ArchestrA™ SQLDataGrid Symbol User's Guide

Invensys Systems, Inc.

Revision A
Last Revision: 3/18/08

Copyright

© 2008 Invensys Systems, Inc. All Rights Reserved.

All rights reserved. No part of this documentation shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Invensys Systems, Inc. No copyright or patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this documentation, the publisher and the author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

The information in this documentation is subject to change without notice and does not represent a commitment on the part of Invensys Systems, Inc. The software described in this documentation is furnished under a license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of these agreements.

Invensys Systems, Inc.
26561 Rancho Parkway South
Lake Forest, CA 92630 U.S.A.
(949) 727-3200

http://www.wonderware.com

For comments or suggestions about the product documentation, send an e-mail message to productdocs@wonderware.com.

Trademarks

All terms mentioned in this documentation that are known to be trademarks or service marks have been appropriately capitalized. Invensys Systems, Inc. cannot attest to the accuracy of this information. Use of a term in this documentation should not be regarded as affecting the validity of any trademark or service mark.

Alarm Logger, ActiveFactory, ArchestrA, Avantis, DBDump, DBLoad, DT Analyst, Factelligence, FactoryFocus, FactoryOffice, FactorySuite, FactorySuite A^2, InBatch, InControl, IndustrialRAD, IndustrialSQL Server, InTouch, MaintenanceSuite, MuniSuite, QI Analyst, SCADAlarm, SCADASuite, SuiteLink, SuiteVoyager, WindowMaker, WindowViewer, Wonderware, Wonderware Factelligence, and Wonderware Logger are trademarks of Invensys plc, its subsidiaries and affiliates. All other brands may be trademarks of their respective owners.

Contents

Welcome 5

Documentation Conventions 5

Technical Support 6

Chapter 1 Configuring the SQLDataGrid Symbol 7

Using the SQLDataGrid Symbol 8

Configuring Authentication Modes 9

Windows Authentication 9

SQL Server 9

Configuring the Database Connection 10

Configuring SQLDataGrid Symbol Custom Properties 10

Using Scripts with the SQLDataGrid 12

Associating Grid Control Actions to InTouch Scripts and Animations 12

Accessing Data from a Selected Row in the SQLDataGrid 13

Chapter 2 Working with the SQLDataGrid in Run Time 15

Viewing Data 16

Populating the Table, View or Query List 16

Solving Hide and Show Window Issues 16

Moving Columns 17

Grouping Columns 17

Viewing Aggregate Values 19
Using Progressive Column Filtering 20
Discarding Changes 22
Viewing Errors 23
Writing Changes to the Database 23
Views and Updates 24
Database Connection Management 26
Connection Sharing and Pooling 26

Appendix A Custom Properties for the SQLDataGrid Symbol 27

Index 35

Welcome

This guide describes configuring and using the SQLDataGrid symbol. The SQLDataGrid is delivered as an ArchestrA Symbol and is used as an InTouch graphic.

You can view this document online or you can print it, in part or whole, by using the print feature in Adobe Acrobat Reader.

This guide assumes you know how to use Microsoft Windows, including navigating menus, moving from application to application, and moving objects on the screen. If you need help with these tasks, see the Microsoft online help.

This guide also assumes that you know how to use Microsoft SQL Server. For help with SQL Server, see the Microsoft online help.

In some areas of the Application Server, you can also right-click to open a menu. The items listed on this menu change, depending on where you are in the product. All items listed on this menu are available as items on the main menus.

Documentation Conventions

This documentation uses the following conventions:

Convention	Used for
Initial Capitals	Paths and file names.
Bold	Menus, commands, dialog box names, and dialog box options.
`Monospace`	Code samples and display text.

Technical Support

Wonderware Technical Support offers a variety of support options to answer any questions on Wonderware products and their implementation.

Before you contact Technical Support, refer to the relevant section(s) in this documentation for a possible solution to the problem. If you need to contact technical support for help, have the following information ready:

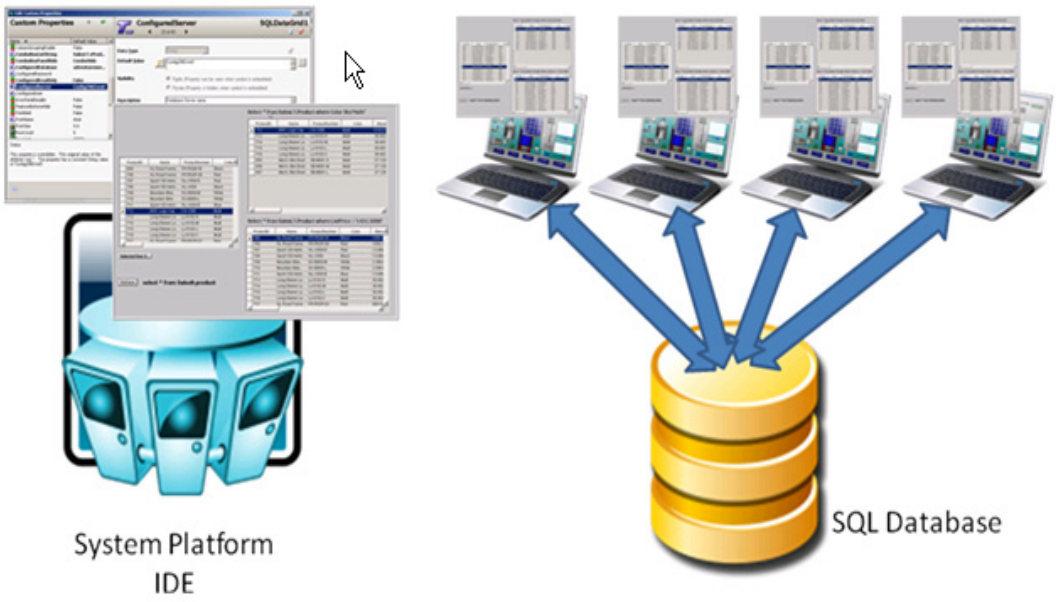
- The type and version of the operating system you are using. For example, Microsoft Windows XP, SP1.
- Details of how to recreate the problem.
- The exact wording of the error messages you saw.
- Any relevant output listing from the Log Viewer or any other diagnostic applications.
- Details of what you did to try to solve the problem(s) and your results.
- If known, the Wonderware Technical Support case number assigned to your problem, if this is an ongoing problem.

Chapter 1

Configuring the SQLDataGrid Symbol

The SQLDataGrid is a symbol that enables you to interact with data stored in SQL databases. You can use the SQLDataGrid symbol in an InTouch application that you create in a published or managed mode.

The following graphic shows the main components associated with the SQLDataGrid.



You can perform the following activities with the SQLDataGrid symbol:

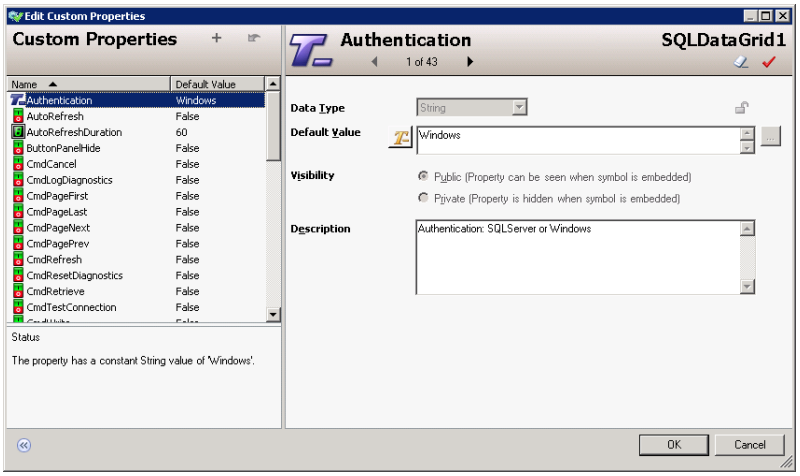
- Insert or modify data in selected database tables and views.
- Delete rows from a database table (select the row and press the **Delete** key).
- Select data from existing tables and views.
- Run a custom SQL query.

This section provides the following information:

- The minimum configuration required by the SQLDataGrid symbol to retrieve data from a SQL database.
- The set of custom properties that must be configured to authenticate users and connect to the database.
- A list of the custom properties that define the parameters associated with a specific database and query to be retrieved.

Using the SQLDataGrid Symbol

The SQLDataGrid symbol appears in the **Graphic Toolbox** of the ArchestrA IDE. Embed the symbol in an existing window or a new window. When you double-click the symbol, the **Edit Custom Properties** dialog box appears.



For detailed information on working with symbols, see the *ArchestrA and InTouch Integration Guide*.

Configuring Authentication Modes

This section describes how to use the custom properties of the SQLDataGrid symbol to authenticate a user to the database. You can assign static values to these custom properties or link them to attributes or tag names in your application.

Note The value fields are not case sensitive.

To configure an authentication mode for the SQLDataGrid symbol

1. In the **Custom Properties** pane, select **Authentication_dg**.
2. In the **Default Value** box, type the name of a valid security type. Acceptable values are:
   - Windows
   - SQL Server

   A more detailed description of these authentication modes follows this procedure.
3. In the **Custom Properties** pane, select **ConfiguredUser_dg**.
4. In the **Default Value** box, type a user name.
5. In the **Custom Properties** pane, select **ConfiguredPassword_dg**.
6. In the **Default Value** box, type the password associated with the user name.

Windows Authentication

If you use Windows as a value for the **Authentication_dg** custom property and do not designate custom properties for a domain, user, or password, the SQLDataGrid symbol uses the credentials of the user who is currently logged on to the node running InTouch.

If you select Windows as a value for the **Authentication_dg** custom property and designate custom properties for a domain, user, and password, the SQLDataGrid symbol uses the specified Windows domain name, user name, and password to connect to the SQL Server database.

SQL Server

This authentication mode uses the specified SQL Server user name and password to connect to the SQL Server database.

Configuring the Database Connection

This section describes the procedure to configure the custom properties required to establish a connection to a specific database and its associated data. You can assign static values to these custom properties or link them to attributes or tag names in your application.

Note The value fields are not case sensitive.

To configure the database connection

1. In the **Custom Properties** pane, select **ConfiguredServer_dg**.
2. In the **Default Value** box, type the network name of the computer hosting the database.
3. In the **Custom Properties** pane, select **ConfiguredDatabase_dg**.
4. In the **Default Value** box, type the name of the target database.
5. In the **Custom Properties** pane, select **SQLQuery_dg**.
6. In the **Default Value** box, type the name of the target table, view or query.

To populate the SQLDataGrid, click **Runtime**, and then click **Retrieve**.

Configuring SQLDataGrid Symbol Custom Properties

This section lists custom properties in functional groups. For a detailed description of each custom property, see Appendix A, Custom Properties for the SQLDataGrid Symbol.

The default run-time window shows the complete symbol. However, you can hide the interface controls and display only the grid.

You can use custom properties to display additional information about the status of the query, the dataset, or to provide interactive buttons that initiate specific actions

In addition to the table, view, and query input, you can specify additional parameters by using custom properties and control them with scripting.

The following table lists the various custom properties you can insert in your scripts by functional groups.

Authentication	Authenticate users to databases **Authentication_dg** **ConfiguredDomain_dg** **ConfiguredPassword_dg** **ConfiguredUser_dg**
Database Connectivity	Designate database connectivity functions **AutoRefresh_dg** **AutoRefreshDuration_dg** **CmdCancel_dg** **CmdRefresh_dg** **CmdRetrieve_dg** **CmdTestConnection_dg** **CmdWrite_dg** **ComboBoxListString_dg** **ConfiguredDatabase_dg** **ConfiguredReadOnly_dg** **ConfiguredServer_dg** **RowCount_dg** **RowLimit_dg** **SelectedRowNumber_dg** **SQLQuery_dg** **SQLQueryCurrent_dg** **SQLQueryPassThrough_dg** **SQLQueryPending_dg**
Diagnostic	Perform diagnostic functions **CmdLogDiagnostics_dg** **CmdResetDiagnostics_dg** **ErrorPanelDisable_dg** **StatusDesc_dg** **StatusError_dg** **StatusIsError_dg**
Appearance	Modify the appearance of the run-time SQLDataGrid that contains data from the database **ButtonPanelHide_dg** **CmdPageFirst_dg** **CmdPageLast_dg** **CmdPageNext_dg**

CmdPagePrev_dg
ColumnAggregateEnable_dg
ColumnFiltering_dg
ColumnGroupingEnable_dg
ComboBoxPanelHide_dg
FontBold_dg
FontName_dg
FontSize_dg
SupportThemes_dg
WriteButtonHide_dg

Using Scripts with the SQLDataGrid

When you use the **SQLDataGridUserCtl** to create a new symbol, you can script all the properties and events of the control.

Associating Grid Control Actions to InTouch Scripts and Animations

You can hide the buttons of the SQLDataGrid symbol and trigger data retrieval from outside the grid interface. For example, you can configure a button to generate a dataset from an InTouch application.

To do this, use the **CmdRetrieve_dg** property and associate it to an InTouch tag. The tag is linked to the InTouch button.

To configure dataset retrieval from InTouch

1. In WindowMaker, open the InTouch window.
2. Double-click the SQLDataGrid symbol. The **Edit Custom Property** dialog box appears.
3. In the **Custom Properties** panel, select the **ButtonPanelHide_dg** property.
4. In the **Default Value** field, type **True**.
5. Select the **CmdRetrieve_dg** property.
6. In the **Default Value** field, select an InTouch tag by one of the following actions:
   a. Type a tag name. If the tag does not exist, you are prompted to define a new tag.
   b. Click the ellipsis to open the Tagname Dictionary and select a tag.

   For this example, type **CmdRetrieve** in the field.
7. Click **OK** and define the tag as a Memory Discrete tag.

8 When you finish defining the tag, close the Tagname Dictionary and click **OK** on the **Custom Properties** panel.

9 Create a button on your InTouch window. For this example, its caption is **Retrieve**.

10 Link the CmdRetrieve tag to the button.

11 Switch to run-time mode and click **Retrieve**.

Accessing Data from a Selected Row in the SQLDataGrid

You can use the SQLDataGrid symbol to read the content of a selected row and associate it to attributes outside the symbol. You can configure scripts to read the data from a selected row and programmatically associate that data to tags in an InTouch application or to attributes in the Galaxy.

The following script example shows how to retrieve data from a selected row. Notice that the **SelectedRowData** property is available in the SQLDataGrid client control and not in the SQLDataGrid symbol.

```
dim rd1 as System.Data.DataRow;
rd1 = SQLDataGridUserCtrl1.SelectedRowData;
if(rd1 <> null) then
  InTouch:Text9 = rd1.ItemArray[4].ToString();
'This assigns the 4th column value for the selected
  row, to the InTouch "Text9" tag.

endif;
```

Chapter 2

Working with the SQLDataGrid in Run Time

During run time, you can select the database table, view, or query that will populate the SQLDataGrid symbol and change the appearance of the data.

You can use the run-time application to perform the following tasks:

- Move columns
- Group columns
- Aggregate column values
- Filter columns
- Sort columns

If you have the appropriate permissions, you can also make the following changes to the database:

- Change the order of columns in an existing row.
- Insert a row into a table.
- Update a row in a table.
- Delete a row from a table.

Note You cannot update or delete a row in a table unless that table has a primary key defined. You can add new rows to a table without a primary key as long as that is the only action you are performing on the table.

Viewing Data

You can use the run-time version of the SQLDataGrid symbol to view data in the following formats:

- Data from a specific table
- Data from a specific view
- Data that is retrieved by a SQL query

You can resize the column width of any table, view, or query result.

To view specific data from a SQL query

1. In the **Table, View or Query** box, type a SQL query.
2. Click **Retrieve**. The appropriate dataset appears.

Populating the Table, View or Query List

You can also populate the **Table, View or Query** box with items that you frequently use, so that you do not repeatedly have to type entries.

To populate the Table, View or Query box

1. In the **Edit Custom Properties** dialog box, click **ComboBoxListString_dg**.
2. In the **Default Value** box, type any of the following items, separated by semicolons (;):
   - Multiple table names
   - Multiple SQL statements
   - A combination of table names and SQL statements
3. Click **OK**.
4. When you go to run time, click the arrow to the right of the **Table, View or Query** box. Each item that you entered is displayed on a separate line.

Solving Hide and Show Window Issues

If you hide the InTouch window that contains the SQLDataGrid and then show it again, the last data values contained in the window might not appear. Use the following procedure to ensure that the data values always appear.

To Ensure that Grid Values Appear When You Hide and Show the InTouch WIndow

1. In WindowMaker, click **Special > Configure > WindowViewer**.
2. Under **WindowViewer Memory**, clear the check box **Always load windows from disk**.

Moving Columns

You can rearrange the order of the columns to meet your viewing needs.

To move columns

1. Click the heading of the column that you want to move.
2. Holding your mouse key down, drag the column heading to another position within the table.

Grouping Columns

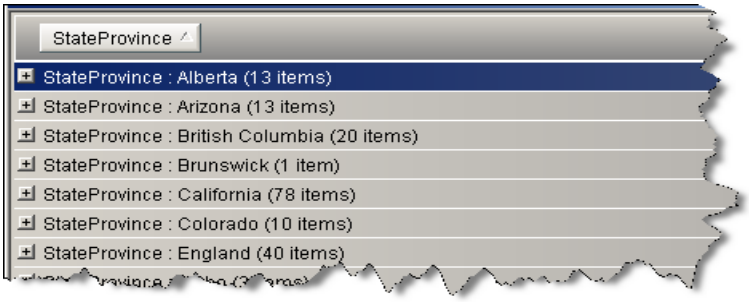
The group-by function lets you sort data at more specific levels. This example shows a view that groups customer names and their business addresses.

Note The ColumnGroupingEnable_dg custom property must be True before you can use this function.

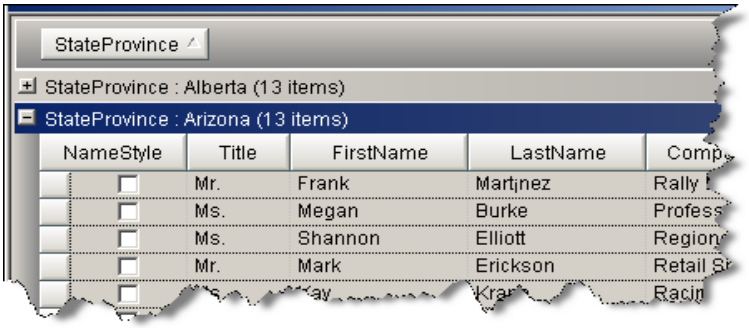
To use the Group-by function

1. In the **Table, View or Query** box, type the name of a table or view or click the arrow to display a list of pre-determined items. For details about pre-determined items, see Viewing Data on page 16.
2. Click **Group-by**. The **Drag a column header** area appears.

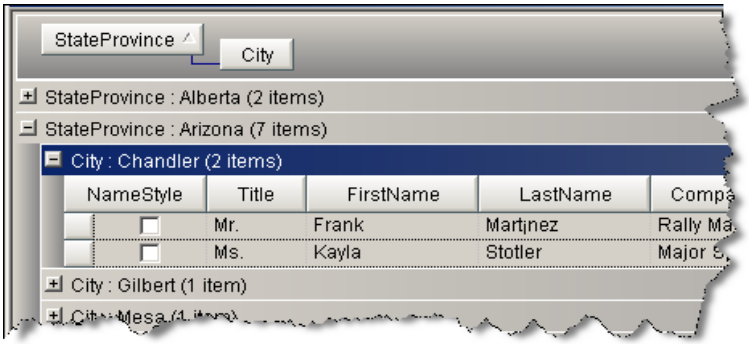
3 Drag a column heading to the upper area.



4 To view the data groups, press **Enter** to expand any of the **StateProvince** items.

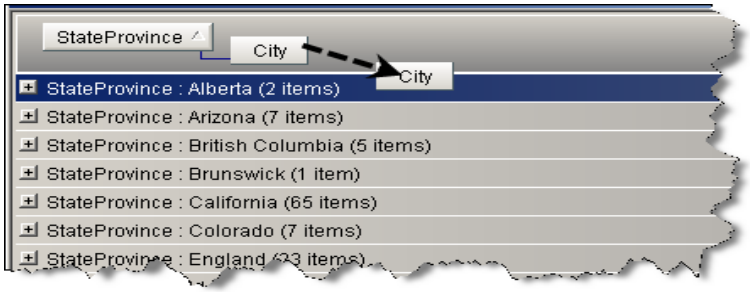


5 To see a more specific sorting of the data, drag another column heading to the upper sorting area.



6 If you want to sort the data at an even finer level, you can repeat steps 3 and 4 and drag another column heading to the sort area.

7 If you want to remove one of the column headings from the sorting area, drag it down to the data area.



8 To return the original dataset, click **Group-by**.

Viewing Aggregate Values

You can perform a numeric operation on data within a single column of the table by clicking the **Aggregate** button.

Note Before you can use this function, the ColumnAggregateEnable_dg custom property must be True.

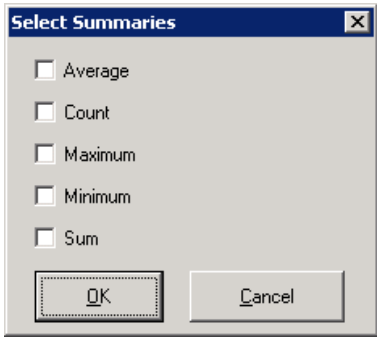
You can perform the following functions on column data. Not all operations may be available for a column because of the data type.

- Average: Calculates the average number of items in the column.
- Count: Lists the number of rows in the column.
- Maximum: Lists the highest number in the column.
- Minimum: Lists the lowest number in the column.
- Sum: Adds the numbers in each row and provides a total.

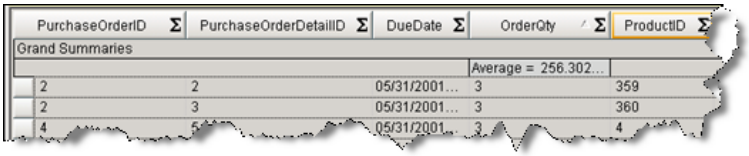
To view aggregate values

1 Click **Aggregate**. The summation sign (Σ) appears at the head of each column.

2 Click the summation sign at the top of the **OrderQty** column. The **Select Summaries** dialog box appears.



3 Select **Average** and click **OK**. The value appears just below the **Grand Summaries** row. You can re-size the column width to view the entire number This following figure shows the average value for **OrderQty**.



4 To hide the **Grand Summaries** row follow these steps:

a Click the summation sign.

b Clear the **Average** check box in the **Select Summaries** dialog box

c Click **OK**.

5 To hide the summation signs, click **Aggregate**.

Using Progressive Column Filtering

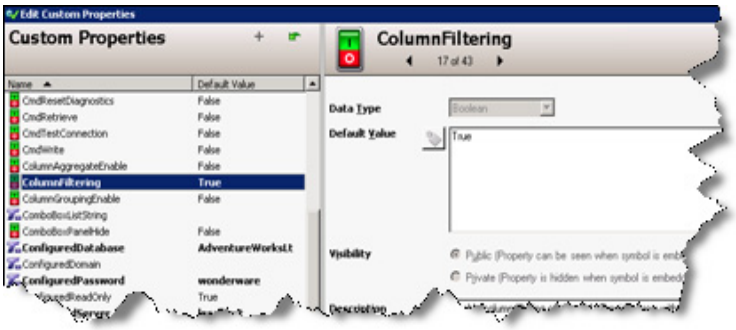
You can configure the SQLDataGrid to support progressive column filtering. Type the first letters of your search, and the matching values appear in the grid.

To configure and use progressive filtering

1 Switch your SQLDataGrid window to WindowMaker mode.

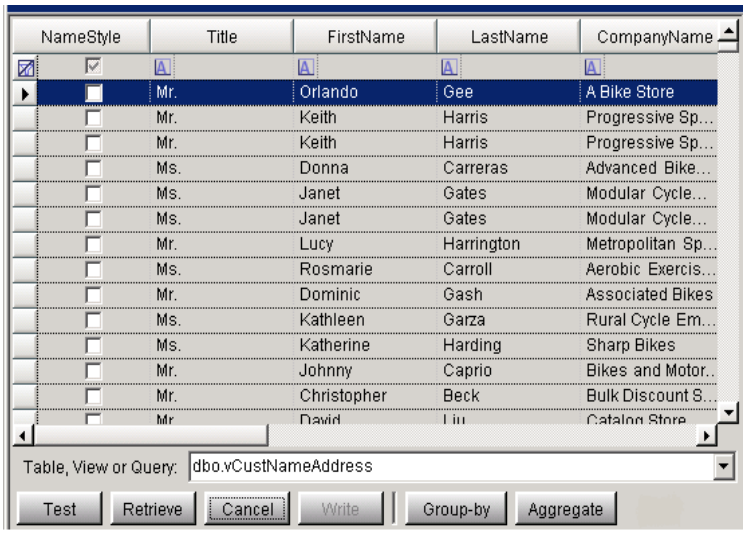
2 Double-click the symbol and select the custom property **ColumnFiltering_dg**.

3 Set the **ColumnFiltering_dg** default value to True.



4 Click **OK** and run the InTouch application.

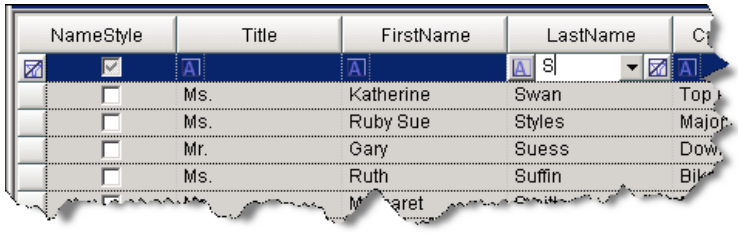
5 To return a dataset, click **Retrieve**. In this example, the dataset contains a table view of customer names and addresses.



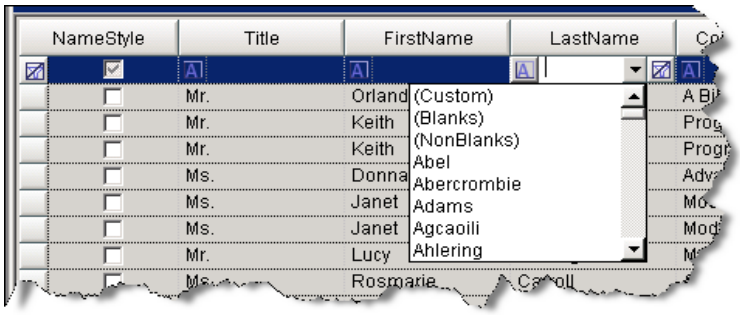
6 Move your cursor over the field below the **LastName** column heading. A list arrow appears.

7 You can filter the values by any of the following actions:

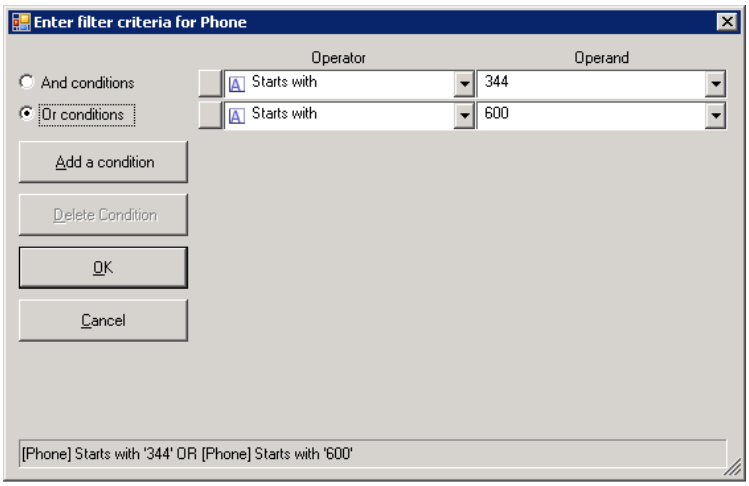
- Click the list arrow and type the first letter of the value If you type additional letters, your sort is more precise.



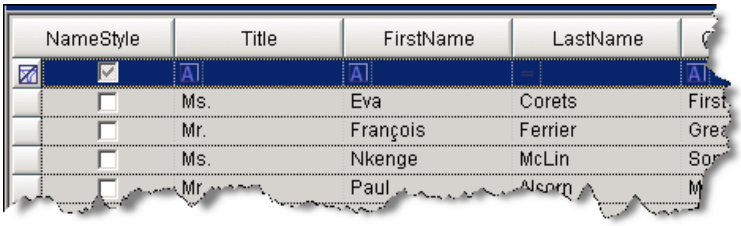
- Click the list arrow and select an item from the list.



- If you select the **(Custom)** item from the list, you can further refine your search in the **Custom** dialog box. This example shows a search for two different area codes.



8 To perform a different search, clear your search criteria by clicking the cancel icon at the left of the filtering row.



Discarding Changes

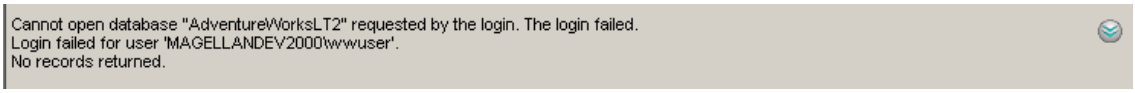
If you need to discard the changes made to grid values, click **Retrieve**. This action reloads the table values from the database and discards any changes. You can also initiate the action by using a script that sets the **CmdRetrieve_dg** property to True.

Viewing Errors

You can manually display or close the **Status** panel in the lower part of the grid display area. The status panel can also automatically appear when an error occurs. However, after the panel appears, it remains on view.

If a command object is successful, the message "Success" appears in the Status panel. If the panel is closed, "Success" messages do not appear for successful command objects.

To close the panel, click the double-arrow icon at the right of the panel. To open the panel on demand, right-click and select **Show Status**.



Note The error panel display does not change the overall area occupied by the grid symbol.

Writing Changes to the Database

You can write changes to the database if the following conditions are met:

- The database query (or table) can be updated.
- The **ConfiguredReadOnly_dg** custom property value is False.
- SQL Server security allows the security mode configured at the grid to perform an update.
- Changes made in the grid are consistent with the database schema.

Changes made in the grid are written to the database using the **Write** command. Changes are handled as a single database transaction. If any change fails, the entire set of changes is rolled back.

If you have sufficient privileges, you can change a value and click **Write**. You can also use a script to set the **CmdWrite_dg** property. This action causes the grid contents to be evaluated for changes and for the changes to be written to the database.

Tip After you modify cells in the grid and write them back to the database, you should also verify that the data is correctly written. In certain cases, such as a database table with an automatically incrementing primary key or interdependent cells, the grid does not show the true contents of the database table after you write back to it. Also, no error message is shown. It is recommended that you execute a retrieve or auto refresh command after writing to the database to make sure it is showing the current data.

Views and Updates

Some views cannot be updated for the following reasons:

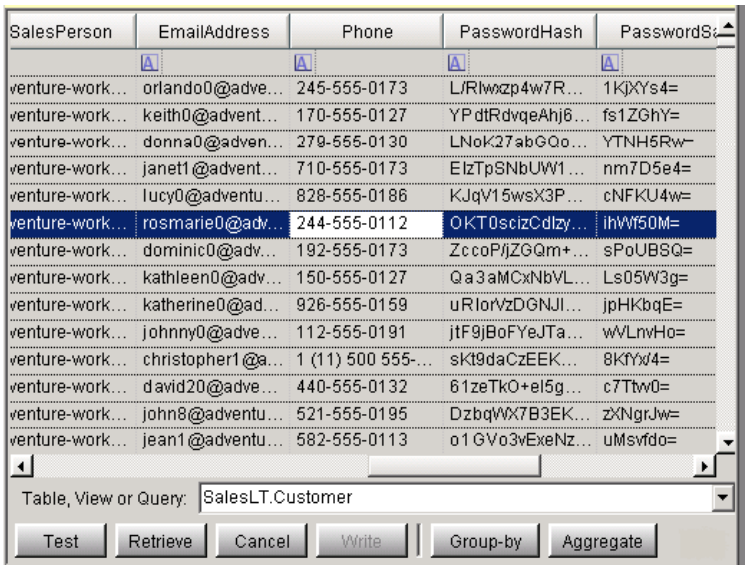
- The views contain table joins on keys other than primary keys.
- The views include security settings that prohibit updates.

If the write process fails, you can view the status in the **Error** panel. You can also monitor the **StatusDesc_dg**, and **StatusError_dg** properties by using text fields on the InTouch window for run-time display.

To make a change to data in the data grid

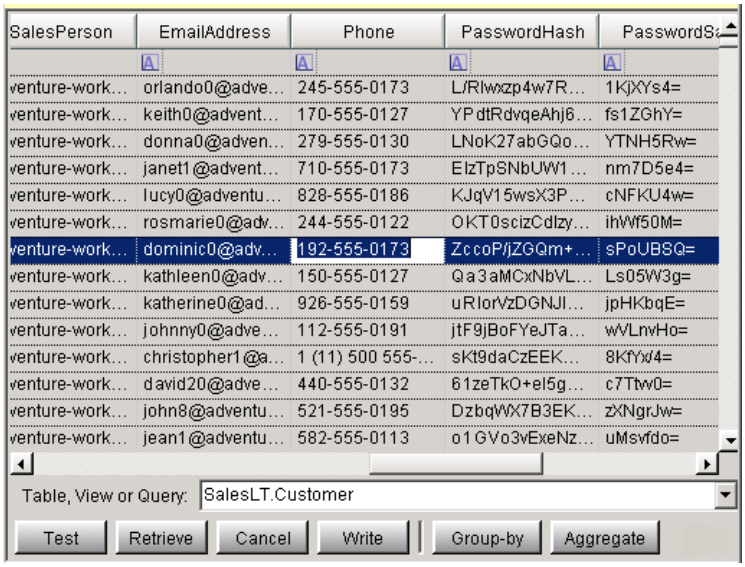
1. Ensure that the **ConfiguredReadOnly_dg** custom property value is False.
2. Click **Runtime**.
3. Enter a SQL statement or the name of a database table or view.
4. Click **Retrieve**. This example shows data from the customer table in AdventureWorks. You will modify a phone number.

5 Select the phone number that you must modify. The area becomes editable.



6 Change the last four digits of the phone number to 0122.

7 Click an area outside the edit area. The **Write** button becomes enabled.



8 Click **Write**.

9 To verify the change, click **Retrieve**.

Database Connection Management

An important feature of the SQLDataGrid is the ability to optimize the management of multiple simultaneous connections to the database. This section describes how the infrastructure handles the database connection pool and sharing for the SQLDataGrid.

Connection Sharing and Pooling

Multiple instances of the SQLDataGrid configured with identical connection strings use a shared connection pool and SQL command queue when they are active in the same instances of the InTouch WindowViewer. To limit the size of the shared pool, you can use the connection pool property **MaxConnections**. For configuration options, see the MaxConnections_dg on page 32.

Physical connections to the SQL database are opened based on command requests and are closed when no requests are pending. All SQL commands are queued and processed based on availability of connections and processing time.

The SQLDataGrid asynchronously processes all SQL commands. Asynchronous command processing ensures availability of interaction with the InTouch WindowViewer during time-consuming command processing.

Any database connection that is established in a script within a symbol using the SQLData Script Library also takes advantage of connection pooling. Similar connection strings from either the SQLDataGrid or the script in the same InTouchView application share a common pool.

Note Connection sharing and connection pooling require that the same security mode and user credentials be used for all SQLDataGrid symbols within the InTouchView application that shares the connection or the pool.

Appendix A

Custom Properties for the SQLDataGrid Symbol

The following table lists all available Custom Properties. Data types are shown in the Custom Properties panel and are not included in this table.

Property Name	Description
Authentication_dg	Designates whether database user authentication is SQL Server or Windows. **Writable at Run Time**: Yes **Default**: Windows
AutoRefresh_dg	Automatically refreshes data appearing in the SQLDataGrid. When True, the query that repopulates the SQLDataGrid runs after the **AutoRefreshDuration** time elapses. **Writable at Run Time**: Yes **Default**: False
AutoRefreshDuration_dg	Designates the requested time interval in seconds for AutoRefresh. The actual refresh rate can be slower than the specified rate. **Writable at Run Time**: Yes **Default**: 60

Property Name	Description
ButtonPanelHide_dg	Hides grid buttons when the value is True. **Writable at Run Time**: Yes **Default**: False
CmdCancel_dg	Cancels the last database query. **Writable at Run Time**: Yes **Default**: False
CmdClear_dg	Clears the contents of the SQLDataGrid. **Writable at Run Time**: Yes **Default**: False
CmdCopy_dg	When True, copies the current selection to the clipboard. **Writable at Run Time**: Yes **Default**: False
CmdLogDiagnostics_dg	Writes the current DCM diagnostic values to the ArchestrA logger. **Writable at Run Time**: Yes **Default**: False
CmdPageFirst_dg	Moves the SQLDataGrid display to the first row. **Writable at Run Time**: Yes **Default**: False
CmdPageLast_dg	Moves the SQLDataGrid display to the last row. **Writable at Run Time**: Yes **Default**: False
CmdPageNext_dg	Moves the SQLDataGrid display down one page. **Writable at Run Time**: Yes **Default**: False
CmdPagePrev_dg	Moves the SQLDataGrid display up one page. **Writable at Run Time**: Yes **Default**: False
CmdPaste_dg	When True, pastes the current text contained on the clipboard to the grid. **Writable at Run Time**: Yes **Default**: False

Property Name	Description
CmdRefresh_dg	Refreshes the current SQL query data. Any pending changes are discarded. **Writable at Run Time**: Yes **Default**: False
CmdResetDiagnostics_dg	Resets the current DCM diagnostic values and clears the message and event counters. **Writable at Run Time**: Yes **Default**: False
CmdRetrieve_dg	Loads the selected table, view, or query to the SQLDataGrid. Any changes to a previous dataset are discarded. **Writable at Run Time**: Yes **Default**: False
CmdTestConnection_dg	Tests the database connection for the specified server, database, security type; and if applicable, user name and password. The result is placed in the Error panel. **Writable at Run Time**: Yes **Default**: False
CmdWrite_dg	Submits changes made in the SQLDataGrid to the database. **Writable at Run Time**: Yes **Default**: False
ColumnAggregateEnable_dg	Specifies whether the column aggregate feature is exposed by default. **Writable at Run Time**: Yes **Default**: True.
ColumnFiltering_dg	Enables column filtering and provides a row with the progressive filtering entry field. **Writable at Run Time**: Yes **Default**: False
ColumnGroupingEnable_dg	Specifies whether the column grouping feature of the SQLDataGrid is exposed by default. **Writable at Run Time**: Yes **Default**: False

Property Name	Description
ComboBoxListReadOnly_dg	If False, you can make changes to the edit box above the combo list box. **Writable at Run Time**: Yes **Default**: False
ComboBoxListString_dg	Enables a list of table or view names, separated by semicolons to be viewable in the **Runtime** combo box. **Writable at Run Time**: Yes **Default**: Blank
ComboBoxPanelHide_dg	Hides the combo box and panel. **Writable at Run Time**: Yes **Default**: False
CommandTimeout_dg	The number of seconds that the SQLDataGrid waits for a pending command to run before timing out. **Writable at Run Time**: Yes **Default**: 30
ConfiguredDatabase_dg	The name of the database that connects to the SQLDataGrid. **Writable at Run Time**: Yes **Default**: Blank
ConfiguredDomain_dg	The domain to use with impersonation if the authentication mode is Windows. **Writable at Run Time**: Yes **Default**: Blank
ConfiguredPassword_dg	SQL Server domain if the security type is Windows. **Writable at Run Time**: Yes **Default**: ********
ConfiguredReadOnly_dg	Prevents the SQLDataGrid from making changes to the database. **Writable at Run Time**: Yes **Default**: True

Property Name	Description
ConfiguredServer_dg	The name of the computer hosting the database. **Writable at Run Time**: Yes **Default**: Blank
ConfiguredUser_dg	The SQL Server user name if security type is SQL. This property is optional when security type is Windows. **Writable at Run Time**: Yes **Default**: Blank
ConnectionTimeout_dg	The number of seconds that the SQLDataGrid tries to connect to a database before timing out. **Writable at Run Time**: Yes **Default**: 15
ContextMenuEnable_dg	When True and you right-click the grid, the context menu appears. When False and you right-click the grid, the context menu does not appear. **Writable at Run Time**: Yes **Default**: True
ErrorPanelDisable_dg	Disables the error message display. **Writable at Run Time**: Yes **Default**: False
ErrorPanelShow_dg	When True, the error panel is visible. When False, the error panel is hidden. **Writable at Run Time**: Yes **Default**: False
FeatureButtonsHide_dg	Hides the **Aggregate** and **Group-by** buttons. **Writable at Run Time**: Yes **Default**: False
FontBold_dg	Specifies whether to display the SQLDataGrid font in bold text. **Writable at Run Time**: Yes **Default**: False

Property Name	Description
FontName_dg	Specifies the name of the font. **Writable at Run Time**: Yes **Default**: Arial
FontSize_dg	Specifies the point size of the font. **Writable at Run Time**: Yes **Default**: 9.0
InstanceName_dg	A name that you choose to identify a specific SQLDataGrid instance. The name appears in the **ConnectionName** diagnostic. **Writable at Run Time**: Yes **Default**: Blank
MaxConnections_dg	The number of active connections allowed in the connection pool for a given connection string. **Writable at Run Time**: Yes **Default**: 100
RowCount_dg	The row count returned in the last successful data request. **Writable at Run Time**: No **Default**: 0
RowLimit_dg	Modifies the query to return a specified number of records. To prevent the database query from being modified, set the **SQLQueryPassThrough_dg** custom property to True. **Writable at Run Time**: Yes **Default**: 20,000
RowsChanged_dg	If True, one or more rows have been changed in the grid (added, odified, or deleted). If False, no rows were changed in the grid. **Writable at Run Time**: No **Default**: False
SelectedRowNumber_dg	The number of the selected grid row. **Writable at Run Time**: Yes **Default**: 0

Property Name	Description
SQLQuery_dg	Specifies the SQL query, table, or view data to be retrieved to the SQLDataGrid. If **SQLQueryPassThrough_dg** is True, you must add the SELECT statement along with the table or view name. For example, *SELECT * FROM person.contact*, rather than *person.contact*. **Writable at Run Time**: Yes **Default**: Blank
SQLQueryCurrent_dg	Specifies the SQL command that was run to show the current data in the SQLDataGrid. **Writable at Run Time**: No **Default**: Blank
SQLQueryPassThrough_dg	If True, enables the SQL statement specified in the **SQLQuery_dg** custom property to be passed directly to the database without modification. When this property is True and the **SQLQuery_dg** custom property is set only to a table name or view, the grid does not insert "SELECT * FROM" in front of the table or view name. Additionally, if this property is True, the top "n" is not inserted into a SELECT statement (where "n" is the value of the RowLimit_dg custom property). **Writable at Run Time**: Yes **Default**: False
SQLQueryPending_dg	Indicates a pending query. **Writable at Run Time**: No **Default**: False
StatusDesc_dg	Specifies the error description from the last query. If no error exists, the string is empty. **Writable at Run Time**: No **Default**: Blank
StatusError_dg	Specifies the error number from the last request. **Writable at Run Time**: No **Default**: 0

Property Name	Description
StatusIsError_dg	Indicates whether an error occurred in the last query. TRUE = Error FALSE = no error **Writable at Run Time**: No **Default**: False
SupportThemes_dg	Specifies the SQLDataGrid window is formatted with Windows XP or Vista visual themes. You should always set the value of this custom property to True. This custom property is not yet supported. **Writable at Run Time**: No **Default**: True
WriteButtonHide_dg	Hides the **Write** button used to send changes to the database. **Writable at Run Time**: Yes **Default**: False

Index

A

aggregate values 19
authentication
  modes 26

C

changes
  database 23
  discarding 22
columns
  aggregating values 19
  grouping 17
  moving 17
  progressive filtering 20
  rearraranging order 17
components of SQLDataGrid 7
configuration
  custom properties 10
  database connection 10
  progressive column filtering 20
  security 26
connection pooling 26
connection sharing 26
conventions, documentation 5
custom properties
  alphabetical list 27
  configuring 10
customizing the grid 10

D

data, viewing 16
database
  configuring connection 10
  updating 23
discarding changes to grid values 22
Documentation 5
documentation conventions 5

E

errors, viewing 23

F

functions
  aggregation 19
  group-by 17

G

grid values, discarding changes 22
group-by function 17
grouping columns 17

I

InTouch window, hide and show issues 16

M

moving columns 17

O

order of columns, rearranging 17

P

pooling connections 26
progressive column filtering 20
properties, custom 10, 27

R

run-time interface
  aggregating values 19
  entering SQL queries 16
  group-by function 17
  progressive column filtering 20
  viewing data 16
  viewing errors 23

S

security, configuring 26
sharing conections 26
SQL queries, entering 16
SQLDataGrid
  components 7
  customizing 10
status panel 23

T

technical support, contacting 6

U

updates
  database 23
  views 24

V

views, updating 24

W

writing changes to database 23