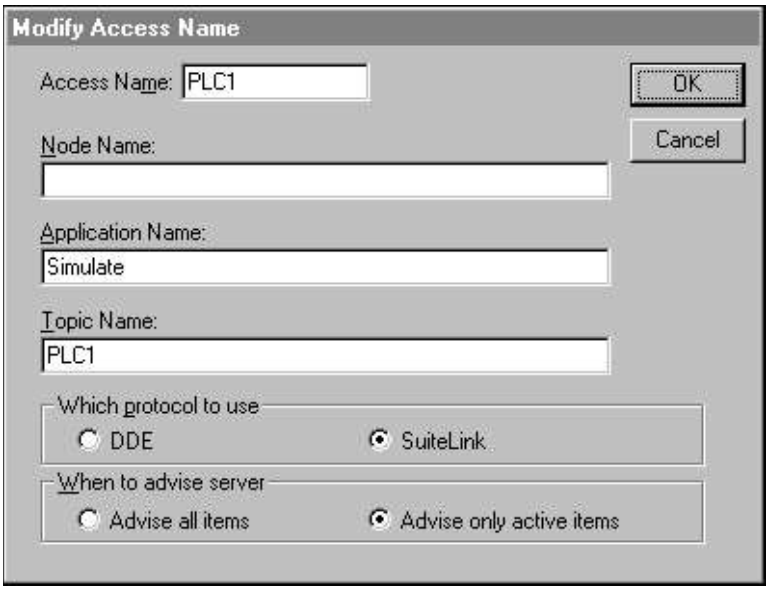
1. On the **Special** menu, click **Tagname Dictionary**, or in the Application Explorer, double-click **Tagname Dictionary**. The **Tagname Dictionary** dialog box appears.
2. Create an **I/O Discrete** type tagname. (In this example, for simplicity, we will make our tagname the same as the *topic name* that we want to monitor.)

   When you are monitoring a I/O conversation using **IOStatus**, you must define at least one I/O type tagname to the Access Name being monitored.



3. Click **Access Name** to assign the tagname to an Access Name definition that defines **IOStatus** for its *topic name*. The **Access Name** dialog box appears.

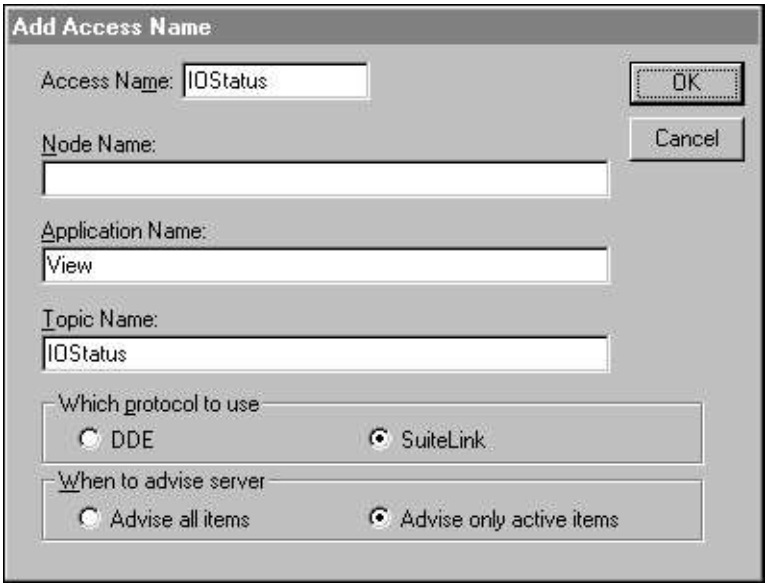
Notice that an Access Name definition called **PLC1** (the topic we want to monitor) currently exists. To be sure that this is the correct Access Name (its **Topic Name** is **PLC1**), click **Modify** to view the definition.



Finding the Access Name containing the right *topic name* in this example was easy because we kept the tagname and the **Topic Name** the same.

4. Click **Cancel** to close the dialog box and return to the initial **Access Name Definition** dialog box.

5. Click **Add**. The **Add Access Name** dialog box appears.



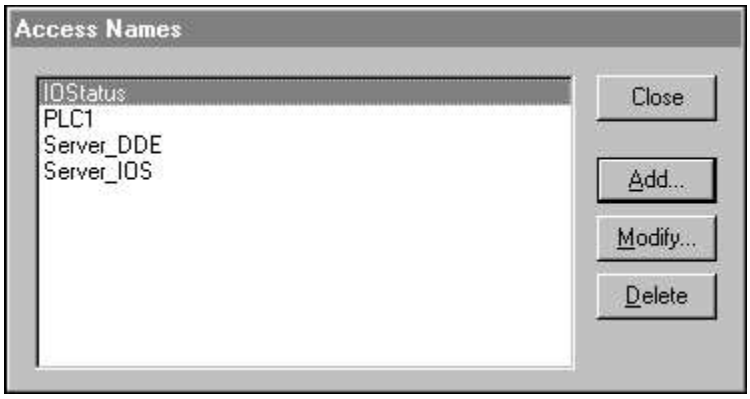
6. In the **Access Name** box, type **IOStatus**.

   Since you are going to monitor the status in WindowViewer, in the **Application Name** box, type "View."

7. In the **Topic Name** box, type the InTouch internal *topic*, **IOStatus**.

8. Select **Advise only active items**.

9. Click **OK** to close the dialog box. The initial **Access Name Definition** dialog box reappears displaying your new **Access Name**, **IOStatus**, in the list:



10. Click **Close** to close the dialog box and associate the new **Access Name** with your **I/O Discrete** tagname:

11. In the **Item** box, type the **Access Name** for the actual **Topic Name** that you want to monitor. In this case, **PLC1**.

    Since your tagname is the same as the **Topic Name**, you can select **Use Tagname as Item Name** and automatically enter it for the **Item**.

Note When using the built-in topic **IOStatus** (**DDEStatus** prior to InTouch Version 7.0) to monitor an I/O conversation, the name you type in the **Access Name** box is always also used for the **Item**.

Using IOStatus Topic Name in Excel

Excel can also be used to perform this same type of monitoring by entering the same information in a formula in a spreadsheet cell. For example, to monitor the same topic as above, the following would be entered:

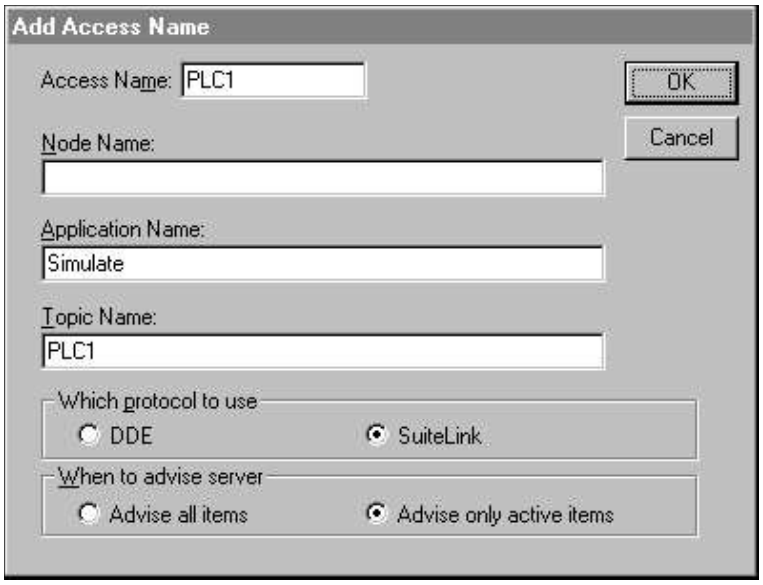
```
=view|IOStatus!'PLC1'
```

Monitoring I/O Server Communications Status

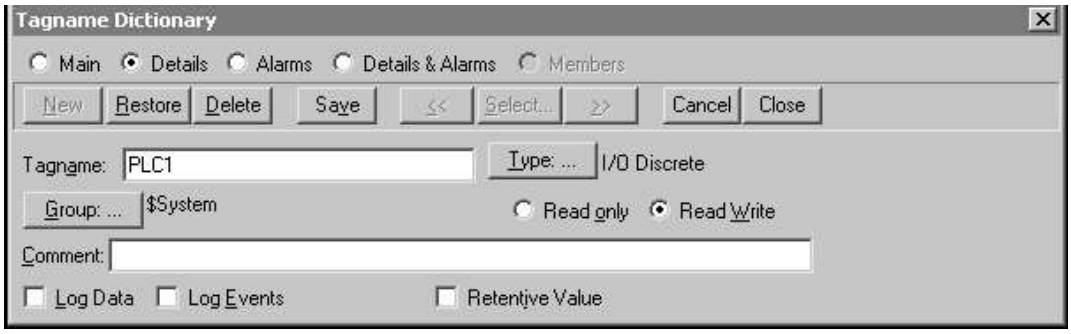
For each *topic name* being used, there is a built-in discrete *item*, **Status**, that you can use to monitor the state of your communications with the I/O Servers program. **Status** is set to "0" when communications with the device fails (cable disconnected, PLC is powered down, and so on.) and set to "1" when communications is successful.

Note When you monitor the status of a topic using the Status item, there must at least one I/O point active to the topic being monitored.

From InTouch, you can read the state of the server communications by defining a tagname and associating it with the *topic* configured for the device by using the word **Status** as the *item name*. For example, if WindowViewer is communicating with a PLC using the Wonderware Simulate I/O Server, the Access Name definition would be:



To monitor the status of all communication to the *topic*, PLC1, you would create the following tagname definition:



For more information on troubleshooting I/O communications see your Wonderware I/O Server's User Guide.

Tip From Excel, you can read the status of the PLC communications by entering the following formula in a cell:

=SIMULATE|PLC1!'STATUS'

Monitoring Multiple Input Device Status

This section describes how you can display the status of an object using multiple inputs.

Example 1

In this example, the status of a spring-return motorized valve using two inputs is being viewed. The two inputs represent a pair of limit switches installed on the valve. One input is only on when the valve is in the open position and off when the valve is in travel or closed. The other input is only on when the valve is in the closed position and off when the valve is in the travel or open position. A truth table of the inputs would appear as follows:

Valve Truth Table 1

Input #1 (opened)	Input #2 (closed)	Valve Position	Result
1	0	Opened	1 + 0 = **1**
0	1	Closed	0 + 1 = **1**
0	0	InTravel	0 + 0 = **0**
1	1	InValid Position	1 + 1 = **1**

0 = OFF 1 = ON

The inputs can be weighed by multiplying the closed input by 2. The results of the valve positions then change to the following:

Valve Truth Table 2
(Input #2 x2)

Input #1 (opened)	Input #2 (closed)	Valve Position	Result
1	0 x 2 = 0	Opened	1 + 0 = **1**
0	1 x 2 = 2	Closed	0 + 2 = **2**
0	0 x 2 = 0	InTravel	0 + 0 = **0**
1	1 x 2 = 2	InValid Position	1 + 2 = **3**

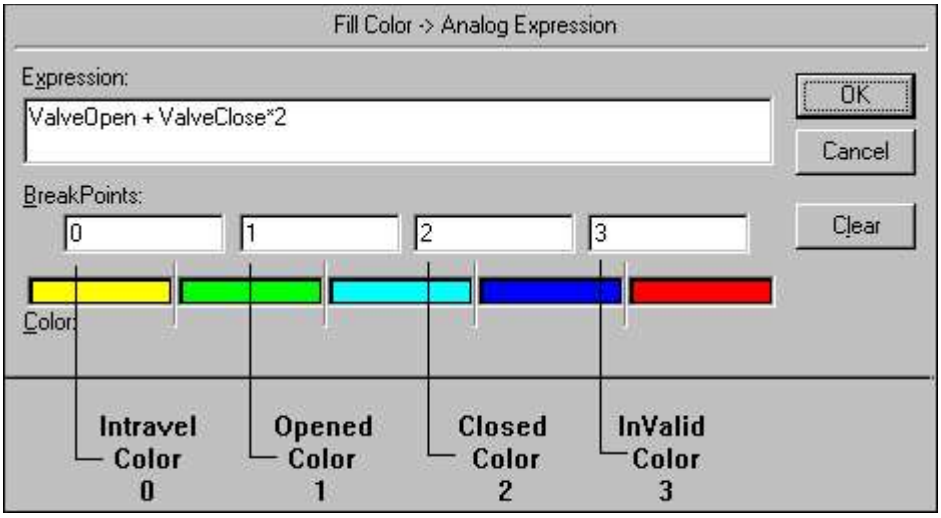
0 = OFF 1 = ON

Note The invalid position can be used to show a defective limit switch.

Now that there is a significant numerical difference between the valve positions, a **Fill Color - Analog** animation link can be used to display the valve status.

For more information on creating animation links, see Chapter 7, "Creating Animation Links."

Two I/O Discrete tagnames are created. One for the valve open input and one for the valve closed input. For example, **ValveOpen** and **ValveClosed**. An object is created to display the valve status. This object is assigned to a **Fill Color - Analog** animation link with the following properties:



Example 2

In this example, one more input has been added to the existing two. This new input is the actual output to the valve that causes it to open. The new input will be on when the valve is opening or open, and off when the valve is closing or closed. The new truth table appears as follows:

Valve Truth Table 3

Input #1 (opened)	Input #2 (closed)	Input #3 (open)	Valve Position	Result
0	0	1	Opening	0 + 0 + 1 = **1**
1	0	1	Opened	1 + 0 + 1 = **2**
0	0	0	Closing	0 + 0 + 0 = **0**
0	1	0	Closed	0 + 1 + 0 = **1**

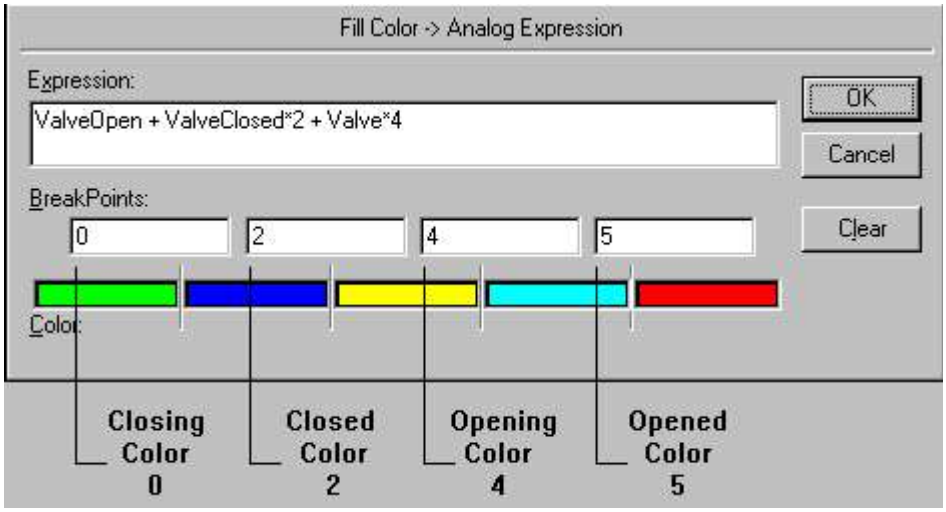
0 = OFF 1 = ON

Once again the inputs are weighed. As previously explained, the closed input will be multiplied by 2 and the new input will be multiplied by 4 for the following results:

Valve Truth Table 4
(Input #2 x 2 Input #3 x 4)

Input #1 (opened)	Input #2 (closed)	Input #3 (open)	Valve Position	Result
0	0 x 2 = 0	1 x 4 = 4	Opening	0 + 0 + 4 = **4**
1	0 x 2 = 0	1 x 4 = 4	Opened	1 + 0 + 4 = **5**
0	0 x 2 = 0	0 x 4 = 0	Closing	0 + 0 + 0 = **0**
0	1 x 2 = 2	0 x 4 = 0	Closed	0 + 2 + 0 = **2**
0 = OFF	1 = ON			

Another I/O Discrete tagname (**Valve**) is created for the new open input and assigned to a **Fill Color - Analog** animation link with the following properties:



Using this method, additional inputs can be used. The fourth input would be multiplied by 8, the fifth by 16, and so on.