

## Tech Note 154

# Configuring InTouch To Query Data From IndustrialSQL Server

---

All Tech Notes and KBCD documents and software are provided "as is" without warranty of any kind. See the [Terms of Use](#) for more information.

Topic#: 000816  
Created: September 1999

Under FactorySuite™ 2000, InTouch™ has the ability to query and display data logged in IndustrialSQL™ Server in one of three ways:

- Using the InSQL ActiveTrend ActiveX object
- Using the InTouch Historical Trend wizard with the distributed history configuration
- Using a direct query through the SQLAccess Manager option

This *Tech Note* provides installation and configuration guidelines for InTouch to query and display data from IndustrialSQL Server for each of these three methods.

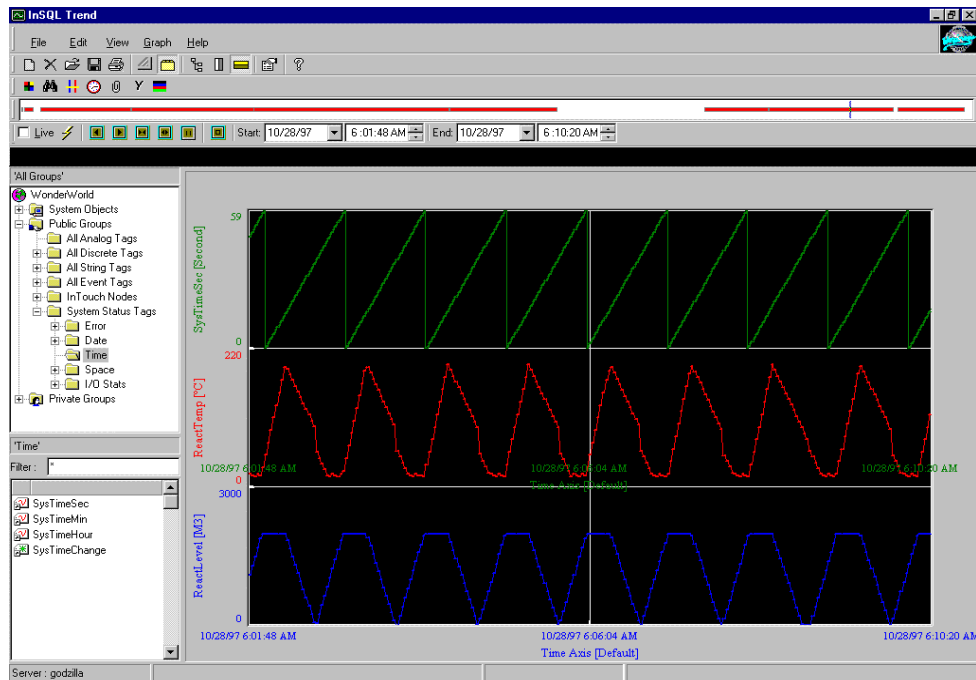
### Method 1: Using the InSQL ActiveTrend ActiveX control

The ActiveTrend ActiveX control is the easiest way to connect to the InSQL database by providing simple tag and time selection. The ActiveTrend data display can be pre-configured in WindowMaker to show specific information in a pre-set format. It may also be dynamically re-configured in WindowViewer by addressing the properties and methods via scripting. The InSQL ActiveTrend object provides the ability to select InSQL tags and view logged data up to the current time in either a realtime "live" mode or in a "history" mode, as shown in figure 1.

### Installing the ActiveTrend Object

1. Install the ActiveTrend object from the first FactorySuite 2000 CD-ROM and select **IndustrialSQL/IndustrialSQL Client Tools**.
2. Start up WindowMaker and select **Special / Configure / Wizard/ActiveX Installation**. Click the **ActiveX Control Installation** tab.
3. Select **Wonderware ActiveTrend** from the Available ActiveX Controls list and click the **Install** button.

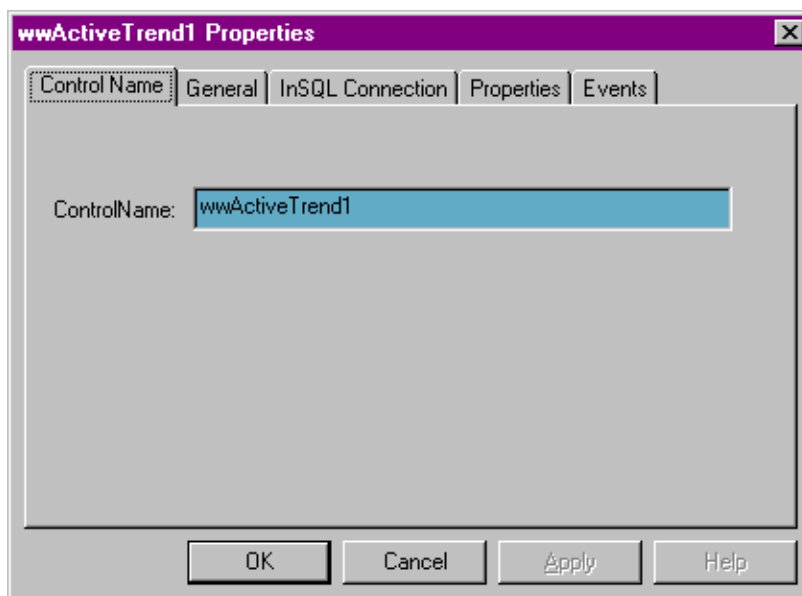
The ActiveTrend object will now appear in the Installed ActiveX controls list.



**FIGURE 1. EXAMPLE OF THE ACTIVETREND ACTIVEX OBJECT**

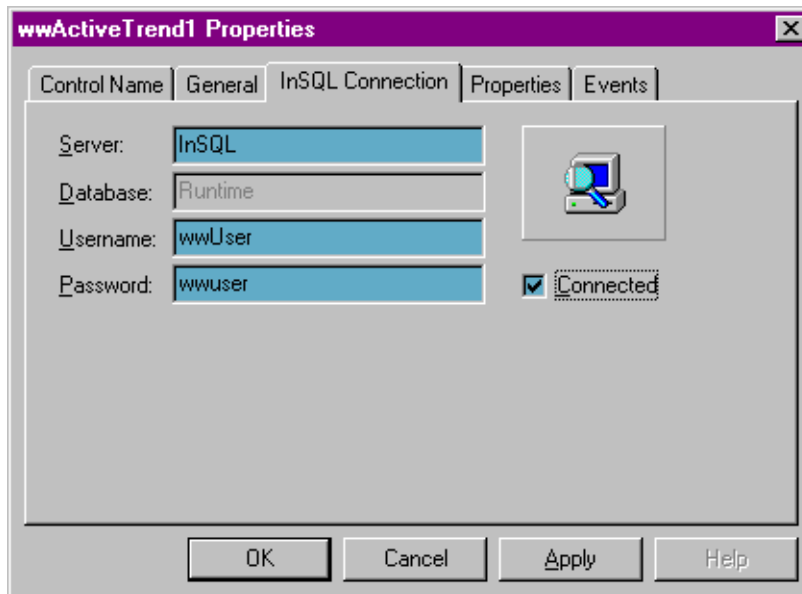
## Using the ActiveTrend Control

1. Click on the Wizard icon on the WindowMaker/ActiveX toolbar and drag-and-drop the ActiveTrend object from the Wizard Library onto an application window.
2. Configure the ActiveTrend object by double-clicking on the object. The ActiveTrend *Properties* dialog box will appear as shown in figure 2. Type in a unique **ControlName** for the object. This ControlName will be used later on to reference the object's properties and methods through an InTouch script during runtime.



**FIGURE 2. CONFIGURING THE ACTIVE TREND OBJECT**

The ActiveTrend *Properties* dialog box also provides a way to connect the ActiveTrend object to the IndustrialSQL Server database by clicking the **InSQL Connection** tab as shown in figure 3. The object may be connected from both WindowMaker and WindowViewer. Database connectivity requires a valid node name for the Server (example, *InSQL*) and a valid Username and Password for the IndustrialSQL Server Runtime database.

**FIGURE 3. CONFIGURING THE ACTIVE TREND OBJECT TO CONNECT TO AN INSQL DATABASE**

The ActiveTrend object connects to the IndustrialSQL Server database by using the SQL Server Client alias, which was defined during the InSQL installation to use a Named Pipes connection. (Client aliases may be viewed by selecting from the Windows NT Start menu **Programs/ Microsoft SQL Server 6.5-SQL Client Configuration Utility**, then click the **Advanced** tab.)

If the ActiveTrend object will connect to an InSQL database on the *same* computer, enter the "InSQL" alias or the computer node name in the "Server" field of the *Properties* dialog box. If the ActiveTrend object will connect to an InSQL database on a remote computer, enter the node name of the IndustrialSQL Server computer in the "Server" field.

The ActiveTrend object has a wealth of features, including: automatic connection to the database, history and live displays, define and save curve sets, a tag browser, and view and hide the toolbars at runtime. For more information on these and other features, see the *Wonderware IndustrialSQL Client Tools Guide*, Chapter 9.

## Notes on Using the ActiveTrend Control

1. More than one instance of the ActiveTrend object may be used in an InTouch application, but only *one* instance at a time may be connected to the InSQL database.
2. The ActiveTrend object may be prevented from connecting to the InSQL database if Productivity Pack wizards are installed. Objects known to interfere include: 16-Pen Trend, Document Viewer, Event Chart, OLE2 and SQLGrid. To resolve this issue, uninstall these Productivity Pack wizards from the WindowMaker wizard library and retry the database connection. Then reinstall any wizard to be used and verify that the ActiveTrend object can

still connect.

- ActiveX Events are not supported for the ActiveTrend object.

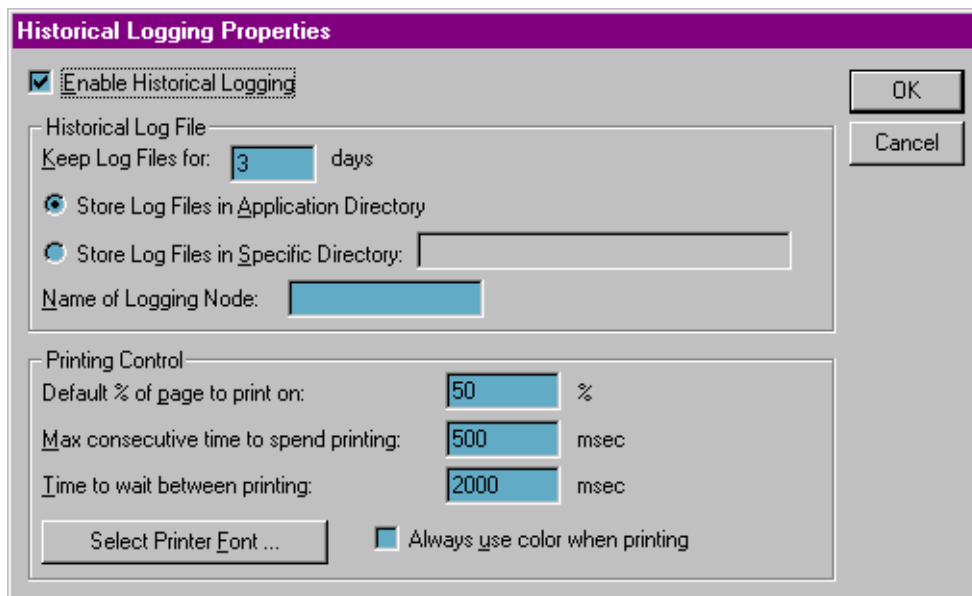
## Method 2: Using the InTouch Historical Trend Wizard

IndustrialSQL Server data may be displayed in InTouch by using the Historical Trend wizard. InSQL data may be displayed simultaneously with data logged by InTouch. All the functionality needed to view the InSQL logged data in the InTouch Historical Trend wizard is included when InTouch is installed.

The InTouch Historical Trend wizard will connect to the InSQL database through a Microsoft SQL Server client alias, which is automatically installed and configured (under Common Components) when you install InTouch. Specifying the node name of the IndustrialSQL Server PC where a dialog requests "Servername" or "Data Source" should always provide the connection to the InSQL database.

## Procedure

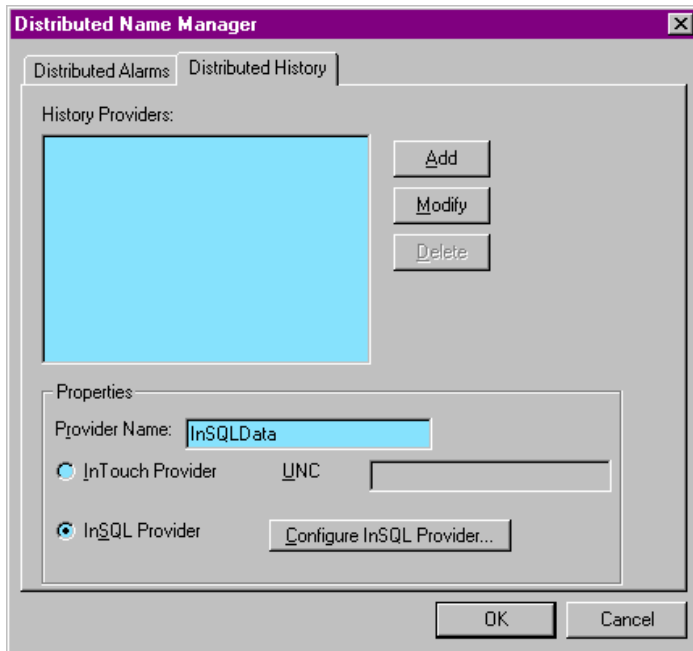
- Open an InTouch application in WindowMaker. WindowMaker must create two files in the InTouch application directory: **Dhist.cfg** and **Histprv.ini**. To do this, select **Special/Configure/Historical Logging**. The *Historical Logging Properties* dialog box will appear as shown in figure 4.



**FIGURE 4. CONFIGURING THE ACTIVETREND OBJECT TO CONNECT TO AN INSQL DATABASE**

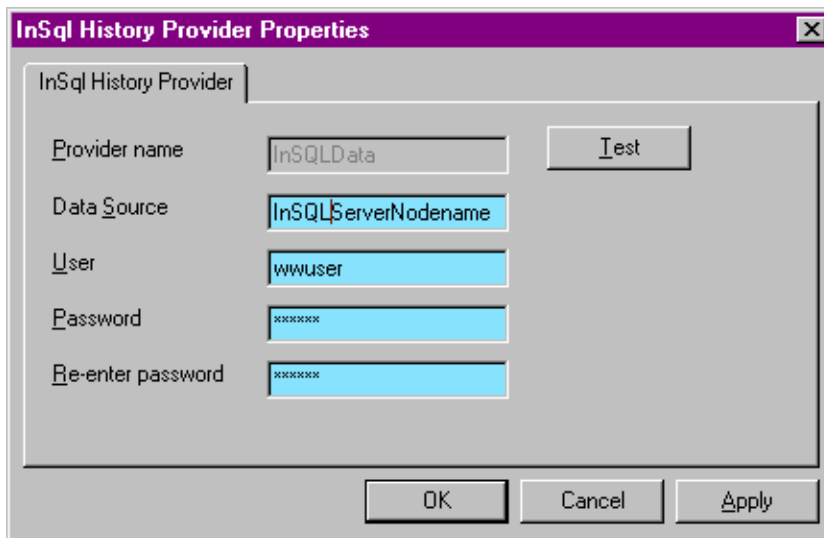
Select the **Enable Logging** checkbox, then click OK. The two files will then be created by WindowMaker. If InTouch logging is not desired, the **Enable Logging** checkbox may be deselected and these two files will remain.

- Define the Distributed History Provider name by selecting **Special/Configure/Distributed Name Manager**. The *Distributed Name Manager* dialog box will appear as shown in figure 5.



**FIGURE 5. DEFINING THE DISTRIBUTED HISTORY PROVIDER NAME**

Click on the **Distributed History** tab. Type in a unique name for the InSQL database connection in the **Provider Name** listbox (example, "InSQLData"). Select **InSQL Provider**, then click the **Configure InSQL Provider** button. The *InSql History Provider Properties* dialog box will appear as shown in figure 6.



**FIGURE 6. DEFINING THE DISTRIBUTED HISTORY PROVIDER NAME**

3. Type in the node name of the computer for the InSQL Server in the **Data Source** listbox. Type in a valid InSQL userid and password in the **User** and **Password** listboxes, respectively. (The wwuser account, along with its password, has sufficient database privileges for InTouch to read data from the InSQL database.

Test the InSQL database connection by clicking the **Test** button. Then click **OK**.

4. View the InSQL data during runtime in the InTouch Historical Trend object. InTouch will automatically connect to the InSQL database when the object's window is opened. InSQL tags may be assigned to pens in the Historical Trend object by preceding the tagname with the History Provider name and a dot (.) (example, "InSQLData.tagname"). Pen assignments may also be made using function calls in an InTouch script.

The InTouch demonstration application accompanying this *Tech Note* (available on the Wonderware Knowledge Base at [www.wonderware.com/support](http://www.wonderware.com/support)) uses the following Button script to assign an InSQL tagname to a pen in the Historical Trend object:

```
IF InSQLTagName == "_None" THEN
HTSetPenName( "HistTrend1", 5, "ReactLevel" );
HistTrend1.Pen5 = None;
ELSE
HTSetPenName( "HistTrend1", 5, "InSQLData." + InSQLTagName );
ENDIF;
```

This script will also unassign a tagname from the pen if "\_None" has been selected for the tagname from the Listbox.

**Note** To unassign the pen after a distributed tagname has been assigned to it, you must first assign the pen to a local InTouch tagname, then reassign it to the tagname None. This tag must be defined in the InTouch application as tag type TagID.

### Method 3: Using the InTouch SQLAccess Manager Option

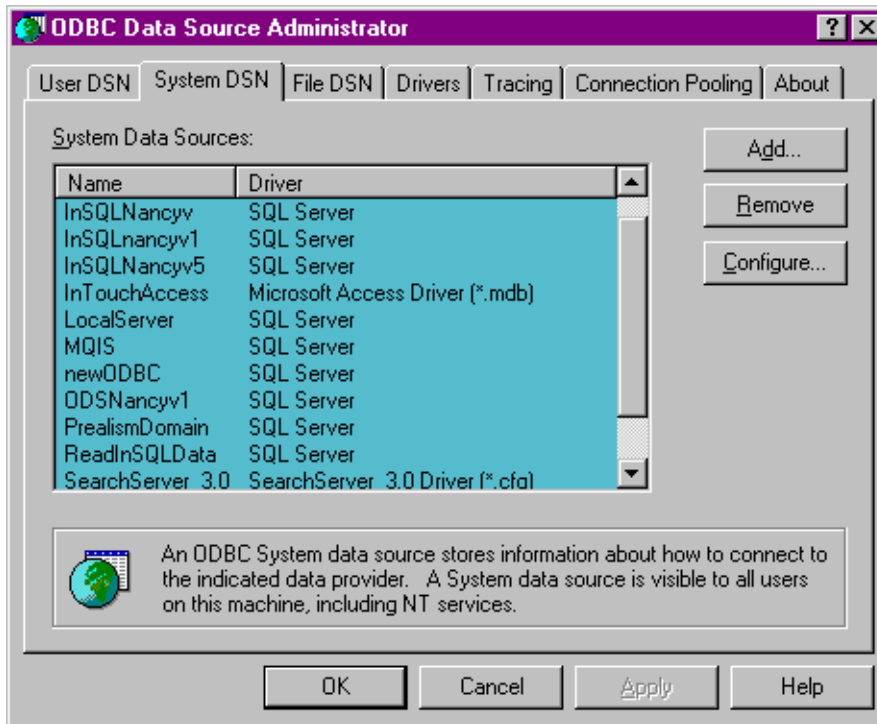
SQLAccess Manager script function queries provide the most flexible means of retrieving data from the InSQL database. This method also requires the most effort as InTouch scripts must be written to retrieve the data and window objects must be created to display the query results. There are 4 basic steps:

1. Create a BindList to associate the InSQL Runtime database table columns to InTouch tags.
2. Connect to the InSQL database via an ODBC DataSourceName.
3. Execute the desired query using SELECT statements.
4. Display the query results in a window using InTouch tags.

**Note** A sample InTouch application shown in figure 10 is used to illustrate how a typical window would be configured to use the SQLAccess Manager option. Though you can still follow this method without needing the application, the application is available on the [Wonderware Knowledge Base](#). The Knowledge Base is available exclusively to Comprehensive Support subscribers. Contact your local Wonderware distributor to learn more about our Comprehensive Support program.

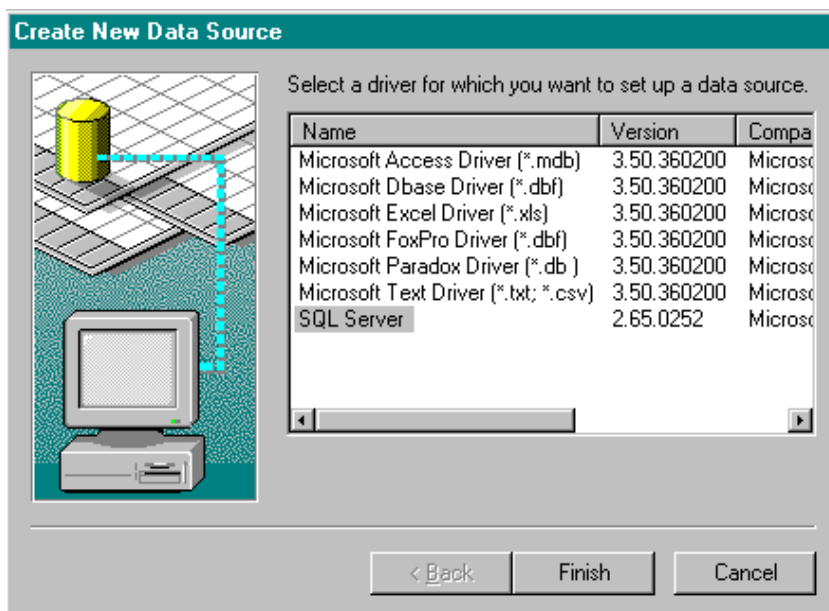
### Establishing the ODBC Connection to the InSQL Database

1. Log onto the client computer with Administrator privileges and bring up the Control Panel by selecting **Settings/Control Panel** from the Start menu. Double-click on the **ODBC Admin** program icon. The *ODBC Data Source Administrator* dialog box will appear as shown in figure 7.



**FIGURE 7. ADDING A ODBC CONNECTION**

2. Click either the **User DSN** or **System DSN** tab and click on the **Add** button. The *Create New Data Source* dialog box will appear as shown in figure 8.



**FIGURE 8. SELECTING SQL SERVER FOR THE ODBC DATA SOURCE**

Select **SQL Server** as shown in the figure and click **Finish**. The *ODBC SQL Server Setup* dialog box will appear as shown in figure 9.

3. Enter a unique name in the **Data Source Name** field as shown in figure 9. In the **Server** field, enter either "InSQL" or the node name of the computer where the InSQL database is running. Enter "Runtime" for the **Database Name**. Make sure the option **Generate Stored Procedure for Prepared Statement** is *not* selected as shown in the figure.

For more information on establishing ODBC connections to SQL Server, see the *Wonderware FactorySuite System Administrator's Guide*, Chapter 4 "Configuring ODBC."

**FIGURE 9. DEFINING THE ODBC DSN FOR THE INSQL DATABASE**

## Configuring a SQLAccess Manager BindList

The BindList is used to associate values retrieved from a database with InTouch tags so that the data may be viewed in a window or referenced in a script. To create a BindList:

1. Select **Special/SQLAccess Manager/BindList**, then click **New**.
2. Enter a BindList name and then type a tag name or browse to the tag list to specify an existing InTouch tag. Next enter the name of the database column to be associated with this InTouch tag.

The BindList may have as many pairs of Tagname/Column associations as necessary to retrieve and display the desired data.

**Note** The tags must be defined prior to creating the BindList. No hot link exists between the BindList definition window and the tagname dictionary.

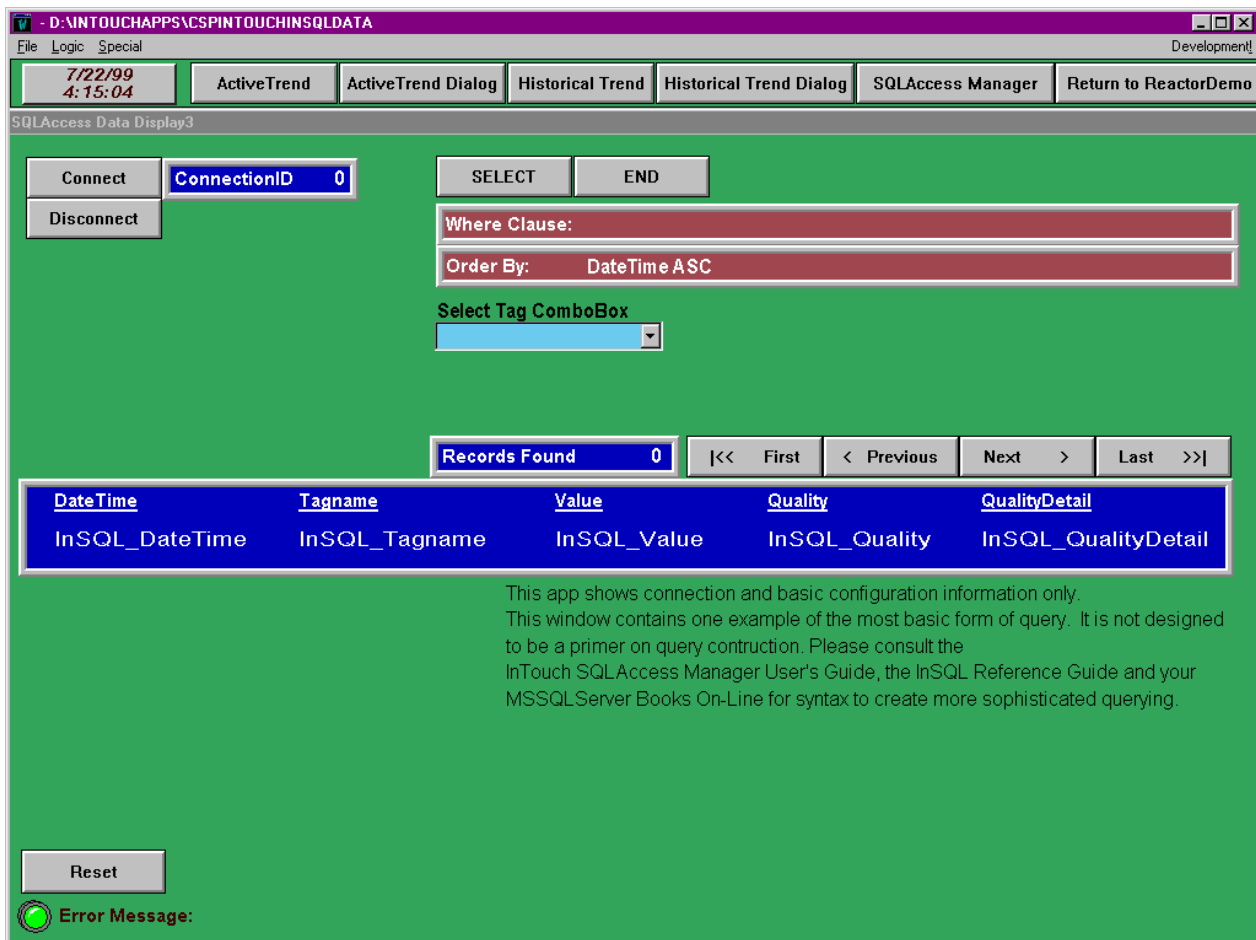
## Querying With the SQLAccess Manager Script Functions



Once the preliminary tasks of creating the ODBC DataSourceName and building the BindList are accomplished, getting the data is as simple as the following:

1. Connect to the InSQL database via and ODBC DataSourceName.
2. Execute the desired query using SELECT statements.
3. Display the query results in a window using InTouch tags and graphic objects.

Figure 10 shows a sample application window that displays some retrieved data. Each object has the necessary scripting to achieve data retrieval. Table 1 at the end of this *Tech Note* lists the tag definitions used in the scripting.



**FIGURE 10. SAMPLE INTouch APPLICATION WINDOW DISPLAYING RETRIEVED INSQL DATA**

Based on the sample application shown in figure 10, here's the Window script that's used:

#### Window "On Show" script

```
TableName = "AnalogHistory";
BindList = "InSQL_Data";
```

For the **Connect** button, here's the Button script used:

```
ReturnCode = SQLConnect( ConnectionId, "DSN=ReadInSQLData;UID=wwuser;PWD=wwuser" );
```

For the **Disconnect** button, here's the Button script used:

```
ReturnCode = SQLDisconnect( ConnectionId );
ConnectionId = 0;
```

The **Select Tag ComboBox** sets the value of the Message tag **InSQLTagNameWhereExp** from a list of tagnames contained in a text file to be used in the Select statement.

For the **Select** button, here's the Button script used:

```
ReturnCode = SQLSelect( ConnectionId, TableName, BindList, WhereExpr, OrderByExpr );
SQLFirst( ConnectionId );
RowsFound = SQLNumRows( ConnectionId );
```

For the **End** button, here's the Button script used:

```
ReturnCode = SQLEnd( ConnectionId );
```

The **First**, **Previous**, **Next** and **Last** buttons are used to navigate through the resulting set of data returned from a query. Here are the Button scripts used for each button:

```
ReturnCode = SQLFirst( ConnectionId );
ReturnCode = SQLPrev( ConnectionId );
ReturnCode = SQLNext( ConnectionId );
ReturnCode = SQLLast( ConnectionId );
```

Here are the tagname definitions for the tags in the sample InTouch application shown in figure 10. to store configuration data and display returned values in the window

**TABLE 1. TAGNAME DEFINITIONS FOR SAMPLE INTOUCH APPLICATION**

<b>Query Tagnames</b>	<b>Tag Type</b>	<b>Description</b>
ConnectionId	Memory Message	InSQL database connection Id number.
BindList	Memory Message	Name of BindList that pairs the database column with the InTouch tagnames.
InSQLTagNameWhereExp	Memory Message	WHERE expression for the SELECT statement.
TableName	Memory Message	Name of the runtime database table to query.
<b>Display Tagnames</b>	<b>Tag Type</b>	<b>Description</b>
InSQL_DateTime	Memory Message	Display of the date and time for the tag value returned by the query.

InSQL_Tagname	Memory Message	Display of the tagname returned by the query.
InSQL_Value	Memory Integer	Display of the tag value returned by the query.
InSQL_Quality	Memory Integer	Display of the quality information for the tag value returned by the query.
InSQL_QualityDetail	Memory Integer	Display of the quality detail for the tag value returned by the query.
ReturnCode	Memory Integer	Success/Failure code returned from a SQL function call.
RowsFound	Memory Integer	Number of rows returned by the query.
SQLErrorMessage	Memory Message	Display of the test message translation of ReturnCode.

N. Venable

The *Tech Note* is published occasionally by Wonderware Technical Support. Publisher: Invensys Systems, Inc., 26561 Rancho Parkway South, Lake Forest, CA 92630. There is also technical information on our software products at [www.wonderware.com/support/mmi](http://www.wonderware.com/support/mmi)

For technical support questions, send an e-mail to [support@wonderware.com](mailto:support@wonderware.com).



[back to top](#)

©2011 Invensys Systems, Inc. All rights reserved. No part of the material protected by this copyright may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, broadcasting, or by any information storage and retrieval system, without permission in writing from Invensys Systems, Inc.