

# **DataSplice®**

## **Mobile Computing**

# **Administration Client**

## **User Manual**

**Revised: 6/9/2005**



DataSplice, LLC  
414 E. Oak Street  
Fort Collins, CO 80524  
Web: <http://www.datasplice.com>  
Email: [datasplice@datasplice.com](mailto:datasplice@datasplice.com)  
Phone: 800.377.1974  
Fax: 970.484.0965

## Copyright Information

DataSplice® is a registered trademark. ©2000-2005 DataSplice Corporation. All rights reserved. Other product and company names may be the trademarks of their respective owners.

MAXIMO® software is a registered trademark of MRO Software, Inc. DataSplice is not owned, licensed, or in any way affiliated with MRO Software, Inc. MRO Software, Inc. is not responsible for the content or accuracy of this document or for the content, performance or quality of DataSplice products or services.

Microsoft®, Visual Studio, Windows and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries/regions.

Other names and products mentioned herein may be the trademarks of their respective owners.

# Table of Contents

---

Introduction.....	5
Purpose.....	5
Version Information.....	5
User Interface.....	6
Menus.....	7
File Menu.....	7
Edit Menu.....	8
View Menu.....	9
Help Menu.....	9
Toolbar Buttons.....	11
Shared Buttons.....	11
Section-Specific Buttons.....	11
Left Pane.....	15
Right Pane.....	16
Connecting to the DataSplice Server.....	17
System Information.....	18
Options.....	19
Properties Tab.....	20
Connection Profiles Tab.....	22
System Attributes Tab.....	23
Licenses Tab.....	24
Plug-ins.....	27
Status.....	28
Connected Clients Tab.....	29
Current Sessions Tab.....	31
Worker Threads Tab.....	32
Database Handles Tab.....	33
Users and Groups.....	34
Creating Users.....	35
Creating Groups.....	36
Deleting Users & Groups.....	37
User & Group Properties.....	38
Domains.....	41
Views.....	43
Creating Views.....	43
Deleting Views.....	43
Renaming Views.....	44
Connection Tab.....	45
Data Tab.....	46
Fields Tab.....	47
Creating Field Validation.....	48
Attributes Tab.....	50
Precedence.....	50
Built-in Attributes for the Current Connection.....	50

Standard Attributes .....	51
Query-Based Attributes .....	54
View Criteria Tab .....	58
Adding Criteria Sets to a View .....	58
Edit View Criteria Dialog Box Properties .....	59
Creating Simple Filters .....	59
Creating Complex Filters .....	60
Combining Filters .....	61
Advanced Filters .....	61
Filter Operators .....	62
Wildcards .....	62
Sorting Tab.....	62
Relationships Tab.....	63
Creating a New Relationship .....	64
Deleting a Relationship.....	64
Editing Relationship Dialog Box Properties.....	65
Events Tab .....	66
Creating Actions for View Events .....	66
View Event Types.....	66
Creating a Database Procedure Action .....	67
Creating an Email Notification Action .....	69
Creating a Run Application Action.....	71
Creating a Plug-in Action .....	73
Preview Tab .....	75
Setting Default Column Widths.....	75
Editing Data in the Preview Tab.....	75
View Permissions.....	76
Overriding Defaults .....	78
Reference .....	81
ODBC .....	82
Acquiring ODBC .....	82
Data Sources .....	82
Creating a Data Source in Microsoft ODBC Data Source Administrator .....	82
Drivers.....	84
Installing ODBC Drivers .....	85
Connection Pooling.....	85
SQL .....	87
Accessing Information through DataSplice .....	87
Basic SQL Queries.....	87
Logical Operators.....	88
Complex SQL Queries.....	89
More Information.....	90
Format Strings.....	91
Formatting Date and Time .....	91
Formatting Decimals.....	91
Formatting Options .....	92

# Introduction

## Purpose

This document details the DataSplice Administration Client application. It provides information on all of the different areas of the program, and explains each in turn. The DataSplice Administration Client enables an administrator to configure and monitor mobile applications provided by the DataSplice Server. All server configurations and settings can be accessed through the Administration Client.

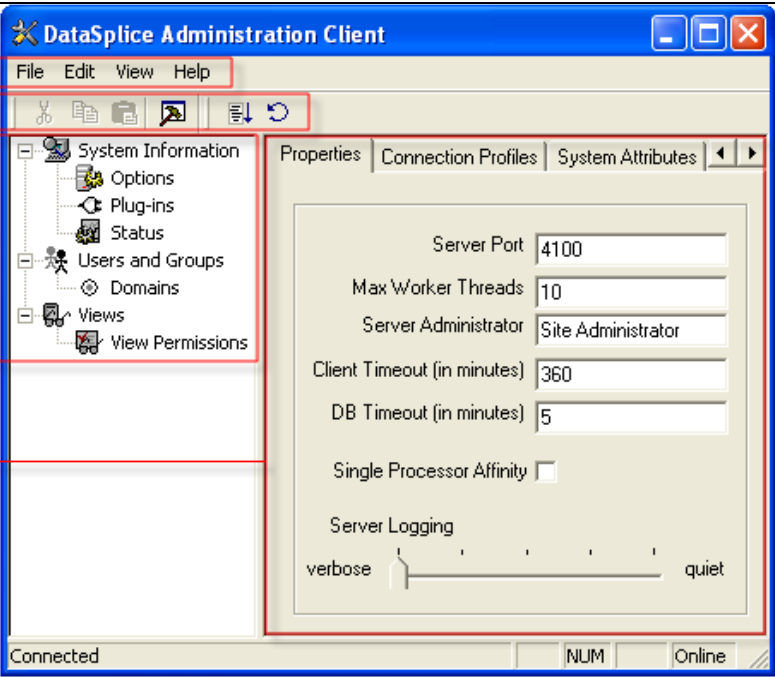
For additional documentation and support resources, please visit <http://support.datasplice.com>.

## Version Information

The information in this document pertains to DataSplice version **2.3.10.0**.

# User Interface

The first step in learning how to successfully use the DataSplice Administration Client is to familiarize yourself with the user interface of the program. The user interface can be broken down into the following important sections:

User Interface	
<ol style="list-style-type: none"> <li>1. <b>Menus</b></li> <li>2. <b>Toolbar Buttons</b></li> <li>3. <b>Left Pane</b></li> <li>4. <b>Right Pane</b></li> </ol>	

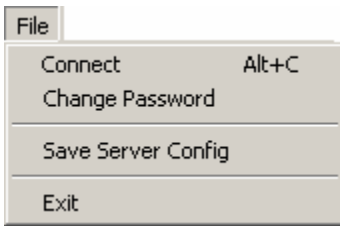
## Menus

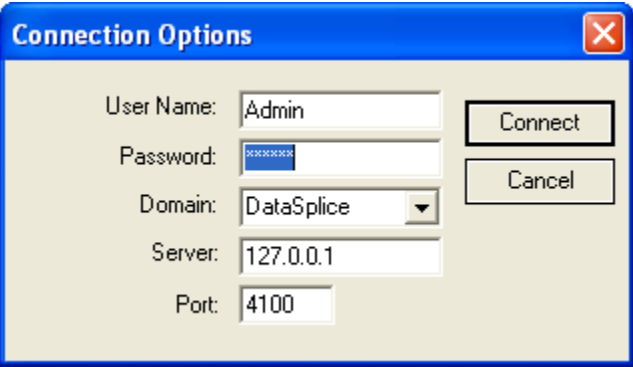
The DataSplice Administration Client has four main menus to choose from:

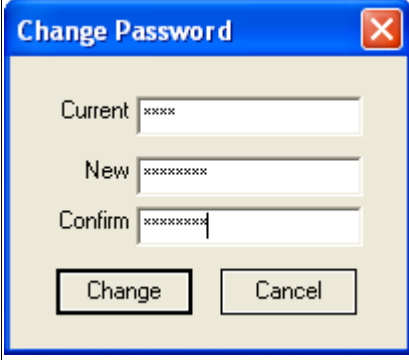

1. **File**
2. **Edit**
3. **View**
4. **Help**

### File Menu

The **File** menu provides access to basic file and connection options.

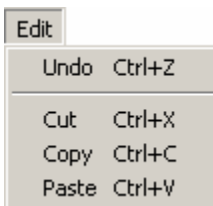


Menu Item	Explanation
<p><b>Connect</b></p>	<p>This menu option connects you to the DataSplice Server. When clicked, you will be prompted to log in with the <b>Connection Options</b> dialog box:</p>  <ul style="list-style-type: none"> <li>• <b>User Name</b> – The user name of the DataSplice administrator. The first time you log in, the user name will be <i>Admin</i>. Remember that user names in DataSplice are case-sensitive.</li> <li>• <b>Password</b> – The password associated with the administrator user name. The first time you log in, the password will be <i>changethis</i>. Remember that passwords in DataSplice are case-sensitive.</li> <li>• <b>Domain</b> – The authentication domain. This will almost always be <i>DataSplice</i>.</li> <li>• <b>Server</b> – The hostname or IP address of the computer running the DataSplice Server. If the DataSplice Server application is running on the same computer as the Administration Client application, enter <i>localhost</i> or <i>127.0.0.1</i>.</li> <li>• <b>Port</b> – The port that the DataSplice Server is using to listen for incoming connections. By default, this is <i>4100</i>.</li> </ul>

	<p><b>Note:</b> When you are connected to the DataSplice Server, this menu item changes to <b>Disconnect</b>. At that time, clicking it will disconnect you from the DataSplice Server.</p>
<b>Change Password</b>	<p>This menu option allows you to change your password. Clicking it brings up the <b>Change Password</b> dialog box, where you must first enter your existing password and then a new one:</p>  <ul style="list-style-type: none"> <li>• <b>Current</b> – Enter your existing password.</li> <li>• <b>New</b> – Enter your new password.</li> <li>• <b>Confirm</b> – Re-enter your new password.</li> </ul> <p>Click <b>Change</b> to establish your new password, or click <b>Cancel</b> to maintain your existing password.</p>
<b>Save Server Config</b>	<p>This option permanently saves the configuration changes you have made to the DataSplice Server.</p> <p><b>Note:</b> If you have clicked the <b>Update</b> button  and you can see your changes, they are only saved in memory. To save the configuration changes permanently to disk, you also need to click the <b>Save Server Config</b> menu option.</p>
<b>Exit</b>	<p>Clicking this menu item disconnects you from the DataSplice Server and closes the Administration Client program.</p>

## Edit Menu

The **Edit** menu provides access to the basic editing functions.

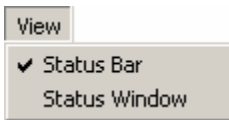


Menu Item	Explanation
<b>Undo</b>	<p>Clicking this menu item will reverse your most recent change.</p> <p><b>Note:</b> It is not possible to undo most deletes.</p>

<b>Cut</b>	When you have an item selected, choosing this menu item will remove it from its current location and load it into the clipboard.
<b>Copy</b>	When you have an item selected, choosing this menu item will leave the original selection unchanged and place a copy of it into the clipboard.
<b>Paste</b>	After you have loaded an item into the clipboard, through either a <b>Cut</b> or <b>Copy</b> command, clicking the <b>Paste</b> menu item will place that item from the clipboard into the selected location.

## View Menu

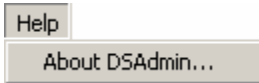
The **View** menu allows you to change what you see in your DataSplice Administration Client program.



Menu Item	Explanation
<b>Status Bar</b>	This menu item toggles the visibility of the information bar at the bottom of the window. This item is checked (visible) by default.
<b>Status Window</b>	This menu item toggles the visibility of the <b>Status Window</b> dialog box. This item is unchecked (not visible) by default.

## Help Menu

The **Help** menu provides version and other information about your DataSplice Administration Client application.



Menu Item	Explanation
<b>About DSAdmin...</b>	This menu item displays the <b>About DSAdmin...</b> dialog box, which contains numerous fields of information about the DataSplice Administration Client application:



## Toolbar Buttons

The user interface of the DataSplice Administration Client has numerous components. One of these components is the toolbar, which is composed of a series of buttons. Some of these buttons are shared among all of the sections of the Administration Client. Other buttons are specific to a section.

### Shared Buttons

The shared toolbar buttons of the Administration Client interface are available while you are in multiple areas of the application.



This button cuts the current selection and places it on the clipboard.



This button copies the current selection and places it on the clipboard.



This button takes the contents of the clipboard and inserts them into the currently active field.



Clicking this button switches between hiding and showing the status window.



This button saves the current DataSplice Server configuration to memory. As you are making changes in the Administration Client and want to test them out, you will want to click this button to reflect your changes. It is important to remember that this button saves the current configuration to memory, not to disk. To save the configuration to disk, click **Save Server Config** from the **File** menu.



This button discards any uncommitted changes and reloads the server options.

### Section-Specific Buttons

Certain toolbar buttons are only available when you are in a specific section of the Administration Client program.

### Users and Groups



**Add User**

This button can be used to create a new user. Please see the **Creating Users** section for additional information.

**Add Group**

This button can be used to create a new group. Please see the **Creating Groups** section for additional information.

**Delete**

This button deletes the selected user or group. Please see the **Deleting Users & Groups** section for further information.

## Domains

**Add Domain**

This button creates a new domain. Please refer to the **Creating Domains** section for additional information.

**Delete**

This button deletes the selected domain.

## Views

**Create View**

This button creates a new view. Please see the **Creating Views** section for more information.

**Delete View**

This button deletes the selected view. Please see the **Deleting Views** section for more information.

**Rename View**

This button allows you to rename the currently selected view. Please see the **Renaming Views** section for more information.

**Validate View**

This button validates the currently selected view.

**Validate All Views**

This button validates all of the views.

## View Permissions



### Add Access

This button allows you to grant access to the current view. Please see the **View Permissions** page for additional information.

### Delete Access

This button allows you to remove access from the current view.

## View Criteria Tab



### Add Criteria

This button creates new criteria. Please see the **View Criteria Tab** section for more information.

### Delete Criteria

This button deletes the selected criteria.

## Relationships Tab



### Add Relationship

This button creates a new relationship. Please see the **Creating a New Relationship** section for more information.

### Delete Relationship

This button deletes the selected relationship. Please see the **Deleting a Relationship** section for more information.

### Move Up

This button moves the current selection up one place in the listing.

### Move Down

This button moves the current selection down one place in the listing.

## Events Tab



### Add Action

This button allows you to create a new event action. Please see the **Events Tab** page for further information.

**Delete Action**

This button deletes the currently selected action.

**Move Up**

This button moves the current selection up one place in the listing.

**Move Down**

This button moves the current selection down one place in the listing.

**Preview Tab**

Notice that the buttons that are specific to the **Preview** tab are like those found on the Remote Client application. Please see the **Preview Tab** section for additional information.

**Next**

This button cycles through the next set of records.

**Previous**

This button goes back to the previous set of records.

**New Record**

This button creates a new record.

**Delete Record**

This button deletes the current record.

**Commit Changes**

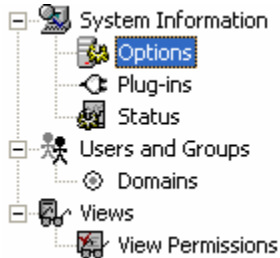
This button commits the pending record modifications to the server.

**Search**









This button retrieves the records according to the search criteria being used.

## Left Pane

The main DataSplice Administration Client interface is split into a left and a right pane. When you select an item in the left pane, the options for that selection become available in the right pane. The left pane of the Administration Client looks like this:

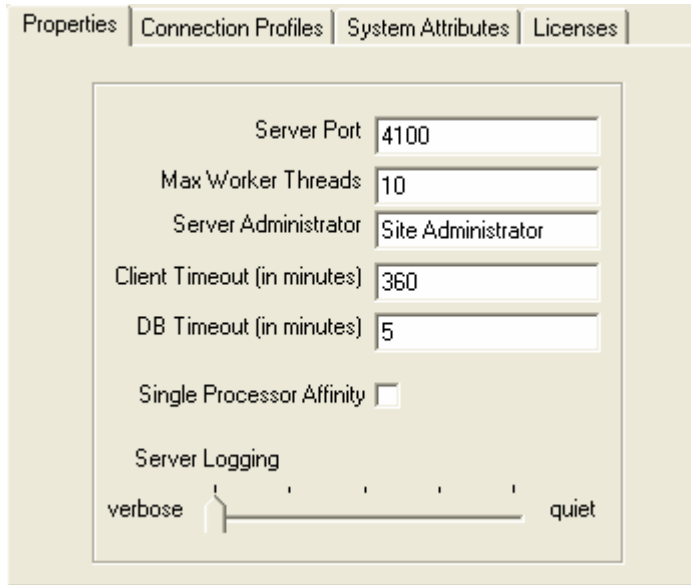


**The following is a brief description of the items in the left pane:**

-  **System Information** – Displays basic system information about the DataSplice Server.
  -  **Options** – Configures basic server options, attributes, and connection profile settings.
  -  **Plug-ins** – Displays information about the plug-ins loaded by the server.
  -  **Status** – Displays information about the current connections, sessions, etc. on the server.
-  **Users and Groups** – Screen for editing user and group properties.
  -  **Domains** – Configures external authentication domains for the server.
-  **Views** – Shows available views and enables the administrator to configure view settings.
  -  **View Permissions** – Enables views to be assigned to users and groups within DataSplice.

## Right Pane

The right pane of the DataSplice Administration Client is where you will actually enter your configuration settings. It contains different tabs and fields, depending on which item is currently selected in the left pane. The following screenshot shows an example of the right pane of the Administration Client:

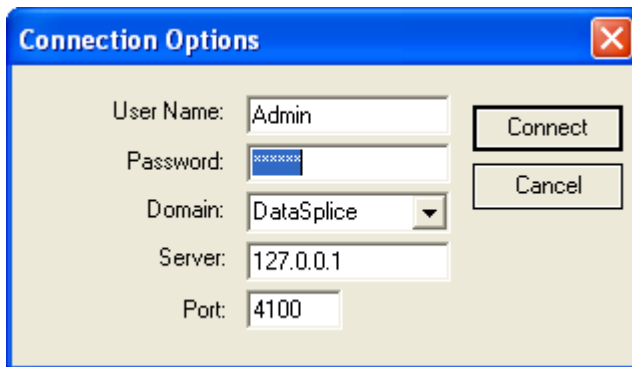


## Connecting to the DataSplice Server

All DataSplice Server configurations and settings can be accessed through the Administration Client. To gain access to the DataSplice Server, you must first connect to it.

### To establish a connection with the DataSplice Server:

1. Ensure that the DataSplice Server is running, either as an application or as a service.
2. Open the Administration Client by clicking **Start > Programs > DataSplice > Administration Client**.
3. From the menu at the top of the window, click **File > Connect**. This brings up the Connection Options dialog box:

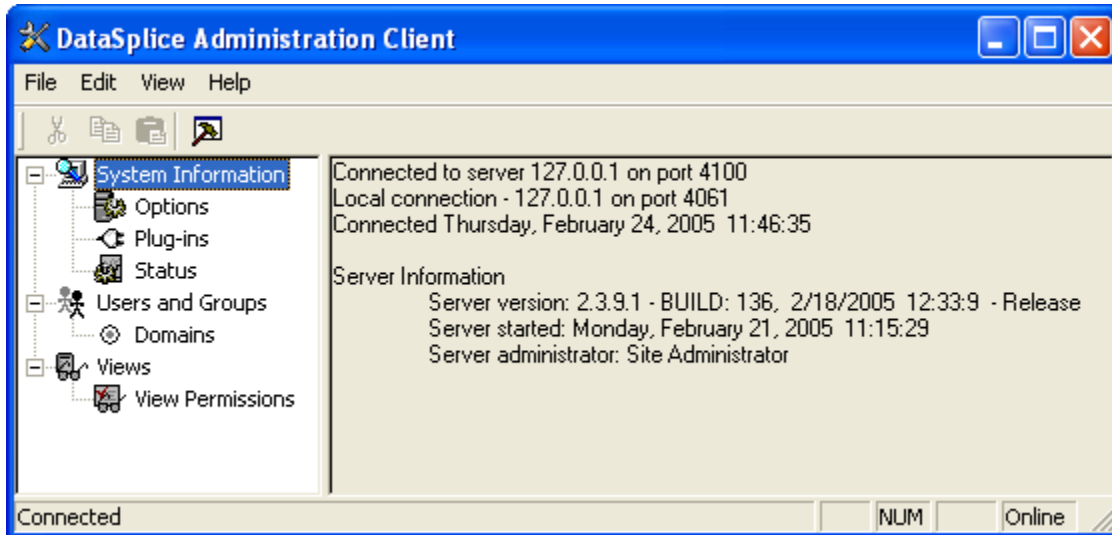


4. Enter your user name and password. **Important note when logging in for the first time:** The default administrator user name for DataSplice is *Admin*, and the default password is *changethis*. After logging in successfully for the first time, you will be prompted to change the password.
5. If necessary, alter the remaining fields in the dialog box accordingly. In most cases, the default values will be correct. For a more detailed description of these dialog box fields, please refer back to the **Connection Options** description in the **Menus** section.
6. Click the **Connect** button.

Upon successful login, you will initially see the **System Information** screen. For more information regarding this screen, please see the following section.

## System Information

The initial System Information screen displays general information about the DataSplice Server, such as its version and how long it has been running.



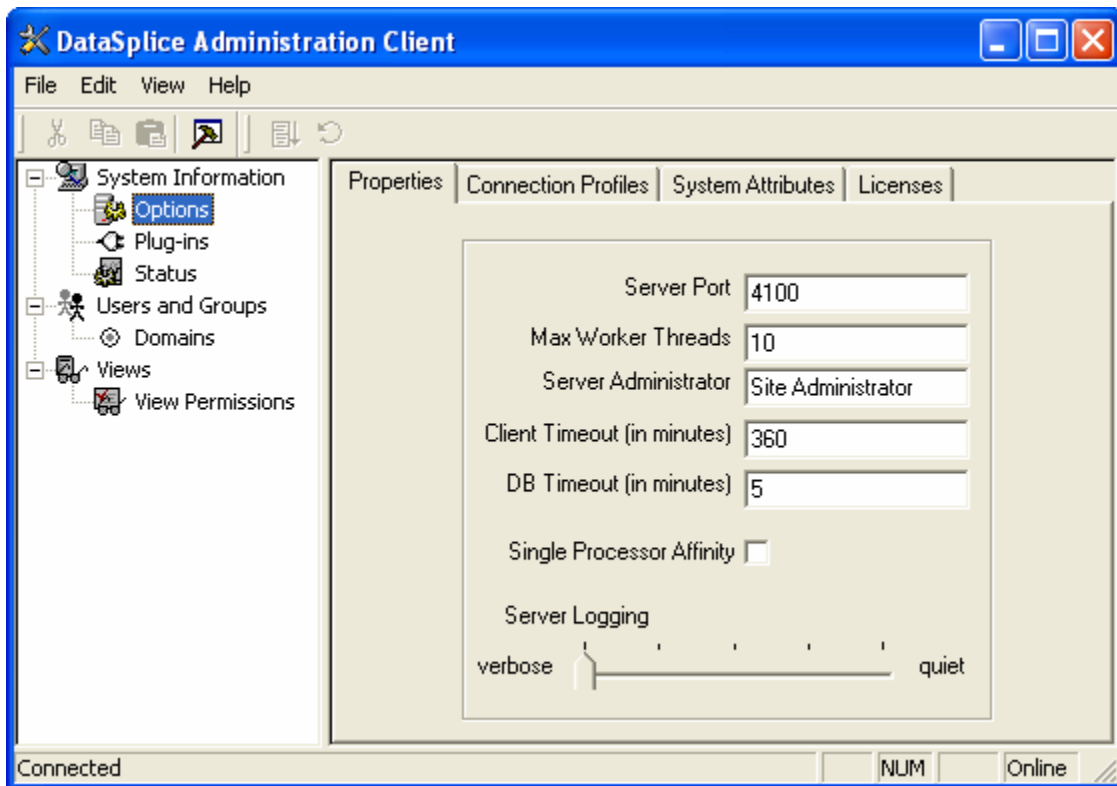
Expanding the **System Information** section in the left pane provides access to the following areas:

- **Options**
- **Plug-ins**
- **Status**

## Options

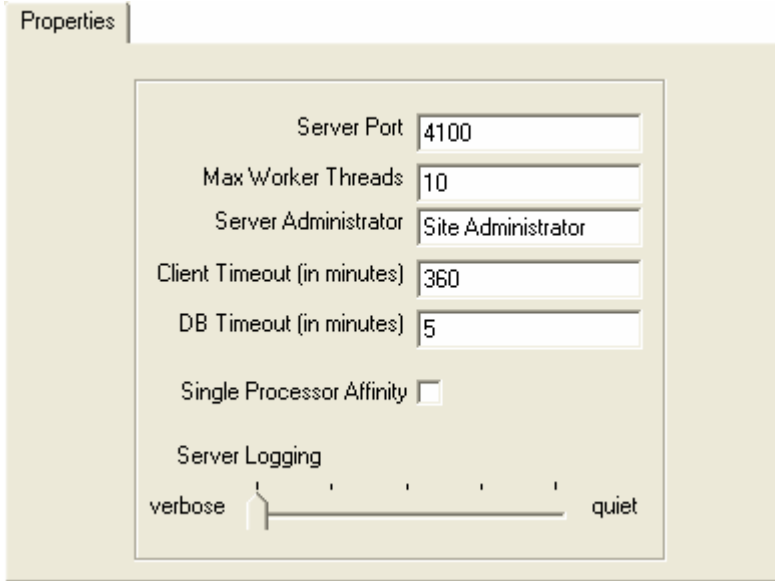
Selecting the Options item in the left pane will display screens that allow various server options to be set. There are four tabs to choose from:

1. **Properties**
2. **Connection Profiles**
3. **System Attributes**
4. **Licenses**



## Properties Tab

The **Properties** tab displays basic server settings, such as connection options. The following settings are available:



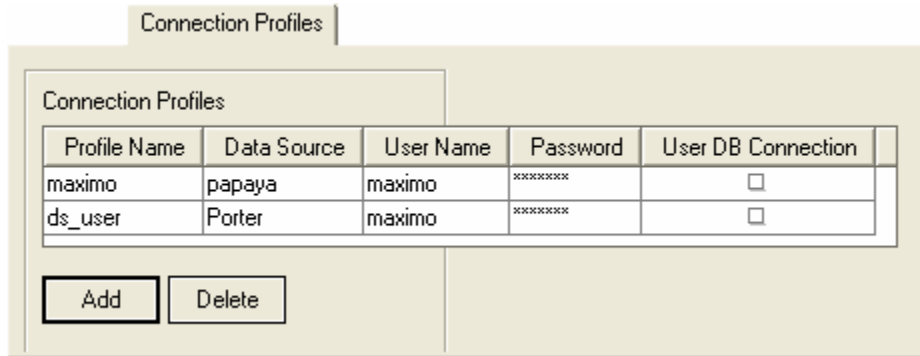
<b>Server Port</b>	<p>Specifies which TCP port the DataSplice Server will use to listen for incoming connections. This will not normally need to be changed. To obtain this information, contact your system administrator.</p> <p><b>Note:</b> Changes to this option will not be applied until the DataSplice Server is restarted.</p>
<b>Max Worker Threads</b>	This controls the maximum number of worker threads used by the DataSplice Server to handle remote connections.
<b>Server Administrator</b>	This is the name of the server administrator, such as <i>John Smith</i> . It is presented to remote users in case of certain problems so they can contact an administrator for assistance. This field is only a reference and is not used by DataSplice.
<b>Client Timeout</b>	This setting determines the amount of time, in minutes, before the DataSplice Server disconnects clients that have been idle. Setting this to zero disables the timeout feature.
<b>DB Timeout</b>	This setting specifies how long, in minutes, a database connection must be idle before being closed by the DataSplice Server. This can be helpful to minimize the total number of database connections kept open by the DataSplice Server. Setting this to zero disables the timeout feature.
<b>Single Processor Affinity</b>	By default, the DataSplice Server will run on all available processors on an SMP machine. Selecting this option will cause the server to only use a single processor on multi-processor machines. For certain hardware configurations, running on multiple processors can cause issues under heavy loads.
<b>Server Logging</b>	<p>The DataSplice Server creates log files in the <i>Logs</i> directory of the main DataSplice installation path. By default, this would be <i>C:\Program Files\DataSplice\Logs</i>. The slider controls how much detail is included in the logs.</p> <p><b>Verbose</b> logging will log <i>ALL</i> incoming and outgoing messages. This is often useful when attempting to diagnose a problem. <b>Quiet</b> logging only writes out</p>

	critical information and errors. The default setting, in the middle, is usually sufficient for logging all essential messages.
--	--

## Connection Profiles Tab

The DataSplice Server uses connection profiles to define the settings used to connect to external data sources. Each view specifies a particular connection profile to use.

Connection profiles allow settings for multiple views to be quickly changed, since the details only need to be modified for the connection profile once. This is particularly useful for moving from a test to a production setup, for example.



Profile Name	Data Source	User Name	Password	User DB Connection
maximo	papaya	maximo	*****	<input type="checkbox"/>
ds_user	Porter	maximo	*****	<input type="checkbox"/>

Buttons: Add, Delete

### To add a connection profile:

1. Click the **Add** button. A new profile will be added at the bottom of the list.
2. Change the **Profile Name** to something meaningful for the particular application.
3. Double click the **Data Source** field and select the desired **ODBC DSN** from the list. Please refer to the **ODBC reference page** for more information about managing ODBC Data Sources.
4. In the **User Name** field, enter the account name needed to connect to the database.
5. In the **Password** field, enter the password associated with the account name.
6. Check the **User DB Connection** field if the server should pass the user credentials supplied by the remote client to the database instead of using the **User Name** for this connection profile. This is useful in setups that use an external authentication domain and would like to enforce existing database permissions through DataSplice.

## System Attributes Tab


This defines attributes that will be available system-wide. These attributes are useful to define default behavior that can then be overridden on a user or group basis as needed.

System Attributes

System-wide Attributes


Name	Value	
DS_AUTO_COMMIT	<input checked="" type="checkbox"/>	
DS_AUTO_RELOAD	<input type="checkbox"/>	
DS_ONLINE_USER	<input checked="" type="checkbox"/>	
DS_PERSIST_PASSWORD	<input checked="" type="checkbox"/>	
MAIL_SERVER	mail.datasplice.com	

### To set a system attribute:

1. Select one of the system attributes from the **Name** dropdown list, or type a name to create a custom attribute.
2. Enter the value to associate with the attribute in the **Value** field.
3. Click the **Update** button  or change focus from the pane.
4. A dialog box will appear to confirm changes.

### Example:

#### To create a system level attribute that defines the default SMTP email server:

1. In the **Name** column, enter the name for the email server attribute, e.g. *MAIL\_SERVER*.
2. In the **Value** column, enter the name of the SMTP email server, e.g. *mail.datasplice.com*.
3. Click the **Update** button .
4. The value *#{MAIL\_SERVER}* can now be used in email actions to reference this attribute. See the section about **Events** for more information.

For more details about using attributes, see the section describing **Attributes**.

## Licenses Tab

This tab enables the administrator to enter a license key for the DataSplice server, as well as define license pools to partition the available licenses.

### License Usage

DataSplice uses a *concurrent licensing* scheme. This means that whenever remote clients are being used they will consume a single license. This occurs when a user is connected to the DataSplice Server *or has obtained an offline session and is no longer connected*.

**Note:** Shutting the DataSplice Server down and restarting it does not release licenses acquired through offline sessions. The administrator can revoke the offline license using the **Status** tab, but in doing so, will prevent the offline client from being able to reconnect with that session and commit their changes. The only reason for revoking offline sessions is if the session has been orphaned and the administrator needs to free up the offline license.

Licenses

Server License

Licensed Clients       Expiration Date

Shared Pool

License Pools	Features
License Pool Name	Licenses
DataSplice::Administrators	5

DEMO  
OFFLINE

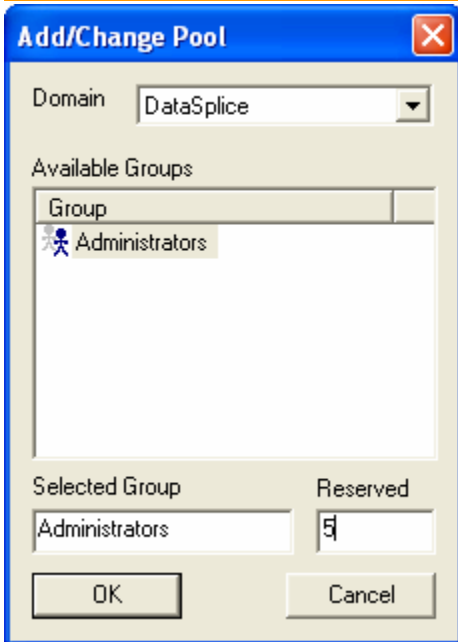
<b>Server License</b>	<p>If the license key is missing or expired, the Administration Client will have limited functionality available. In the <b>Server License</b> field, enter the license key provided to you by DataSplice. It is recommended that you copy and paste the key value, rather than attempting to enter it manually, because of the likelihood of mistyped characters. Once a new license key is entered, be sure to click the <b>Update</b> button . You will be prompted to reconnect for the changes to take effect. Once you have reconnected, the product’s functionality will be unlocked.</p>
<b>Licensed Clients</b>	<p>This is the allowed number of concurrent users, which is controlled by the license key. To purchase additional licenses, please contact DataSplice at <a href="mailto:datasplice@datasplice.com">datasplice@datasplice.com</a>.</p>

	<p><b>Note:</b> While you are connected to the DataSplice Server in the Administration Client, you are consuming a license. In addition, remember that shutting down the DataSplice Server and restarting it does not release licenses acquired through offline sessions. The administrator can revoke the offline license manually, but in doing so, will prevent the offline client from being able to reconnect with that session and commit their changes. The only reason for revoking offline licenses (sessions) is if the session has been orphaned and the administrator needs to free up the license. Please see the <b>Status</b> section for more information about manually revoking sessions.</p>
<b>Shared Pool</b>	<p>The Shared Pool indicates the number of remaining licenses that have not been reserved for use by a particular group. Before drawing from the shared pool, licenses will be obtained from the individual license pool to which the user belongs. If no licenses are available there, then the system will draw licenses from the shared pool. Initially, all licenses belong to the shared pool until they are reserved.</p>
<b>Expiration Date</b>	<p>This field shows the date, if any, on which the DataSplice license key will expire. Permanent license keys do not expire, and will say <i>NEVER</i> in this field. Demo license keys expire within 60 days of initial use, and will display a specific expiration date in this field.</p>
<b>License Pools</b>	<p>This section is used by the administrator to create license pools for the purpose of reserving fixed numbers of licenses for use by a specific group. See the following section for more information about setting up license pools.</p>

**To set up a license pool:**

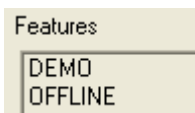
1. Click the **Add** button. This will open the **Add/Change Pool** dialog box.
2. Select the domain from the **Domain** drop-down box.
3. Select the desired group to associate with this license pool from the **Available Groups** list. The **Selected Group** box will display the current group selection. Enter the number of licenses to reserve for this pool in the **Reserved** box.
4. Click the **OK** button. The new pool will be added to the list of existing license pools, and the **Shared Pool** will be updated to indicate the remaining number of non-reserved licenses available.

If you wish to reserve licenses for users in multiple groups, create a new group and add the individual users, that you want reserved licenses for, to this new group. Then you can create a license pool for the new group with the desired number of licenses.



**Note:** It is possible to reserve more licenses than are available. This is indicated by a shared license pool number that is 0 or *negative*. If the shared license pool is zero or less, then you can either purchase more licenses and get a new license key or you can modify the number of licenses that are reserved for the license pools.

The **Features** box displays the enabled features for the DataSplice license key that was entered. There are currently three features: **DEMO**, **PERMANENT**, and **OFFLINE**.



**DEMO**

Demo license keys are for temporary evaluation purposes. As such, they have an expiration date, not to exceed 60 days from initial use.

**PERMANENT**

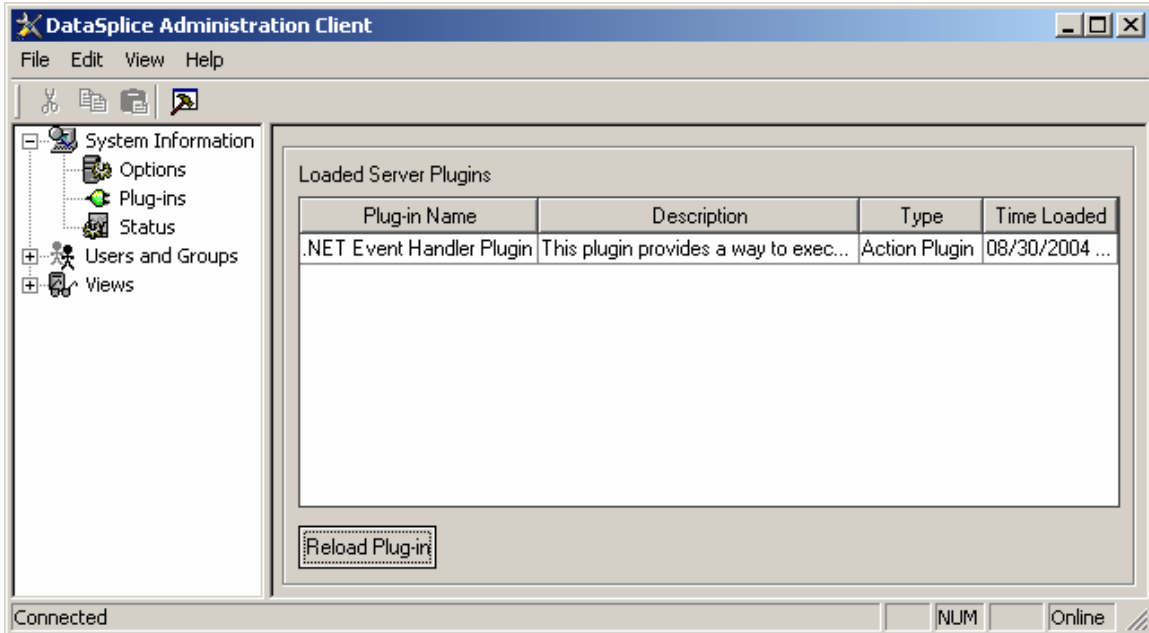
Permanent license keys are designed to provide full functionality of the DataSplice software for an unlimited duration of time.

**OFFLINE**

This license key feature enables DataSplice to operate in offline mode in addition to the regular online functionality. This feature can be used in conjunction with both demo and permanent license keys.

## Plug-ins

The **Plug-ins** screen shows information about the loaded server plug-ins, such as authentication modules or barcode printer plug-ins. The only option available in this screen is **Reload Plug-in**, which will have the plug-in reload itself. The effects of this will depend on the particular plug-in being reloaded.

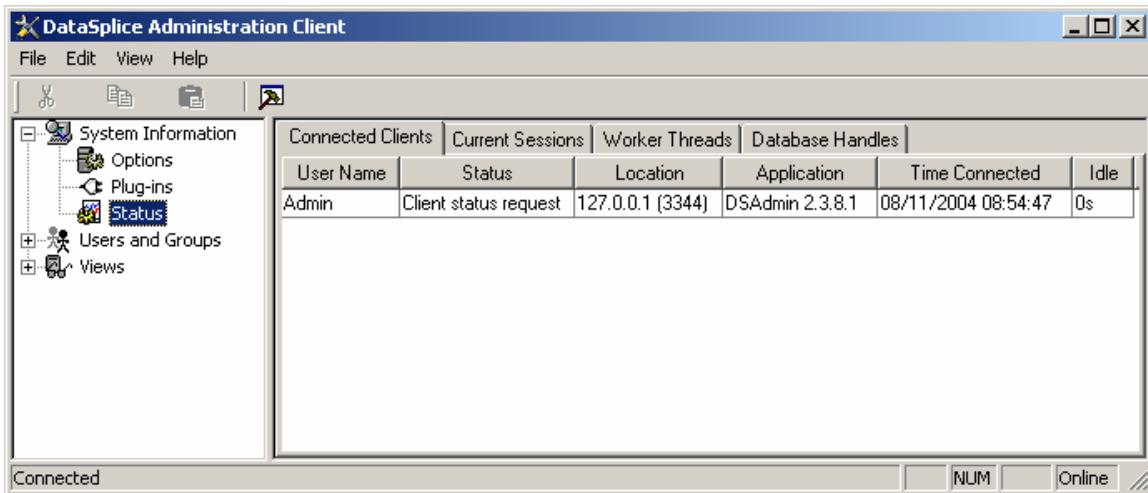


<b>Plug-in Name</b>	The name of the plug-in
<b>Description</b>	A description of the plug-in
<b>Type</b>	The type of plug-in
<b>Time Loaded</b>	The date and time that the plug-in was loaded

## Status

Selecting the **Status** option in the left pane will display useful information about the current status of the DataSplice Server. There are four status tabs available:

1. **Connected Clients**
2. **Current Sessions**
3. **Worker Threads**
4. **Database Handles**



## Connected Clients Tab

This displays the list of active client connections with the server. The following information is displayed:

Connected Clients					
User Name	Status	Location	Application	Time Connected	Idle
WINSTON	Idle	127.0.0.1 (2407)	DataSplice 2.3.8.1	08/16/2004 08:26:19	21m, 3s
Admin	Client status request	127.0.0.1 (2409)	DSAdmin 2.3.8.1	08/16/2004 08:26:47	0s

<b>User Name</b>	This shows the login name associated with the connection. This may be a name created just for DataSplice, or it may be part of an external domain. This name is created in the <b>Users and Groups</b> section.
<b>Status</b>	This displays any current activity of the connection, such as requesting data or updating a record. DataSplice displays any changes to this information immediately.
<b>Location</b>	This displays the network address (IP Address and Port) of the remote connection.
<b>Application</b>	This displays the application name and version of the DataSplice client being run by the user.
<b>Time Connected</b>	This displays the date and time when the remote connection was established.
<b>Idle</b>	This displays the length of time since the remote connection requested any information or performed an action against the DataSplice Server.

By right clicking on a field in the grid, several actions can be performed:

### Write Client

This will allow the administrator to send a message to the selected client.

### Write All

This is similar to **Write Client**, except the message will be sent to all active connections.

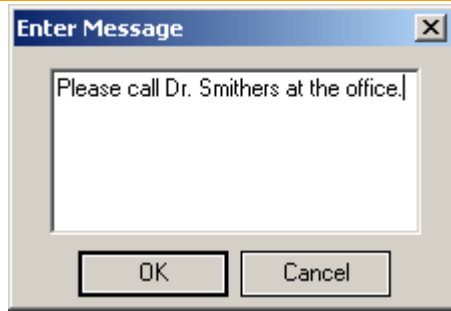
### Disconnect Client

This will close the selected connection. This can be used to disconnect users that have been idle for too long and free up licenses if needed.

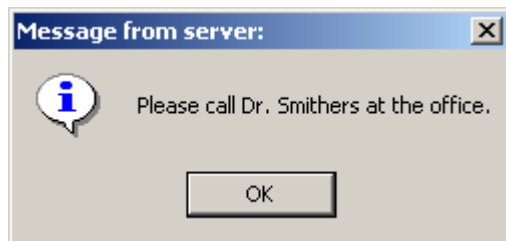
## Writing to a Client / Writing to All

### To send a message to all users or a particular user:

1. Right click on the user of interest in the tab.
2. From the menu, select **Write Client** to send a message to that particular user or **Write All** to send a message to all users.
3. The **Enter Message** dialog box appears.
4. Enter the message exactly as it is to be viewed by the user(s).



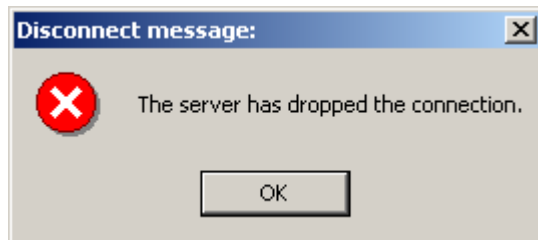
5. Click **OK**.
6. The message will appear as a pop-up box to the particular user if you selected **Write Client** or to all users if you selected **Write All**.



## Disconnecting a Client

### To disconnect a user:

1. Right click on the user of interest.
2. Select **Disconnect Client** from the menu.
3. The user will be removed from the **Connected Clients** list.
4. An error message will appear as a pop-up box on the user's device, notifying them that the server has dropped the connection.



## Current Sessions Tab

This displays a list of the current sessions on the server. This includes both online and offline sessions. The following information is listed:

Current Sessions					
Session Name	User Name	Time Created	Time Accessed	Duration	Session Type
WINSTON-1092666...	MAXIMO::WINSTON	08/16/04 08:26:32	08/16/04 08:26:32	3m, 49s	Online
Admin-1092666407	DataSplice::Admin	08/16/04 08:26:47	08/16/04 08:26:47	3m, 34s	Online

<b>Session Name</b>	This is the unique, server-generated name for the session, which is based off the user name and the time the session was created.
<b>User Name</b>	The login name of the user associated with the session.
<b>Time Created</b>	The date and time the session was initially created.
<b>Time Accessed</b>	The date and time of the last connection with the session.
<b>Duration</b>	The amount of time that has elapsed since the time of the last connection with the session.
<b>Session Type</b>	This indicates whether the session is <i>online</i> or <i>offline</i> .

By right-clicking a session, the administrator can choose **Revoke Session** to delete the session if desired. This can be used to free up the license for an orphaned session. Be careful, once a session has been revoked, the remote user will not be able to reconnect with the associated offline session and any uncommitted changes will be lost.

**Note:** Only offline sessions that do not have an active connection can be revoked. Otherwise, choose **Disconnect Client** from the **Connected Clients** tab.

## Worker Threads Tab

This displays information about the current worker threads on the server, which are used to process requests from remote clients. The following fields are displayed:

Worker Threads		
Id	Status	Duration
0	Thread status request	0s
1	Idle	7m, 58s
2	Idle	2d, 23:09:50
3	Idle	2d, 23:09:50
4	Idle	2d, 23:09:50
5	Idle	2d, 23:09:50
6	Idle	2d, 23:09:50
7	Idle	2d, 23:09:50
8	Idle	2d, 23:09:50
9	Idle	2d, 23:09:50

<b>Id</b>	The identification number assigned by DataSplice to the worker thread.
<b>Status</b>	Displays the current activity of the thread. If the thread is not processing any request, it will display <i>Idle</i> .
<b>Duration</b>	Indicates how long the thread has been in its current state.

## Database Handles Tab

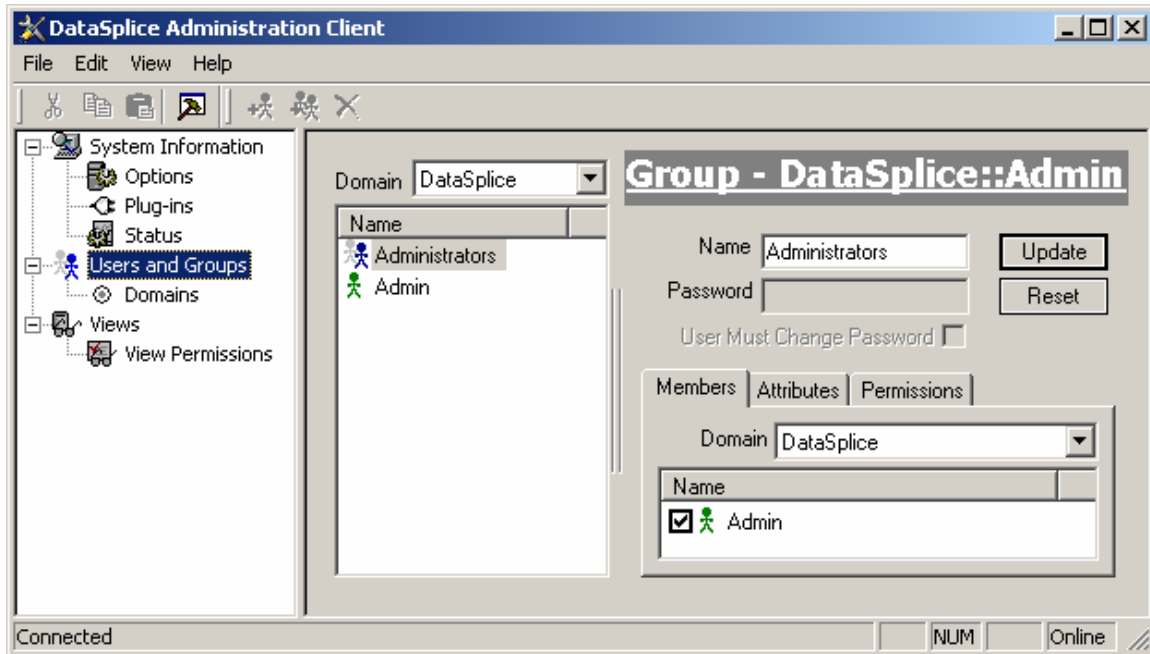
This displays a list of all the connections to various databases the server currently has open. The following information is displayed:

Database Handles				
Data Source	User Name	Cursor Lib	Recordsets	In Use
abbyale	maximo	Yes	0	No
abbyale	maximo	No	0	No
abbyale	WINSTON	No	0	No
abbyale	MAXIMO	No	0	No

<b>Data Source</b>	This displays the ODBC DSN of the connection.
<b>User Name</b>	This shows the login name that was used to connect to the data source.
<b>Cursor Lib</b>	This indicates whether or not the ODBC Cursor Library is being used by the database handle.
<b>Recordsets</b>	This indicates how many recordsets are currently open through this database connection.
<b>In Use</b>	This field indicates if the connection is currently being used to communicate with the data source.

# Users and Groups

The **Users and Groups** screen allows you to view and/or change user and group configurations. The screen looks like the following:



## Users

User accounts created with DSAdmin are used by all of the applications in the DataSplice Suite. Users may belong to multiple groups, or no groups, and any permissions set for individual users will override the permissions specified for the group(s) to which that user belongs.

## Groups




Groups manage access to views, data displayed in those views, and permissions to make changes to the data. The same view may give supervisors permission to change a description field, while workers may have permission to view the field only. Users may belong to more than one group, and individual permissions set will override the group permissions.

## Domains

Many companies already have systems set up with user names and passwords for a variety of systems, including Windows Networking and database applications such as MAXIMO. DataSplice is able to use these user names and passwords, thus decreasing clerical maintenance by keeping only one set of users and avoiding redundant data entry. Administration of the users and groups in the domain must take place at the source of the domain. For example, a domain created from Windows Networking must have all users created, deleted, and added to groups with Windows tools. DataSplice provides tools for managing users and groups in the DataSplice domain. For further information regarding domains, please see the following **Domains** section.

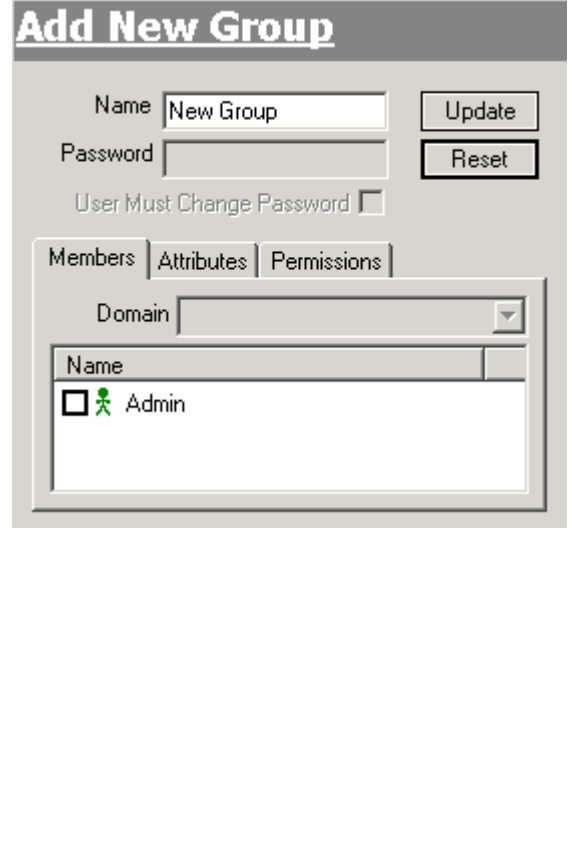


## Creating Users

Users may be added to the DataSplice domain. Users for other domains must be added according to the specifications of the other domain.

	<p><b>To create a user:</b></p> <ol style="list-style-type: none"> <li>1. Open the <b>Users and Groups</b> section.</li> <li>2. Change the focus from the left pane to the right pane by clicking anywhere in the right pane. The <b>Add User</b> and <b>Add Group</b> icons in the toolbar will change from grey to color when the focus is changed. This means that the buttons are now enabled.</li> <li>3. Select a domain from the <b>Domain</b> dropdown list. <i>DataSplice</i> is the default domain.</li> <li>4. Click the <b>Add User</b> button  in the toolbar.</li> <li>5. Enter the new user name in the <b>Name</b> field.</li> <li>6. Enter the new password in the <b>Password</b> field.</li> <li>7. To force a new user to change his password upon first login, check the <b>User Must Change Password</b> checkbox. This box may also be checked at a later time.</li> <li>8. A list of groups will appear in the <b>Members</b> box. Check any groups to which the user will belong.</li> <li>9. Click the <b>Update</b> button .</li> </ol>
---	--

The **Attributes** tab is used to customize DataSplice for this particular user. For information about attributes, see the **User Attributes** section.


## Creating Groups

	<p><b>To create a group:</b></p> <ol style="list-style-type: none"> <li>1. Open the <b>Users and Groups</b> section.</li> <li>2. Change the focus from the left pane to the right pane by clicking anywhere in the right pane. The <b>Add User</b> and <b>Add Group</b> icons in the toolbar will change from grey to color when the focus is changed. This means that the buttons are now enabled.</li> <li>3. Select a domain from the <b>Domain</b> dropdown list. <i>DataSplice</i> is the default domain.</li> <li>4. Click the <b>Add Group</b> button  in the toolbar.</li> <li>5. Enter the name of the new group in the <b>Name</b> field.</li> <li>6. Users will appear in the <b>Members</b> box. Check all users to be added to the group.</li> <li>7. Click the <b>Update</b> button .</li> </ol>
--	--


The **Attributes** tab is used to customize DataSplice for this particular group. For information on how to define attributes for groups, see the **Group Attributes** section.

## Deleting Users & Groups

### To delete a user:


1. Open the **Users and Groups** section.
2. Change the focus from the left pane to the right pane by clicking anywhere in the right pane. The **Add User** and **Add Group** icons in the toolbar will change from grey to color when the focus is changed. This means that the buttons are now enabled.
3. Select a domain from the **Domain** dropdown list. *DataSplice* is the default domain.
4. Highlight the user to be deleted.
5. Click the **Delete** button  in the toolbar.

### To delete a group:

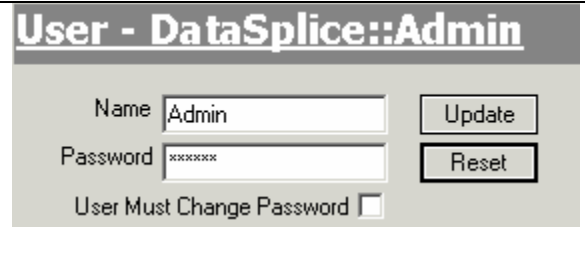
1. Open the Users and Groups section.
2. Change the focus from the left pane to the right pane by clicking anywhere in the right pane. The **Add User** and **Add Group** icons in the toolbar will change from grey to color when the focus is changed. This means that the buttons are now enabled.
3. Select a domain from the **Domain** dropdown list. *DataSplice* is the default domain.
4. Highlight the group to be deleted.
5. Click the Delete button  in the toolbar.

## User & Group Properties

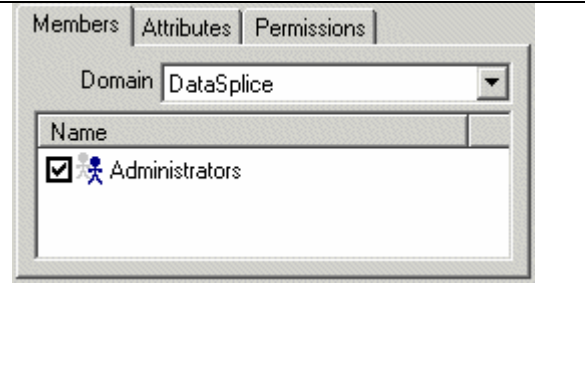
### Name List

	<p>This list displays all the groups and users for the selected domain.</p>
---	---

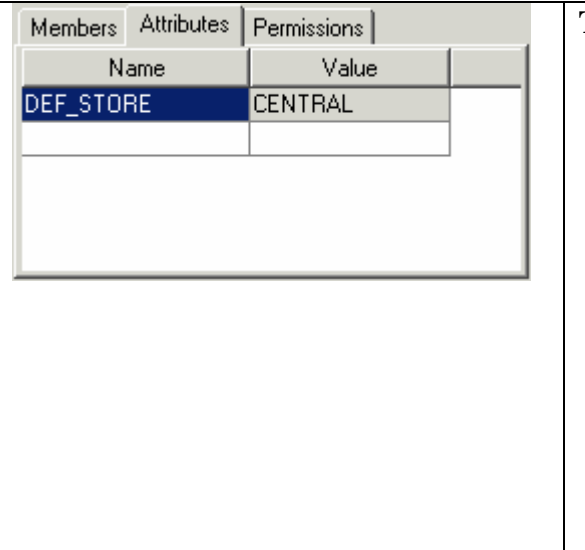
### Name and Password Fields

	<p>The <b>Name</b> field indicates the user name for the selected user. The <b>Password</b> field is used to specify the password for DataSplice domain users. The <b>User Must Change Password</b> checkbox is used to force users to change their password the first time they connect.</p>
---	---

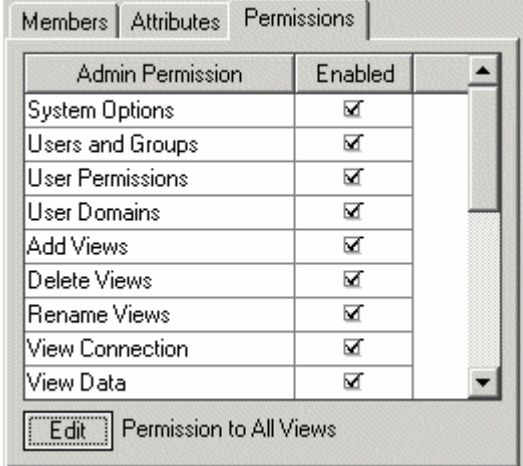
### Members Tab

	<p>The administrator can add (or remove) users from groups by checking (or unchecking) the checkbox next to the desired group in the list.</p> <p><b>Domain</b> – This field indicates which domain you want to select members from.</p> <p><b>Name</b> – This field indicates the name of the user or group you are selecting.</p>
--	---

### Attributes Tab

	<p><b>To set a user attribute:</b></p> <ol style="list-style-type: none"> <li>1. Highlight the user name from the <b>Users and Groups</b> section.</li> <li>2. Click on the <b>Attributes</b> tab.</li> <li>3. Select one of the attributes from the <b>Name</b> dropdown list which contains a list of recently used attributes, or type a name to create a new attribute.</li> <li>4. Enter the value to associate with the attribute in the <b>Value</b> field.</li> <li>5. Click the <b>Update</b> button or change focus from the pane.</li> <li>6. A dialog box will appear to confirm changes.</li> </ol>
---	--

### Permissions Tab

	<p><b>To set an administrator user's permissions:</b></p> <ol style="list-style-type: none"> <li>1. Highlight the user name from the <b>Users and Groups</b> section. The user must be a member of the Administrator group.</li> <li>2. Click on the <b>Permissions</b> tab.</li> <li>3. Double click the <b>Enable</b> checkbox to enable or disable a particular permission for the selected user.</li> <li>4. The <b>Edit</b> button is used to assign Admin permissions to views. E.g. specify which views a user can see and/or change.</li> <li>5. Click the <b>Update</b> button to save the user permission changes.</li> </ol>
---	---

**Note:** When an administrator user has been granted rights to change **User Permissions**, this includes themselves. This means that, when granted **User Permissions** they essentially have super user rights, because they can change their own permission set.

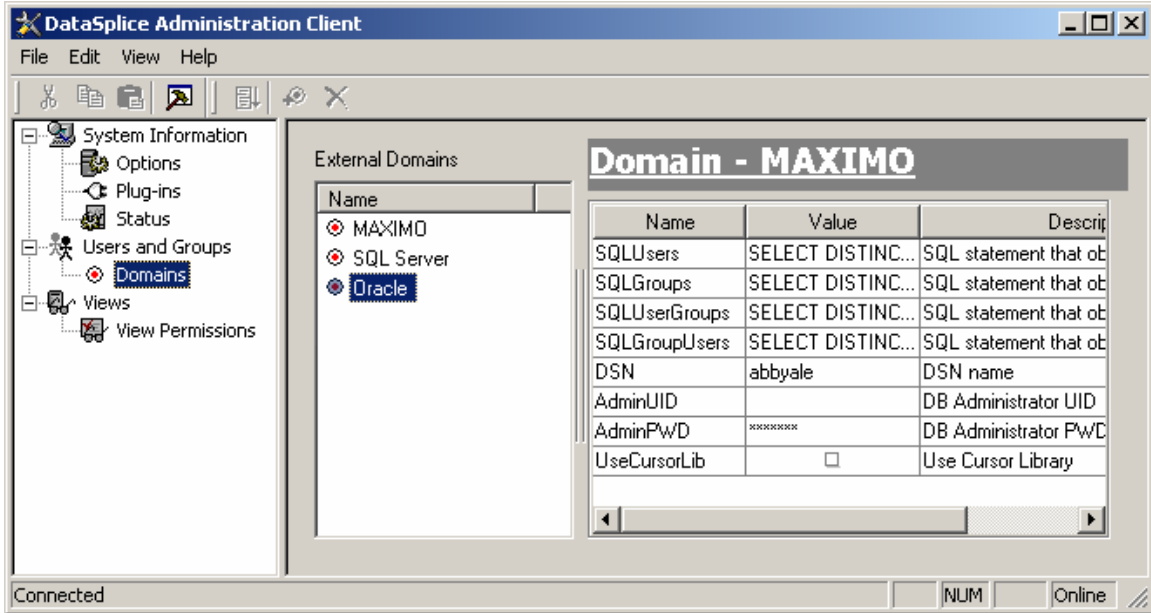
### Admin Permissions List

<b>System Options</b>	Enables the Admin user to view and change the system settings found in the System Information section.
<b>Users and Groups</b>	Enables the Admin User to create, delete, and modify users and groups.
<b>User Permissions</b>	Enables the Admin User to modify the permissions for other admin users (and themselves).
<b>User Domains</b>	Enables the Admin User to create, delete, and modify user domains.
<b>Add Views</b>	Enables the Admin User to create views.
<b>Delete Views</b>	Enables the Admin User to delete views.
<b>Rename Views</b>	Enables the Admin User to rename views.
<b>View Connection</b>	Enables the Admin User to modify the connection settings for a view(s).
<b>View Data</b>	Enables the Admin User to change the SQL statement for a view(s).
<b>View Fields</b>	Enables the Admin User to change individual field attributes for a view(s).
<b>View Attributes</b>	Enables the Admin User to modify the view's attribute settings.
<b>View Criteria</b>	Enables the Admin User to add, remove, and modify the filtering and sorting criteria for a view(s).
<b>View Relationships</b>	Enables the Admin User to add, remove, and modify the parent/child relationships between views. E.g. setup view navigation, offline data caching, etc.
<b>View Events</b>	Enables the Admin User to add, remove, and modify the actions performed in association with DataSplice events.
<b>View Permissions</b>	Enables the Admin User to change the default view permissions. E.g. select, insert, update, and delete.

<b>Override Connection</b>	Enables the Admin User to override connection information for DataSplice users, but not base view connection information.
<b>Override Fields</b>	Enables the Admin User to override field properties for DataSplice users, but not base view field information.
<b>Override Attributes</b>	Enables the Admin User to override attributes for DataSplice users, but not base level attributes for the view.
<b>Override Criteria</b>	Enables the Admin User to override view criteria for DataSplice users, but not base view criteria settings.

## Domains

Selecting the **Domains** option, under **User and Groups**, will display the list of available domains. The following information is accessible:



## Columns


<b>Name</b>	The Name field contains the name assigned to the information contained in the <b>Value</b> field.
<b>Value</b>	The Value field contains the SQL data used to extract users and groups. This field will be filled in according to the template selected. Values may be customized by double clicking in the field. <b>Example:</b> SELECT DISTINCT username, " FROM maxusergroups
<b>Description</b>	The <b>Description</b> field has a short description of what is contained in the <b>Value</b> field. This is an informational field only and is not used by DataSplice.

The **Value** fields in domain can be modified to configure the domain to search for users and groups based on a company's individual needs. The following settings are available to change:

<b>SQLUsers</b>	This field indicates the SQL statement to use for determining the users in this domain. The following SQL statement is an example of how to query Oracle for the list of users:  SELECT DISTINCT username, " FROM dba_users  <b>Note:</b> The %VAR% syntax is required by DataSplice to perform variable substitution for the login name of the user being validated.
-----------------	---

<b>SQLGroups</b>	<p>This field indicates the SQL statement to use for determining the groups in this domain. The following SQL statement is an example of how to query Oracle for the list of groups (roles):</p> <pre>SELECT DISTINCT role, " FROM dba_roles</pre>
<b>SQLUserGroups</b>	<p>The following SQL statement is an example of how to query Oracle for the groups (roles) a user is a member of:</p> <pre>SELECT DISTINCT granted_role, grantee FROM dba_role_privs WHERE grantee = %VAR%</pre> <p><b>Note:</b> The %VAR% syntax is required by DataSplice to perform variable substitution for the login name of the user being validated.</p>
<b>SQLGroupUsers</b>	<p>This field indicates the SQL statement to use for determining the users that are members of a group. The following SQL statement is an example of how to query Oracle for the users that are a member of a particular group (role):</p> <pre>SELECT DISTINCT grantee, granted_role FROM dba_role_privs WHERE granted_role = %VAR%</pre> <p><b>Note:</b> The %VAR% syntax is required by DataSplice to perform variable substitution for the group (role) being validated.</p>
<b>DSN</b>	This field indicates which ODBC data source to use.
<b>AdminUID</b>	This field indicates the login name to use when connecting to the data source.
<b>AdminPWD</b>	This field indicates the password for the login name specified in <b>AdminUID</b> .
<b>UseCursorLib</b>	This field indicates whether or not to use the ODBC cursor library.

**To create an external domain:**

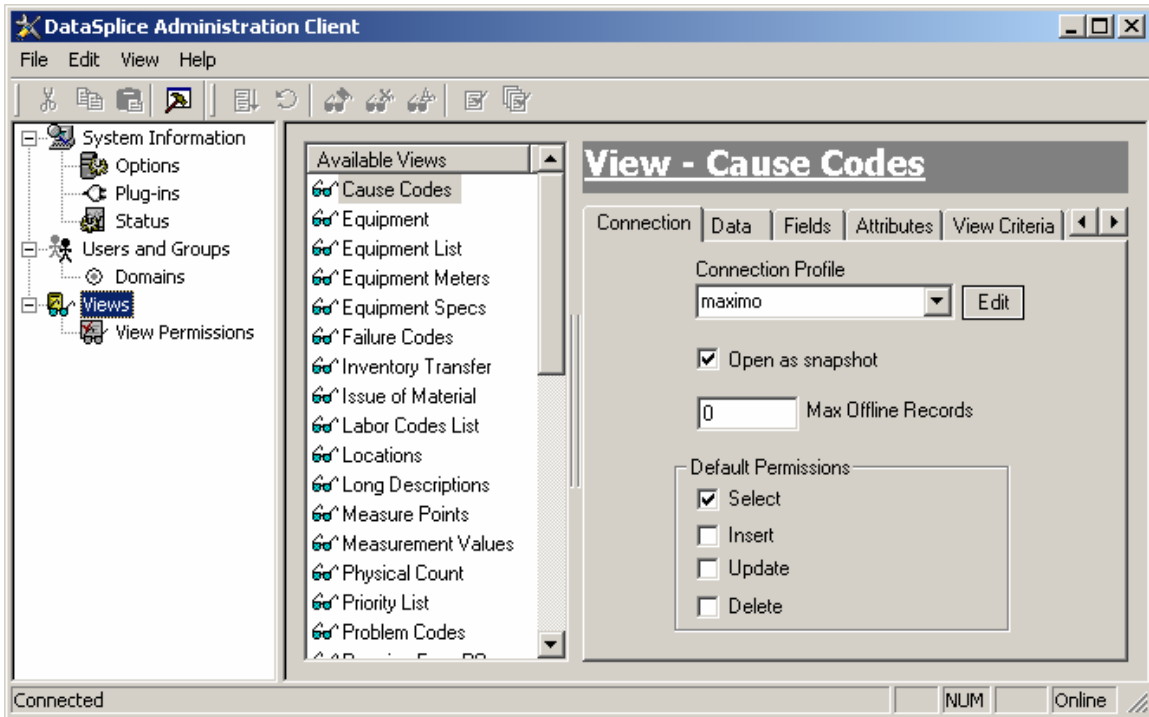
1. Expand the **Users and Groups** section in the left pane.
2. Open the **Domains** section.
3. Click the **Add Domain** button  in the toolbar.
4. The **Create External Domain** dialog box will appear.
5. Enter a name for the domain.
6. Select an **Authentication Module**.
7. A list of templates will appear.
8. Select the appropriate template.
9. Click **OK**.

The template domain will now appear in the list of domains. Changes must be made to the template in order to populate the domain.


**Note for MAXIMO users:** In most cases, every field except the **DSN**, **UID**, and **PID** will be set up by the template.

# Views

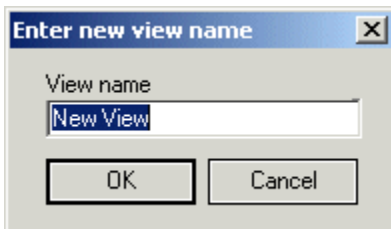
Selecting **Views** in the left-hand pane brings up numerous options which can be seen and altered:



## Creating Views


To create a new view, click on the **Create View** button .

The **Enter new view name** dialog box opens up to enter the name of new view.

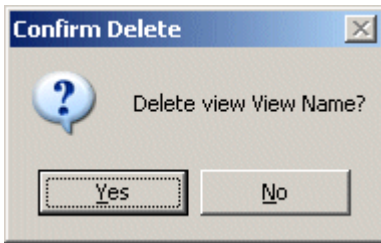


After entering the name of the view, click the **OK** button.

## Deleting Views


To delete an existing view, click on the **Delete View** button .

A message box will appear to confirm the deletion of the view.

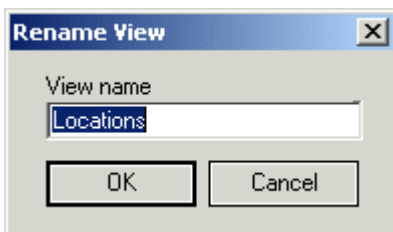


To finish deleting the view, click **Yes**.

## **Renaming Views**

To rename an existing view, click on the **Rename View** button .

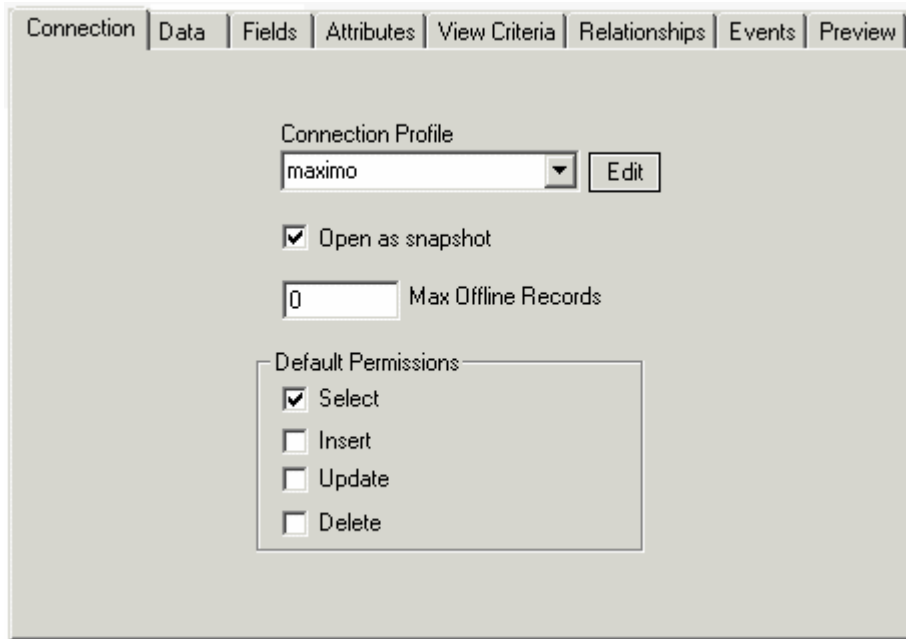
The **Rename View** dialog opens to enter the new name for the view.



After entering the new name, click the **OK** button.

## Connection Tab

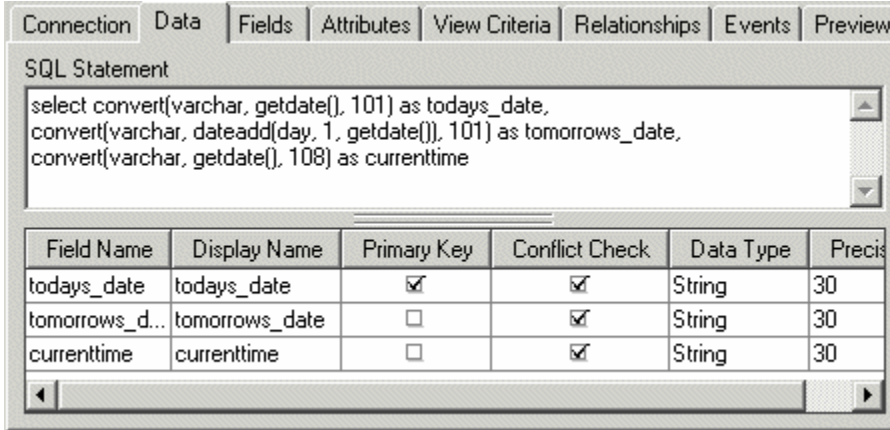
The **Connection** tab contains information regarding how to access the database.



<b>Connection Profile</b>	This is the connection profile to use for this view. See the Connection Profiles Tab section for information on connection profiles.
<b>Open as snapshot</b>	<p>The Open as snapshot option creates an editable, joined-table view. ODBC does not allow dynasets on joined views. Opening the view as a snapshot ensures that the view will not automatically update, but rather wait for a request. If this selection is not active and the view incorporates multiple tables, the view will be read-only by default.</p> <p><b>Note for Oracle users:</b> If a join produces unique records from the primary keys. Oracle will accept the join. If not, the view will fail.</p>
<b>Max offline records</b>	Enter the maximum number of records allowed offline. 0 (or less) is treated as <i>unlimited</i> .
<b>Default permissions</b>	<p>The <b>Default permissions</b> define the type of actions that can be performed while accessing the view for all users and groups. These permissions may be overridden for specific users and groups in the <b>View Permissions</b> section.</p> <ul style="list-style-type: none"> <li>• Select – Allows read-only display of records.</li> <li>• Update – Allows records to be modified.</li> <li>• Insert – Allows records to be inserted.</li> <li>• Delete – Allows records to be deleted.</li> </ul>

## Data Tab

The **Data** tab contains the SQL statement and field list that defines what data (fields) will be included in the view.



<b>SQL Statement</b>	Enter the SQL statement that will be used to access the database. While creating views for use with DataSplice, keep in mind that the first field in the SQL statement that is marked as searchable in the <b>Fields</b> tab is the field used as the default filter field in the DataSplice Remote Client application.
----------------------	---

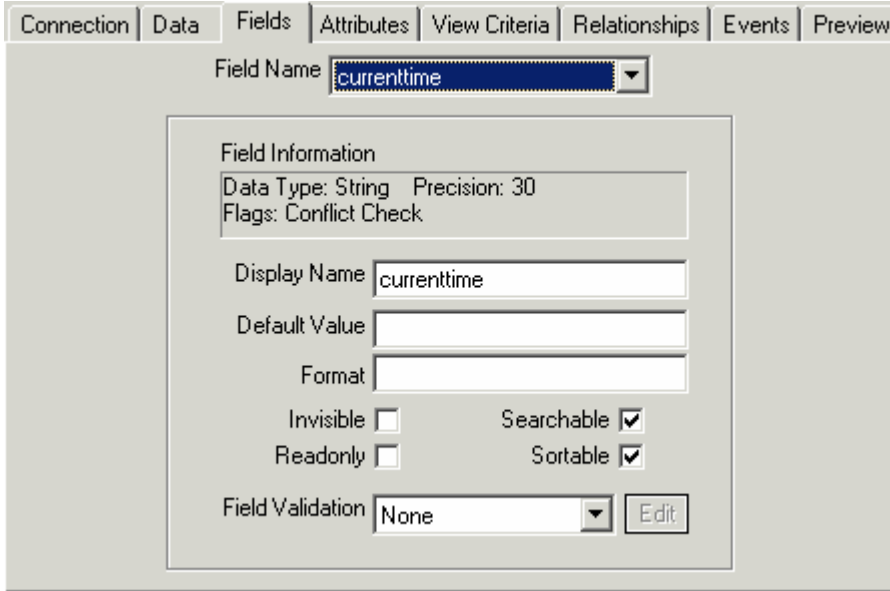
### Field List

The field list displays the fields (columns) of data returned from the SQL statement, along with the data type and precision. The remaining columns are defaulted to an initial set of values, but are changeable by the administrator.

<b>Field Name</b>	The field name is the name of the field as it appears in the SQL statement or database table. This field is filled in by DataSplice.
<b>Display Name</b>	This is the name to display in the column header on the Remote Client. The display name provides a meaningful field name to the end user. To change a display name, double click in the <b>Display Name</b> field and enter the new name. The display name may consist of any combination of numbers, letters, spaces, and special characters.
<b>Primary Key</b>	Checking this option makes the selected field part of DataSplice’s primary key for this view. The primary key for each view is used to determine which, if any, records will be cached offline. Also, the primary key is used during <i>insert</i> , <i>update</i> , and <i>delete</i> events.
<b>Conflict Check</b>	The <b>Conflict Check</b> box indicates whether or not DataSplice should evaluate this field before attempting data modifications to see if a potential conflict exists.
<b>Data Type</b>	The <b>Data Type</b> field contains the type of data stored in the database. This field is filled in by DataSplice.
<b>Precision</b>	The precision field displays the number of characters stored in the field as defined in the database, or returned by the ODBC driver. This field is filled in by DataSplice.

## Fields Tab

The **Fields** tab contains parameters which determine the appearance and function of the DataSplice fields.



The screenshot shows the 'Fields' tab in the DataSplice Administration Client. The 'Field Name' dropdown is set to 'currenttime'. The 'Field Information' box displays 'Data Type: String Precision: 30' and 'Flags: Conflict Check'. Below this, there are input fields for 'Display Name' (currenttime), 'Default Value', and 'Format'. There are also checkboxes for 'Invisible' (unchecked), 'Searchable' (checked), 'Readonly' (unchecked), and 'Sortable' (checked). At the bottom, the 'Field Validation' dropdown is set to 'None', and there is an 'Edit' button.

<b>Field Name</b>	The dropdown list contains all of the fields named in the SQL statement of the <b>Data</b> tab. These fields are also displayed in table format in the <b>Data</b> tab.
<b>Field Information</b>	This display-only box indicates the selected field's data type, precision, and option flag settings.
<b>Display Name</b>	Indicates the value to use as the column header display in the clients. This value can also be set on the <b>Data</b> tab in the <b>Display Name</b> column.
<b>Default Value</b>	The default value will be automatically input into all new records created by DataSplice. This value can then be overridden by the user. If no default value is desired, this field should be left blank.
<b>Format</b>	The optional <b>Format</b> field applies a format to the data in the field for display purposes, such as displaying a date with a uniform appearance. For more information, please refer to the <b>Format Strings</b> page in the reference section.
<b>Invisible</b>	The <b>Invisible</b> checkbox designates the field as a non-display field. The field remains part of the view, but is not visible to users.
<b>Readonly</b>	The <b>Readonly</b> checkbox renders the field uneditable.
<b>Searchable</b>	The <b>Searchable</b> checkbox indicates whether or not a user can filter (search) the view on this field. While creating views for use with DataSplice, keep in mind that the first searchable field in the SQL statement of a view is the field used as the filter field in DataSplice.
<b>Sortable</b>	The <b>Sortable</b> checkbox indicates whether or not a user can sort the view on this field.
<b>Field Validation</b>	Setting up field validation supplies dropdown value lists for use during data entry, which helps to maintain data integrity. <ul style="list-style-type: none"> <li>• <b>None</b> – The default setting of None removes dropdown list behavior.</li> <li>• <b>List Type</b> – The List Type field validation supplies the field with a static (fixed) dropdown value list.</li> </ul>

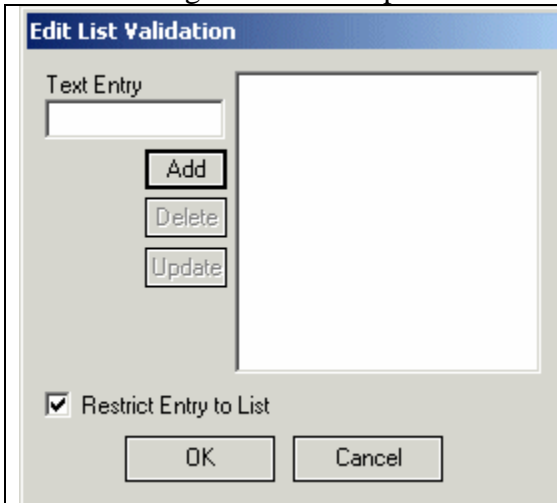
- **View Type** – The View Type field validation supplies the field with dynamic value selection. The actual value look-up is done using another DataSplice view.

## Creating Field Validation

Field validation helps to minimize the frequency of data entry errors by limiting the possible entries for a field to a specific, pre-defined set of values. The user then only has to choose one of these values, rather than type it by hand. This increases efficiency and data integrity.

### Validation Using a List

The following describes the process of using *a list* to validate a field.



#### To create field validation from a list:

1. Select the list entry in the **Field Validation** dropdown box.
2. Click the **Edit** button.
3. The **Edit List Validation** dialog box will open.
4. In the **Edit List Validation** dialog box, add entries to the list by typing the values into the **Text Entry** box in the upper left corner and clicking the **Add** button.
5. Click the **OK** button when finished adding entries to the list.
6. The list entries will appear to the user in the same order in which they are added.
7. Select the **Restrict Entry to List** checkbox to prevent the user from typing data into the field and forcing them to select a value from the list provided.

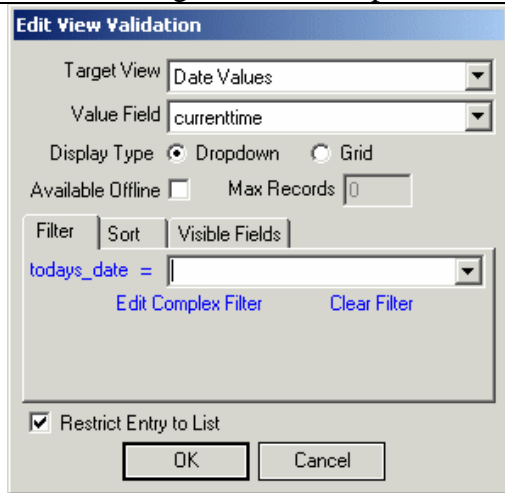
The static list can display additional information to the right of the list of values. To create a two-column list, separate the value from the additional information with the pipe character:

**Example:** “1 | 1<sup>st</sup> Value”

Both sides of the pipe will be displayed in the dropdown, but only the left side of the pipe will be inserted into the field.

## Validation Using a View

The following describes the process of using *another view* to validate a field.



### To create field validation from a view:

1. Select the **View** entry in the Field Validation dropdown box.
2. Click the **Edit** button.
3. The **Edit View Validation** dialog box will open.
4. In the **Edit View Validation** dialog box, do the following:
  - Specify which view to use for looking up data values by selecting the view from the **Target View** dropdown box.
  - Specify which field, from the target view, should be used as the value for data entry by selecting the field name from the **Value Field** dropdown box.
  - Select which **Display Type** option to use for this field validation:
    - **Dropdown** – Use a standard dropdown list for value selection.
    - **Grid** – Use a pop-up dialog for value selection. The pop-up window allows users to alter the default view filtering, in addition to paging through the possible values.
  - Use the **Available Offline** checkbox to specify whether or not the clients should cache this value list data offline.
  - On the **Filters** tab, specify the filter criteria to use for retrieving the initial set of values.
  - Select the **Restrict to List** checkbox if you want to prevent the user from typing data into the field and force them to select a value from the dropdown list or pop-up list provided.
5. Click the **OK** button when finished.

## Attributes Tab

The **Attributes** tab gives you access to the different name/value pairs which make up the attributes of the system. The **Attributes** tab looks like this:

Connection	Data	Fields	Attributes	View Criteria	Relationships	Events	Preview
			Name	Value			
			DS_DATE_FORMAT	%m/%d/%Y			
			DS_DISPLAY_CLASS	vertical			

Attributes enable arbitrary name/value pairs to be assigned to objects in DataSplice. This creates a list of *environment variables* that can be used for many different purposes, such as custom navigation. Attributes can define classes of views, such as *top-level* views, so a DataSplice client can display sets of views differently. For the standard DataSplice client, any top-level views defined are the only views initially available to the user as a start point. Any other views would only be available by navigating through a relationship. Attributes are also useful in providing additional information when performing database events on a record, such as supplying the current login name or the system time. They may be used as filter values by specifying a value using the format  $\${ATTRIBUTE\_NAME}$ . This allows both system and connection-defined attributes to be used in the filter, as well as values from the selected record in the parent view.

Attributes can be edited from the following areas in the Administration Client:

- **System Information Options** – Sets system-wide attributes
- **Users and Groups** – Sets attributes for a particular user or group
- **Views** – Sets both view and view access attributes
- **Relationships** – Sets attributes that define view relationships

## Precedence



1. **System** (such as user name, client version, etc.)
2. **View**
3. **User**
4. **Record** (when applicable)

## Built-in Attributes for the Current Connection

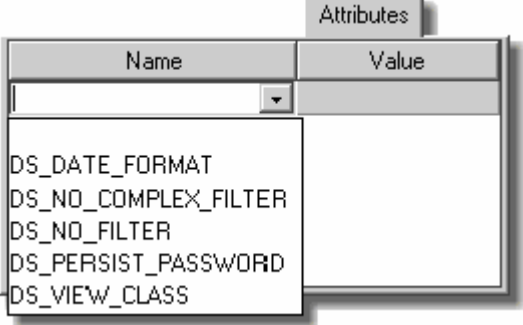

- **DS\_USER** – The name of the user associated with the connection (login user name).
- **DS\_DOMAIN** – The domain the user logged on with.
- **DS\_APP** – The name of the client application.

- **DS\_APP\_VERSION** – The version of the client application.
- **DS\_CURRENT\_TIME** – The current date and time for the DataSplice Client or Server.
- **DS\_EDIT\_TIME** – This will contain the initial time the record was edited, which makes things like time stamping offline modifications much easier.
- **DS\_VIEW** – The name of the current view (if any).

## Standard Attributes

- **DS\_AUTO\_COMMIT** – Specifies that record modifications are automatically committed when the record being modified loses focus (by clicking somewhere off the record). This can be defined system-wide, or for individual users and views. This works like the **Records > Apply Record Changes** menu item, which commits only the changes for the select records. Any other record(s) in the modification history need to be committed with the **Commit Changes** button  in the Remote Client application.
- **DS\_AUTO\_INPUT\_PANEL** – Controls the automatic display of the SIP panel for views on Windows CE devices. This defaults to *false*. If set to *true* for a view (or system level), it will cause the panel to be automatically displayed when needed.
- **DS\_AUTO\_RELOAD** – Causes the view to refresh its displayed records whenever it is redisplayed by clicking the **Back** button  in the Remote Client application.
- **DS\_CLEAR\_PASSWORD** – Causes the client to remove the password from the input text field on disconnect. This is useful when multiple users access the same device. This attribute will supersede the value of **DS\_PERSIST PASSWORD** – i.e., the value of the latter will have not effect if the client is set to clear passwords.
- **DS\_DATE\_FORMAT** – The default date format to use when displaying date fields that do not otherwise have a format string defined. If no value is given to this attribute, the default format is %H:%M:%S %a, %b %Y.
- **DS\_DISPLAY\_CLASS** – Defines how record sets are displayed in the Remote Client. The following values are allowed:
  - *grid* – Default to grid display
  - *vertical* – Default to vertical display
  - *grid\_only* – Only allow grid display
  - *vertical\_only* – Only allow vertical display
- **DS\_HIDDEN\_VIEW** – Indicates that the view should not show in the client view list. This is useful for support views that are never accessed directly by the user. This attribute overrides any view class that might be present for the view.
- **DS\_MESSAGE\_TIMEOUT** – This attribute setting controls the maximum time the client will wait for a server response to any message (default: *60 seconds*).  
**Note:** This is disabled during offline downloads, as the client will never time out while fetching offline records.

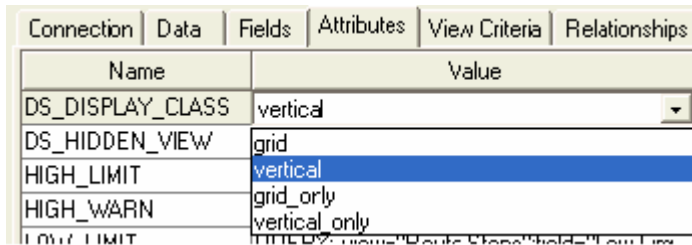
- **DS\_NO\_COMPLEX\_FILTER** – Determines whether creating and editing complex filters is allowed for any user or view that has this attribute defined.
- **DS\_NO\_FILTER** – Boolean value that specifies whether viewing and editing filters is allowed for any user or view that has this attribute defined.
- **DS\_NO\_OFFLINE** – Specifies whether or not a user can get an offline session. This attribute can be set at the **System** or **User** levels.
- **DS\_PERSIST\_DATA** – This attribute is used to control whether the data for views is flushed when sessions are changed. If this is *true*, the client will retain the data for the view so it does not need to be re-downloaded by the new session.
- **DS\_PERSIST\_PASSWORD** – Specifies whether or not the Remote Client application should remember the last user’s password along with the login name. This attribute can be set at the **System** or **User** levels.
- **DS\_PING\_INTERVAL** – This can be set to the number of seconds the client must be idle before a ping is issued prior to a new message being sent to the server (default: *30 seconds*). If this is set to *0* a ping will be sent before each message. The drawback to setting this to a low number is that it will slow down the overall performance of the client, since each ping takes a certain amount of time to succeed.
- **DS\_PING\_TIMEOUT** – This controls how long the client will wait for a ping response before timing out and closing the connection (default: *5 seconds*).
- **DS\_PURGE\_CHANGES\_ON\_CLEAR** – This will cause uncommitted changes to be purged from the device when users select **Records > Clear Screen** or when the clients disconnect.
- **DS\_RESET\_CRITERIA\_ON\_SELECT** – This will cause views to reset the last used criteria and fall back to the default criteria when navigating from the **Select Views** form.
- **DS\_SKIP\_OFFLINE\_UPDATE** – Clients can now specify views to skip when fetching offline records. This can significantly speed up downloads with large, static views. This attribute controls which views are skipped by default. The first time offline records are requested, the **DS\_SKIP\_OFFLINE\_UPDATE** attribute setting is not checked.
- **DS\_VIEW\_CLASS** – This defines view groups as different classes. This allows the client to categorize views when displaying views to select, such as only displaying views with a view class of *Inventory*.
- **DS\_WINDMILL\_UPDATE** – If set, the DataSplice Remote Client can invoke the Windmill software update client, automatically detecting newer versions of installed software. This can be used at the **System**, **User**, or **Group** level.

	<p><b>To set a view attribute:</b></p> <ol style="list-style-type: none"> <li>1. Select <b>Views</b> in the left pane.</li> <li>2. Select the view of interest.</li> <li>3. Click on the <b>Attributes</b> tab.</li> <li>4. Select one of the system attributes from the <b>Name</b> dropdown list, or type a name to create a custom attribute.</li> <li>5. Enter the value to associate with the attribute in the <b>Value</b> field.</li> <li>6. Click the <b>Update</b> button  or change focus from the pane.</li> </ol>
---	--

**Examples:**

**To set a view’s DS\_DISPLAY\_CLASS to single record mode:**

1. Select **DS\_DISPLAY\_CLASS** in the **Name** dropdown list.
2. Select *vertical* from the **Value** column.



3. Click the **Update** button  or change focus from the pane.

**To create a custom view attribute that supplies a high-limit value in relation to measurement points along a route:**

1. In the attribute’s **Name** column, enter the name for the high-limit attribute, such as **HIGH\_LIMIT**.
2. In the attribute’s **Value** column, enter the default high-limit setting, such as *1000*. Or, define the attribute as a query-based attribute by setting the value as follows:

**Query:**

```
view="Route Stops";
field="High Limit";
where=""EQ Num' = $(EQ Num) and 'Point Num' = $(Point Num)''
```

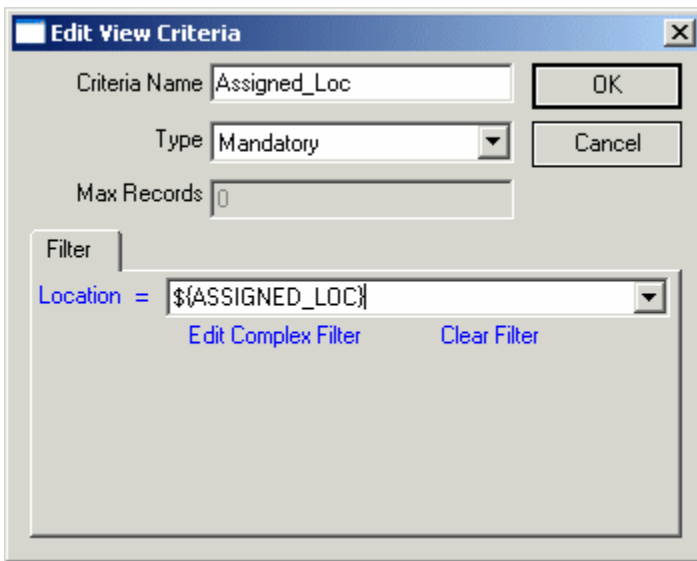
For more information on query-based attributes, see the next section of this guide.

3. Click the **Update** button  or change focus from the pane.

### Query-Based Attributes

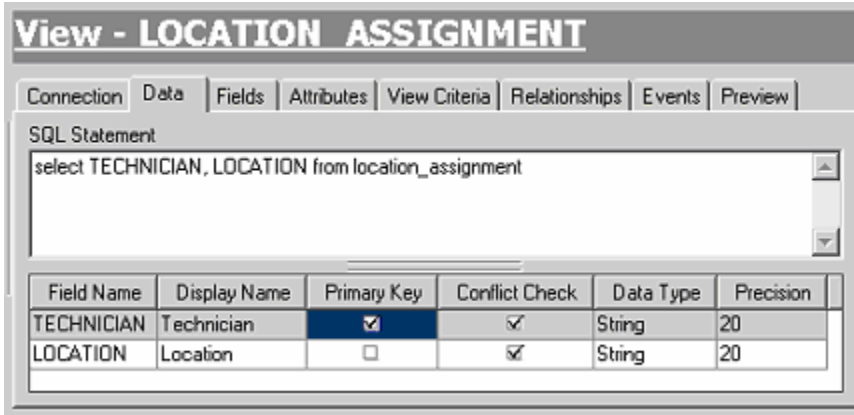
At times, it can be useful to define an attribute whose value is determined *when it is used*, not *when it is defined*. To this end, DataSplice allows you to set up query-based attributes that get their values by performing dynamic lookups against DataSplice views. When online, these lookups are done against the underlying database tables. When offline, they are done against the view data that has been downloaded to the DataSplice Remote Client.

An example will help to clarify this. Suppose maintenance personnel are assigned to specific areas of a plant and their assignments change daily. Further, suppose you would like these personnel to see only the work orders pertaining to their assigned location each day. If each maintenance worker had an **ASSIGNED\_LOC** attribute defined, you could set up a mandatory filter on work order views like this:

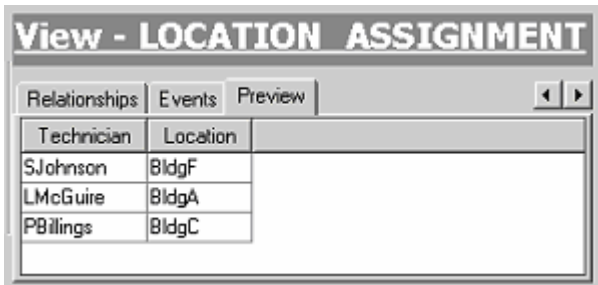


However, this would require you to change the value of **ASSIGNED\_LOC** each day for each maintenance worker. On the other hand, if your CMMS system allowed you to store such assignments in its database, you could use query-based attributes such that DataSplice would look up each worker’s assignment when he or she opened a work order view. The necessary filter would be applied at that point.

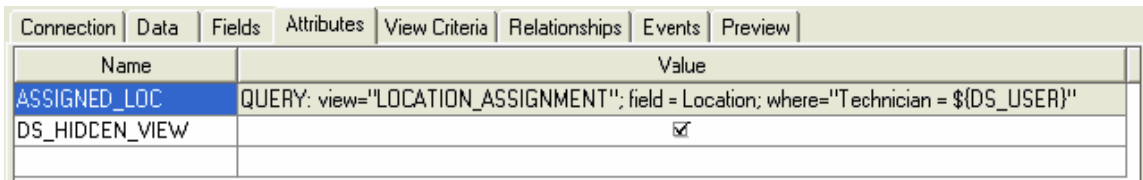
To implement this query-based attribute solution, you would create a view called **LOCATION\_ASSIGNMENT** like this:



The sample data for the view could look like this:

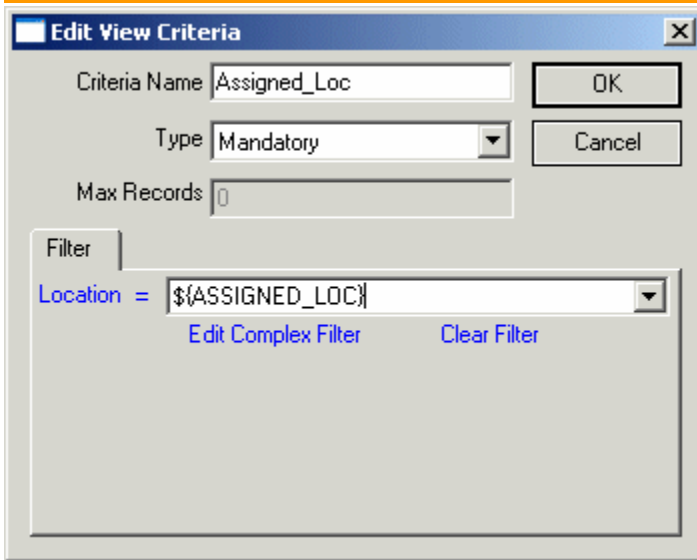


You would then define an attribute called **ASSIGNED\_LOC** as follows:



- *view* is the DataSplice view to query. View names containing white space should be quoted, e.g. *view="View Name"*;
- *field* contains the value we are looking for. The field must match the corresponding field's display name from the DataSplice view. If the display name contains white space, then the value should be quoted, e.g. *field="field display name"*;
- *where* determines which row will match. Again, field references must match their corresponding display names, and any names containing white space need to be surrounded by double-quotes. **Note:** The *where* clause can contain other DataSplice attributes.

Finally, you would create your filter on the **Work Order** view, as described earlier:



With these settings in place, when user *SJohnson* opens the **Work Order** view, the following actions occur:

1. Our query attribute causes a lookup on the **LOCATION\_ASSIGNMENT** view. The row having a **TECHNICIAN** value of *SJohnson* is found, and the corresponding **LOCATION** value (*BldgF*) is assigned to the attribute **ASSIGNED\_LOC**.
2. Our mandatory view filter is applied so that the **Work Order** view only displays items for a location of *BldgF*.

Query attributes are most effective when there is only a single value that can match the lookup criteria. If the query returns more than one value, the first one found will be used. The only way to control which value is found first is to add an *order by* clause to the attribute definition:

**Example:**

**Query:**

```
view="LOCATION_ASSIGNMENT";
field="Location";
where="Technician=${DS_USER}";
orderby="Technician"
```

This would cause the results to sort by **Technician** and the **ASSIGNED\_LOC** attribute to get the value of the first row in the resulting set. In our example, the final value would be *BldgA*. However, the usefulness of this method is limited, since its results depend on the data in the underlying table.

When defining query-based attributes that reference field display names containing white space in the *where* section, the display name must be single quoted. If a display name in the *where* section contains white space and is not quoted, then DataSplice will interpret




that as being two distinct objects, and most likely the query will fail to return the expected result. It may even result in an error message.

**Query:**

```
view="LOCATION_ASSIGNMENT";  
field="Location";  
where="Tech Group'=${my_group_field}";  
orderby="Technician"
```

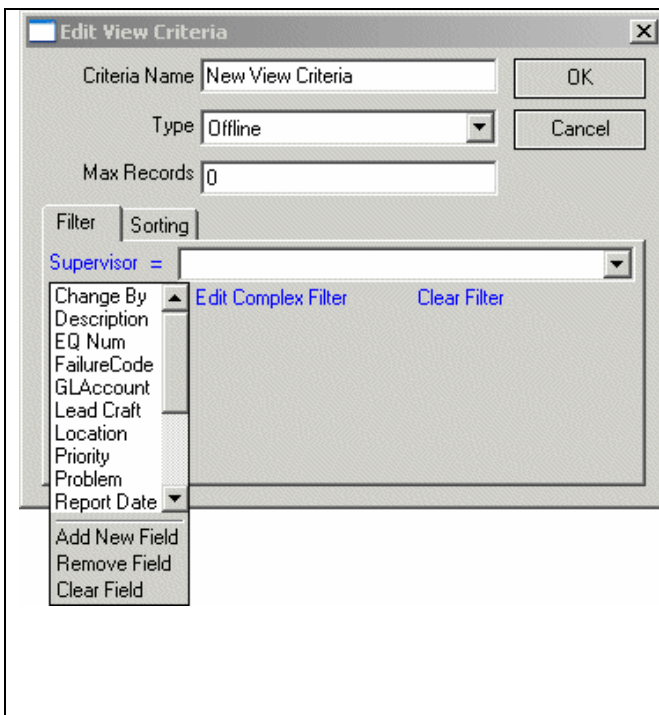
## View Criteria Tab

The **View Criteria** tab allows the administrator to set up one or more sets of criteria that control what data the users see. Also, different types of criteria can be established in conjunction with view and user attributes to allow data selection to vary based on the current client connection.


Criteria Name	Type
 Filter Out WOs by Status	Mandatory
 Offline Criteria	Offline
 Default Criteria	Default

<b>Criteria Name</b>	This is the name of the view criteria.
<b>Type</b>	<p>This is the type of the criteria. It can be either <b>Default</b>, <b>Mandatory</b>, or <b>Offline</b>:</p> <ul style="list-style-type: none"> <li>• <b>Default</b> – This criteria type is used to set up initial filtering of view data. It does not restrict what the users can filter on from the client, and is primarily designed for online sessions.</li> <li>• <b>Mandatory</b> – This criteria type is used to control what data can be seen regardless of session type. It gets applied for both online and offline sessions, and is not changeable by the user.</li> <li>• <b>Offline</b> – This criteria type is used to control what data gets cached offline when clients request an offline session.</li> </ul>

## Adding Criteria Sets to a View



### To add criteria sets to a view:

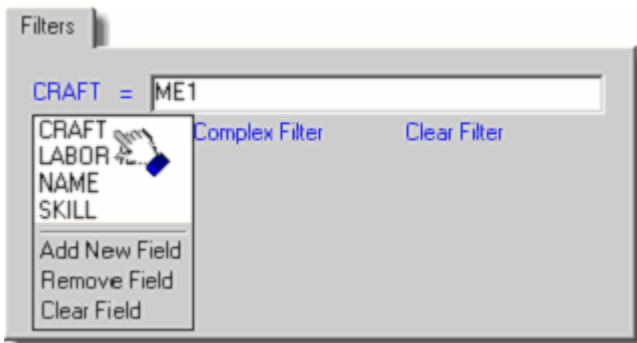
1. Select **Views** in the left pane.
2. Select the desired view from the **Available Views** list.
3. Click the **View Criteria** tab.
4. Click the **Add View Criteria** button .
5. The **Edit View Criteria** dialog will open (as seen at left).
6. Enter the name for the criteria in the **Criteria Name** box.
7. Select the criteria type from the **Type** dropdown box.
8. If the type is **Offline**, specify the maximum number of records to cache in the **Max**

	<p><b>Records</b> box. 0 or less results in an unlimited number of records.</p> <p>9. In the <b>Filter</b> section, specify the filter (if any) to use to limit the data cached offline.</p> <p>10. In the <b>Sorting</b> section, specify the ordering in which the records should be cached.</p> <p>11. Click the <b>OK</b> button.</p>
--	---

### Edit View Criteria Dialog Box Properties

<b>Criteria Name</b>	This is the name of the view criteria. It helps to make the name meaningful, especially when the list of view criteria is long. For example, if the criteria set defines offline work order data for the logged in user, it could be named <i>Offline WO Data by User</i> .
<b>Type</b>	<p>There are three types of criteria – <b>Default</b>, <b>Mandatory</b>, or <b>Offline</b>:</p> <ul style="list-style-type: none"> <li>• <b>Default</b> – This criteria type is used to set up initial filtering of view data. It does not restrict what the users can filter on from the client, and is primarily designed for online sessions.</li> <li>• <b>Mandatory</b> – This criteria type is used to control what data can be seen regardless of session type. It gets applied for both online and offline sessions, and is not changeable by the user.</li> <li>• <b>Offline</b> – This criteria type is used to control what data gets cached offline when clients request an offline session.</li> </ul>
<b>Max Records</b>	This entry is only valid for offline criteria types. Setting <b>Max Records</b> to zero will specify an unlimited number of records. Any number greater than zero will limit the cached offline records on the client to that specified number.
<b>Filter</b>	The <b>Filter</b> tab is used to specify filtering for the view criteria.
<b>Sorting</b>	The <b>Sorting</b> tab is used to specify the ordering of the records for the view criteria.

### Creating Simple Filters



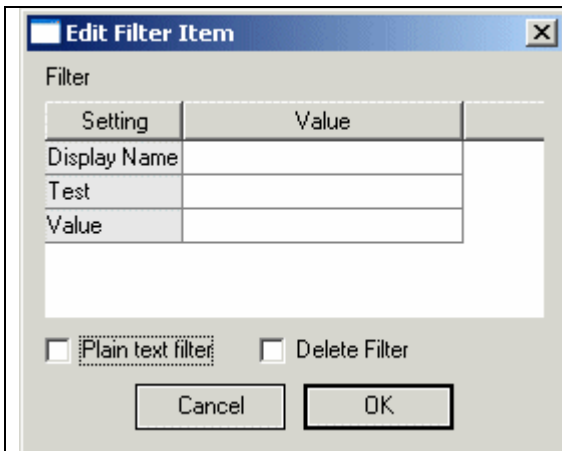
**To create a simple filter:**

1. Double click on the **Criteria Name** field of interest. This will bring up the Edit View Criteria dialog box.
2. Select the field to filter by clicking the blue **Field Name** and selecting a field from the list of searchable fields.
3. Select the filter operator by clicking the “=” next to the **Field Name** and selecting the desired operator.
4. Enter the **Value** to filter on in the box next to the filter operator.
5. Click **OK**.

**To create multiple simple filters:**

1. Double click on the **Criteria Name** field of interest. This will bring up the **Edit View Criteria** dialog box. Select the **Filter** tab.
2. Click the blue **Field Name** to open the **Filter Field** menu.
3. Select **Add New Field** from the **Filter Field** menu.
4. A new filter will appear. Edit this filter as described above.
5. Click **OK**.

**Creating Complex Filters**



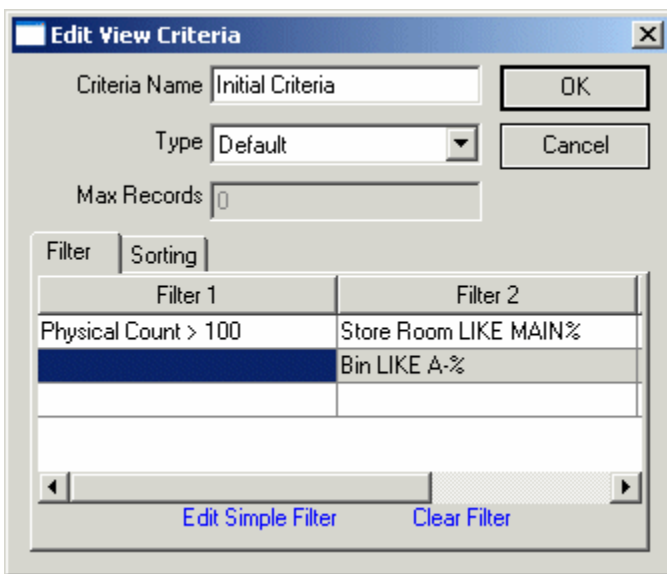
**To create a complex filter:**

1. Double click on the **Criteria Name** field of interest. This will bring up the **Edit View Criteria** dialog box. Select the **Filter** tab.
2. Click the blue **Edit Complex Filter** link.
3. The **Complex Filter** grid will appear.
4. Double click in the grid to add a filter.
5. The **Edit Filter Item** dialog box will appear (as seen here).
6. Edit the **Display Name**, **Test**, and **Value** fields and click **OK**. **Display Name** and **Test** are dropdown lists.

## Combining Filters

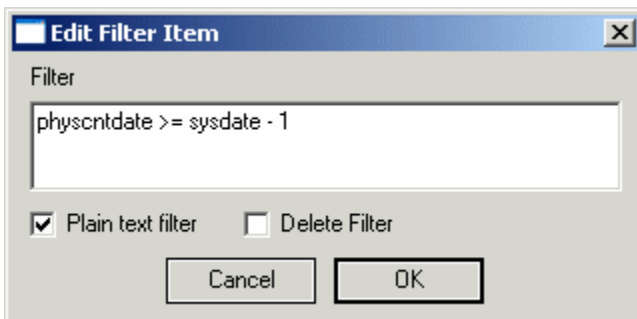
Filters may also be combined to further refine the data displayed. Filters in the *same column* are joined with an “and” operation. The records retrieved with an “and” operation must meet all of the conditions of *all of the filters in that row*. Filters in the *same row* are joined by an “or” operation. The records retrieved with an “or” operation must meet all of the conditions of *at least one of the filters*.

In the following example, the records retrieved would have a **Physical Count** number greater than 100 **OR** they would have both a **Store Room** starting with *MAIN* **AND** a **Bin** starting with A-:



## Advanced Filters

The basic filtering technique (selecting field, test, and value) works for most situations and is preferred because it ensures correct syntax in the query statements. More advanced filters may be created by checking the **Plain Text Filter** box on the **Edit Filter Item** dialog box and entering a custom SQL statement. This provides the opportunity to create unique filters based on the syntax of the particular database view by inputting text that will be inserted into the *where* clause without modification.



## Filter Operators

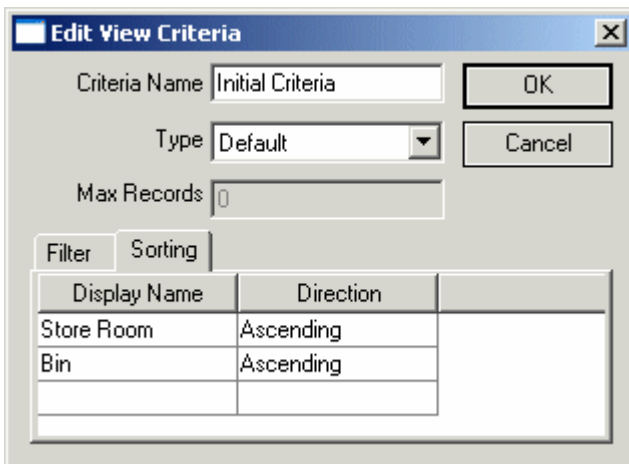
=	Equal To
<	Less Than
>	Greater Than
<=	Less Than or Equal To
>=	Greater Than or Equal To
<>	Not Equal To
<b>LIKE</b>	Compares an item using a wildcard
<b>IS NULL</b>	Returns values which are null
<b>IS NOT NULL</b>	Returns values which are not null

## Wildcards

A wildcard is a placeholder for characters which do not have to be specified. Wildcards can be used in the filter field. For example, the records for all employees whose employee numbers begin with 27 can be retrieved by using a wildcard. In this case, the filter field would be **Employee Number**, and the value in the field would be 27%. The percent sign indicates that anything can follow the 27. In this particular case, all of the following numbers could be returned: 27, 270, 27462983.

## Sorting Tab

The **Sorting** tab is used to specify the ordering of the records for the view criteria. The following image shows a sort order for an offline criteria. The specified sorting orders the records in ascending order based on the value of the **Location** field. Additional fields can be added to the sorting of the view criteria. The first field selected as a sort parameter will establish the primary sorting order for all records. For any subsequent fields, the sorting order will be applied in the order the fields were configured.



<b>Display Name</b>	This dropdown list consists of all the sortable fields in the view.
<b>Direction</b>	This dropdown list indicates whether to sort the data in ascending or descending order.

## Relationships Tab

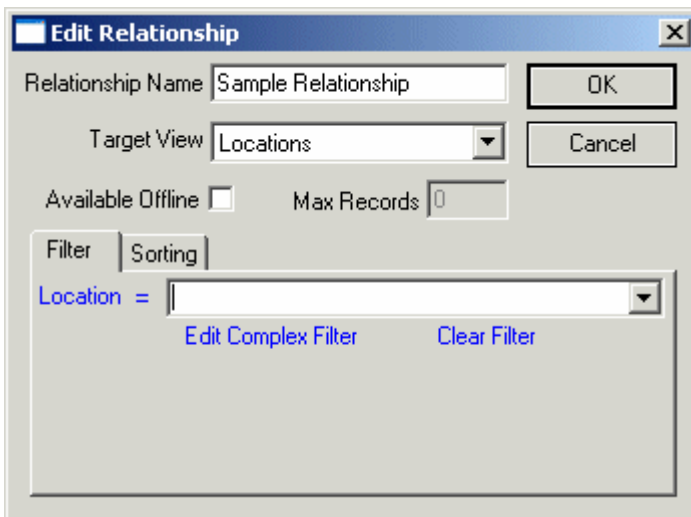
The **Relationships** tab allows the administrator to explicitly describe the relationship between different views. The administrator defines these relationships and gives them a descriptive free-form name. The user then is able to navigate the relationship by a name that clearly explains the relationship.

Relationship Name	Target View
Test Relationship	Search Inventory

A relationship works by building a filter to display the target view with matching records. This filter does not necessarily reference the source record, so relationships can be defined to always return the same set of records regardless of the source record in the parent view.


### Some characteristics of relationships include:

- A parent view is able to have more than one relationship to the same child view using different search criteria.
- Different parent views are able to have relationships to the same child view.
- Relationships can be made available only for a particular user or group.




## Creating a New Relationship

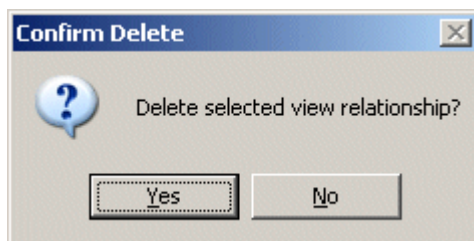
To create a new relationship:

1. Open the **Relationships** tab in the **View** section.
2. Click the **Add Relationship** button  in the toolbar.
3. Enter a name in the **Relationship Name** box. This name will be displayed on the Remote Client device as a navigation option.
4. From the **Target View** list, select the destination view.
5. Specify if the relationship is used when caching offline data by checking the **Available Offline** checkbox.
6. If the relationship is used offline, specify the maximum allowed records in the **Max Records** box. Setting this value to zero specifies an unlimited number of records.
7. Configure the filter section to determine which records are presented in the target view. For more information on filters, please see the **View Criteria** tab.
8. Click **OK**.

## Deleting a Relationship

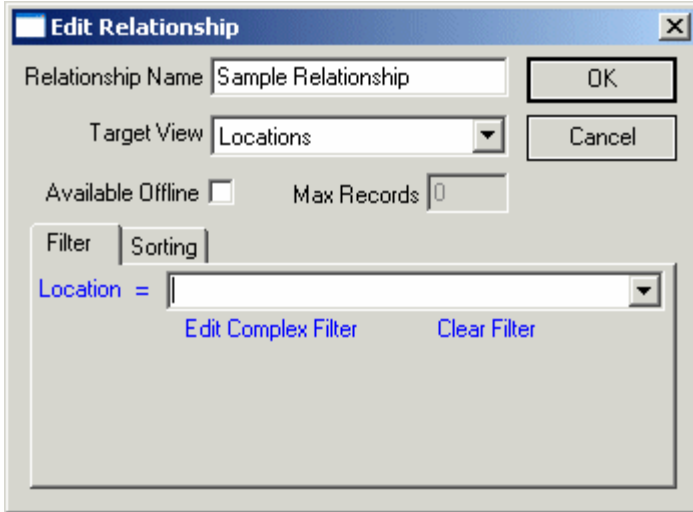
To delete a relationship:

1. Open the **Relationships** tab.
2. Highlight the relationship to be deleted.
3. Click the **Delete Relationship** button .
4. A **Confirm Delete** dialog box will appear. Click **Yes**.



## Editing Relationship Dialog Box Properties

Relationship dialog properties can be edited by accessing the **Edit Relationship** dialog box:



<b>Relationship Name</b>	The <b>Relationship Name</b> can be any name the administrator chooses. However, this name appears in the Remote Client application as a form of navigation. So, the name should reflect the nature of the relationship defined.
<b>Target View</b>	The <b>Target View</b> is the child (destination) view for the relationship. The Remote Client application uses this relationship as a navigation point between a parent and child view that has built-in filtering to predefine the records to display in the child view.
<b>Available Offline</b>	The <b>Available Offline</b> checkbox indicates whether or not to cache offline records for the child view if, and when, records are cached offline for the parent view.
<b>Max Records</b>	The <b>Max Records</b> field indicates the maximum number of child records to cache offline. Specifying zero indicates an unlimited number of records.
<b>Filter</b>	The <b>Filter</b> tab is where the administrator defines the filtering criteria for this relationship. See the <b>View Criteria</b> tab section for more information on configuring filters.

## Events Tab

The **Events** tab allows you to view and create special actions when certain events take place. The tab looks like this:

Connection	Data	Fields	Attributes	View Criteria	Relationships	Events	Preview
Action						Event	
.NET Event Handler Plugin::InventoryModule.SetPhysicalCount						Instead of Update	
.NET Event Handler Plugin::InventoryModule.ReconcilePhysicalCo...						Instead of Update	

### Creating Actions for View Events

When a user accesses and modifies data in a record from a view through the DataSplice Remote Client, an associated event is triggered based on the user's data modification. For instance, if the user changes the value of a quantity field from 5 to 3 in a record displayed in his or her client window, then an update event is requested by DataSplice. If there are no actions associated with the view's update event, then DataSplice will attempt to perform the update directly against the table in the database. However, it is possible to associate customized actions to any of a view's event types.

### View Event Types

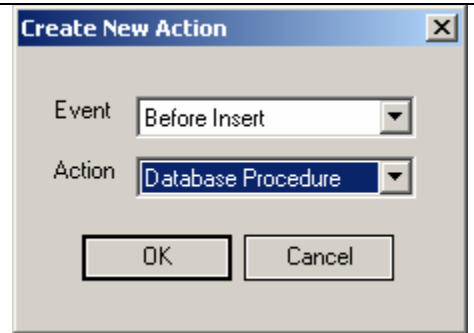

- **Before Insert** – Perform an action before inserting a record into the database table.
- **After Insert** – Perform an action after inserting a record into the database table.
- **Instead of Insert** – Perform an action instead of inserting a record into a database table.
- **Insert Error** – Perform an action if an error occurred when inserting a record into a database table.
- **Before Update** – Perform an action before updating a record in the database table.
- **After Update** – Perform an action after updating a record in the database table.
- **Instead of Update** – Perform an action instead of updating a record in a database table.
- **Update Error** – Perform an action if an error occurred when updating a record in a database table.
- **Before Delete** – Perform an action before deleting a record from the database table.
- **After Delete** – Perform an action after deleting a record from the database table.
- **Instead of Delete** – Perform an action instead of deleting a record from a database table.

- **Delete Error** – Perform an action if an error occurred when deleting a record from a database table.

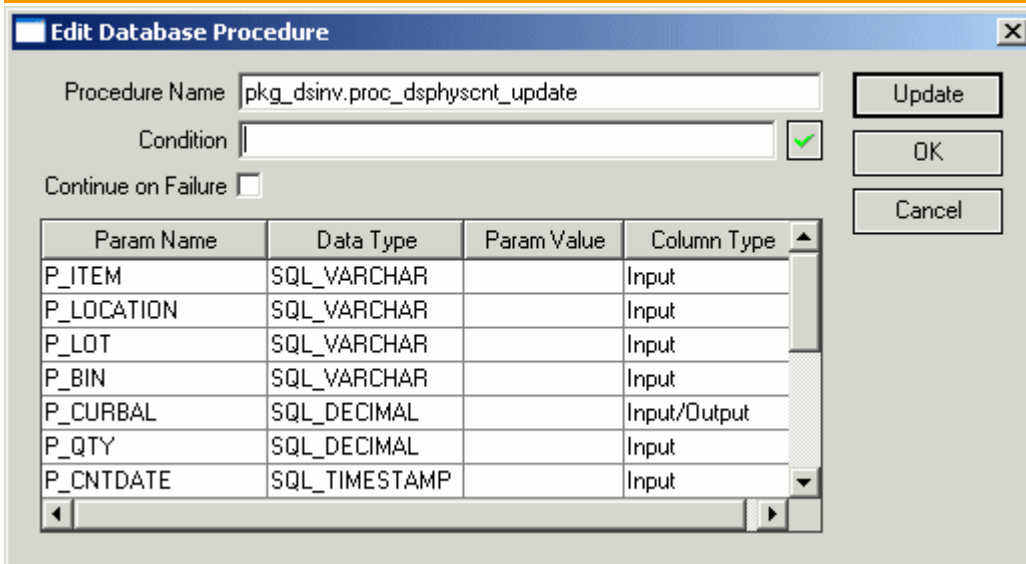
There are four actions that can be associated with any of the above events:

1. **Database Procedure** – Supplies customized data processing by calling a database stored procedure. Database procedures may be performed by DataSplice before, after, or in place of normal operations, such as an insert. These procedures are created with the database, and are then called by DataSplice.
2. **Email Notification** – Supplies email notification capabilities for the associated event type.
3. **Run Application** – Shells to an external application. E.g. based upon an event action (like transferring parts to a new bin), a report writer could automatically print out a new barcode label.
4. **Plug-in Action** – This action type has been set up to support customized data processing based on calling procedures that are external, yet visible, to the DataSplice Server process (i.e. Java routines, C# libraries, etc.).

### Creating a Database Procedure Action

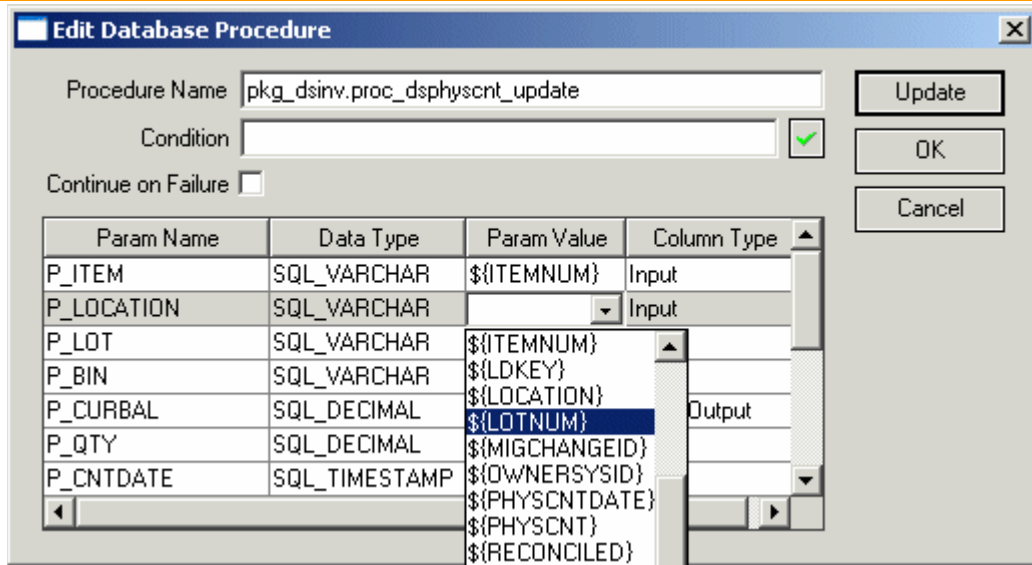
	<p><b>To create a Database Procedure action for an event:</b></p> <ol style="list-style-type: none"> <li>1. Highlight the <b>View Name</b> from the <b>Available Views</b> section.</li> <li>2. Click on the <b>Events</b> tab.</li> <li>3. Click on the <b>Add Action</b> button . This will bring up the <b>Create New Action</b> dialog box (as seen at left).</li> <li>4. In the <b>Event</b> dropdown list, select the event type.</li> <li>5. In the Action dropdown list, select the <b>Database Procedure</b> action type.</li> <li>6. Click the <b>OK</b> button.</li> </ol>
--	--

This will bring up the **Edit Database Procedure** dialog box:



To specify which database stored procedure to call and to bind field and/or attribute values to the procedure call:

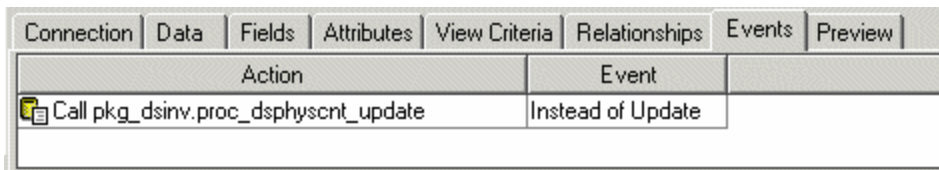
1. Enter the name of the procedure into **Procedure Name** text box. The procedure name may need to be qualified by the schema it resides in. To specify the schema owner, append the schema name followed by a colon to the front of the procedure name. **Example:** *schema:procedure\_name*
2. Click on the **Update** button. If the procedure is valid, the list of parameters for the procedure will be displayed.
3. Enter the condition(s), if there are any, under which this event action should be processed. Leaving the **Condition** field blank means that the action should always be processed.
4. Turn the **Continue on Failure** checkbox on or off, depending on whether or not DataSplice should continue processing event actions if an error is encountered during the processing of this particular action.
5. In the **Param Value** column, bind the desired field or attribute value to the parameter to be passed into the database procedure when it is called. Double clicking in the **Param Value** cell will bring up a dropdown list of available fields. Also, you can select the cell and type in the field's display name or attribute name that you want to bind.



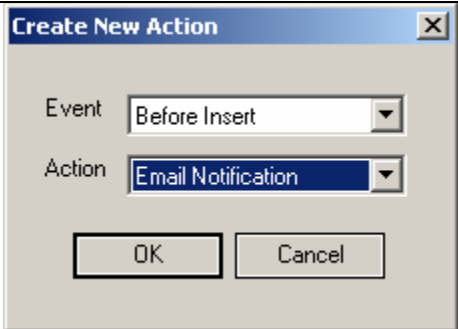

Note: The *`\${field display name}`* syntax is required when binding fields and attributes to a parameter. You can bind text literals and/or numeric constants by entering the desired value without the *`\${value}`* notation.

6. Click the **OK** button.

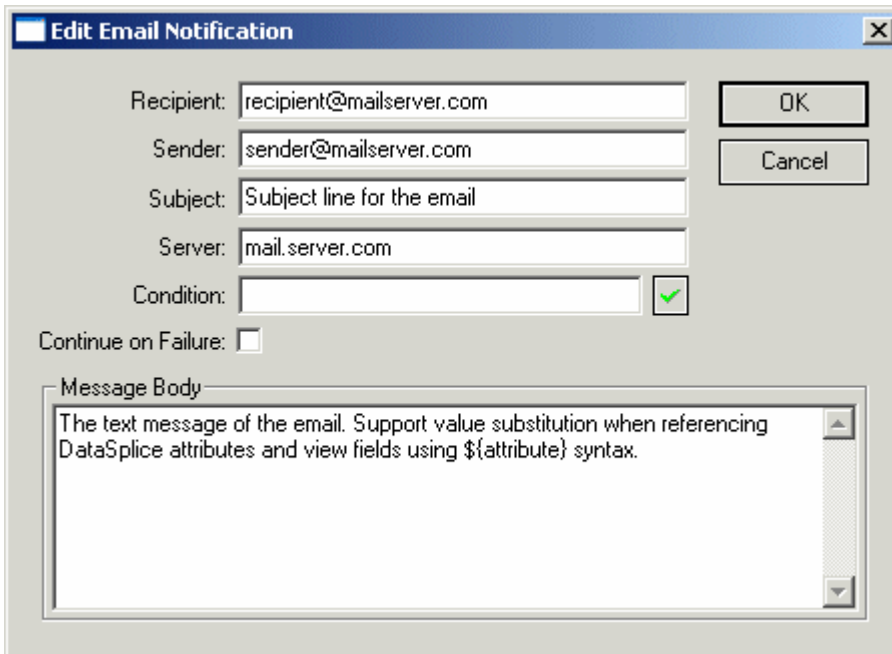
The event action will appear in the view's **Event** list:



## Creating an Email Notification Action

	<p><b>To create an Email Notification action for an event:</b></p> <ol style="list-style-type: none"> <li>1. Highlight the <b>View Name</b> from the <b>Available Views</b> section.</li> <li>2. Click on the <b>Events</b> tab.</li> <li>3. Click on the <b>Add Action</b> button . This will bring up the <b>Create New Action</b> dialog box (as seen at left).</li> <li>4. In the <b>Event</b> dropdown list, select the event type.</li> <li>5. In the <b>Action</b> dropdown list, select the <b>Email Notification Action</b> type.</li> <li>6. Click the <b>OK</b> button.</li> </ol>
---	--

This will bring up the **Edit Email Notification** dialog box:



In the dialog box, enter the following information:

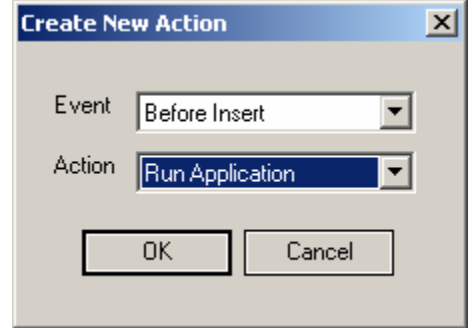

- **Recipient** – The recipient’s email address. In the example above, *recipient@mailserver.com* could be replaced with a reference to an *\${EMAIL\_ADDRESS}* attribute that could be used to determine the email address.
- **Sender** – The sender’s email address. In the example above, *sender@mailserver.com* could be replaced with a reference to an *\${EMAIL\_ADDRESS}* attribute that could be used to determine the email address.
- **Subject** – This is the subject heading for the email.
- **Server** – This is the mail server to use for sending emails. This must be an SMTP server.
- **Condition** – A DataSplice *Where* clause to specify under what conditions the action should be fired.
- **Continue on Failure** – Check box specifying whether or not DataSplice should continue processing event actions if an error occurs during the email attempt.
- **Message Body** – The body text of the email message.

All email settings will be able to contain attribute references that will be replaced when the email is sent. This will allow for useful things such as creating a *MAIL\_SERVER* attribute that is referenced for the SMTP server. E.g. enter *\${MAIL\_SERVER}* in the **Server** text box. Also, if the view’s record contains an email address, then you can use the field to specify the recipient and/or sender’s email address. Example: *\${Field\_Display\_Name}* or *\${Attribute\_Name}*.

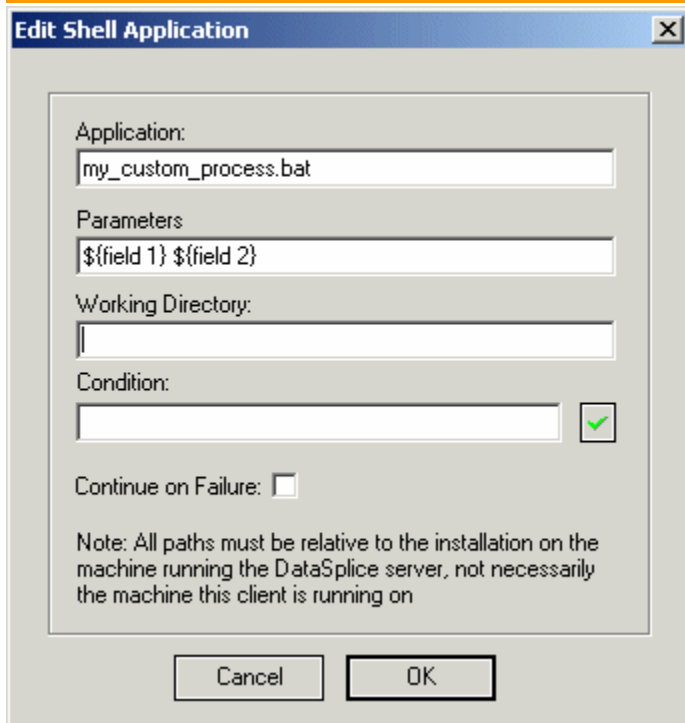
In addition, the body of the email can contain formatting codes (and field references) that will get replaced as follows:

- **%u** – Insert information about the current client connection, including the user name, domain, and IP address.
- **%a** – Insert a table of the current attribute settings.
- **%r** – Insert the settings of the current record (if available).
- **%e** – Insert information about the current error (if any).
- **%%** - Replace with a single parenthesis character.

### Creating a Run Application Action

	<p><b>To create a Run Application action for an event:</b></p> <ol style="list-style-type: none"> <li>1. Highlight the <b>View Name</b> from the <b>Available Views</b> section.</li> <li>2. Click on the <b>Events</b> tab.</li> <li>3. Click on the <b>Add Action</b> button . This will bring up the <b>Create New Action</b> dialog box (as seen at left).</li> <li>4. In the <b>Event</b> dropdown list, select the event type.</li> <li>5. In the <b>Action</b> dropdown list, select the <b>Run Application</b> action type.</li> <li>6. Click the <b>OK</b> button.</li> </ol>
---	---

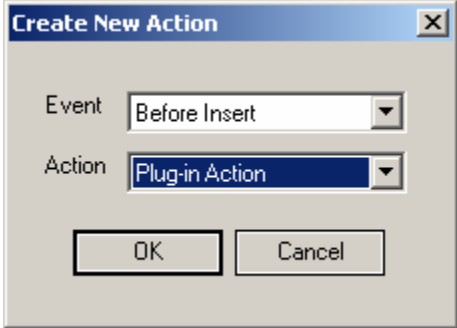
This will bring up the **Edit Shell Application** dialog box:




In the **Edit Shell Application** dialog box, enter the following information:

- **Application** – The application/external process to run. In the example above, *my\_custom\_process.bat* could be replaced with a reference to an *\${APP\_SHELL}* attribute that could be used to determine, in this case, the particular *.bat* to execute.
- **Parameters** – The parameter values list that should be passed to the external process. In the example above, *\${field 1} \${field 2}* represents two of the possible fields from a view.
- **Working Directory** – The directory that the external process should start working from.
- **Condition** – A DataSplice *where* clause to specify under what conditions the action should be fired.
- **Continue on Failure** – Check box specifying whether or not DataSplice should continue processing event actions if an error occurs.

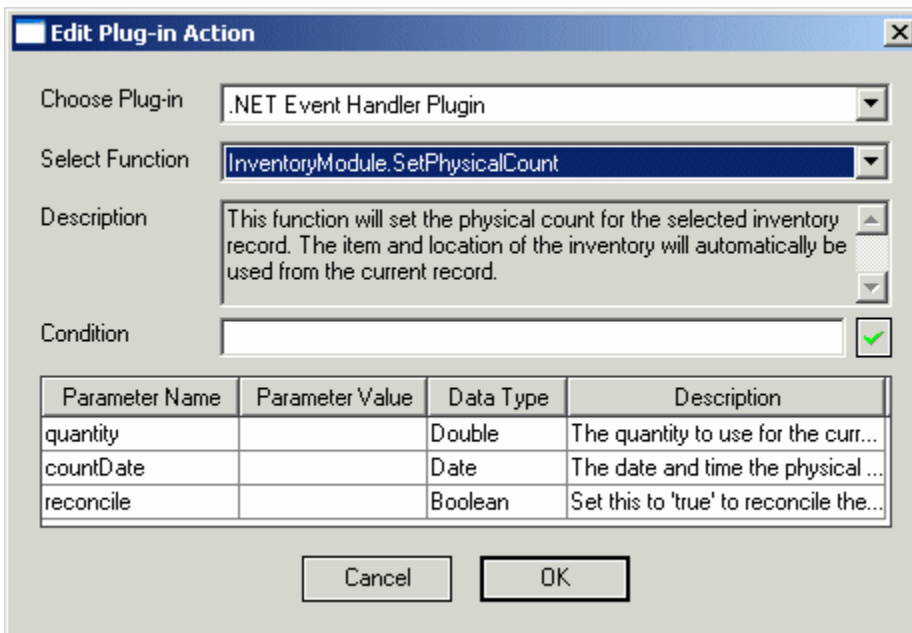
## Creating a Plug-in Action



**To create a Plug-in Action for an event:**

1. Highlight the **View Name** from the **Available Views** section.
2. Click on the **Events** tab.
3. Click on the **Add Action** button . This brings up the **Create New Action** dialog box (as seen at left).
4. In the **Event** dropdown list, select the event type.
5. In the **Action** dropdown list, select the **Plug-in Action** type.
6. Click the **OK** button.

This will bring up the **Edit Plug-In Action** dialog box:



Parameter Name	Parameter Value	Data Type	Description
quantity		Double	The quantity to use for the curr...
countDate		Date	The date and time the physical ...
reconcile		Boolean	Set this to 'true' to reconcile the...

To specify which plug-in function/method to call and to bind field and/or attribute values to the plug-in method call, perform the following steps:

1. Select the target plug-in from the **Choose Plug-in** dropdown selection box.
2. Select the function/method you wish to execute from the **Select Function** dropdown selection box. A description of the function selected will be displayed in the Description box under the function dropdown.
3. Enter the condition(s), if any, under which this event action should be processed. Leaving the **Condition** field blank means that the action should always be processed.


4. Turn the **Continue on Failure** checkbox on or off, depending on whether or not DataSplice should continue processing event actions if an error is encountered during the processing of this particular action.
5. In the **Param Value** column, bind the desired field or attribute value to the parameter to be passed into the plug-in function when it is called. Double clicking in the **Param Value** cell will bring up a dropdown list of available fields. Also, you can select the cell and type in the field's display name or attribute name that you want to bind. **Note:** the *`\${field display name}`* syntax is required when binding fields and attributes to a parameter. You can bind text literals and/or numeric constants by entering the desired value without the *`\${value}`* notation.
6. Click the **OK** button.

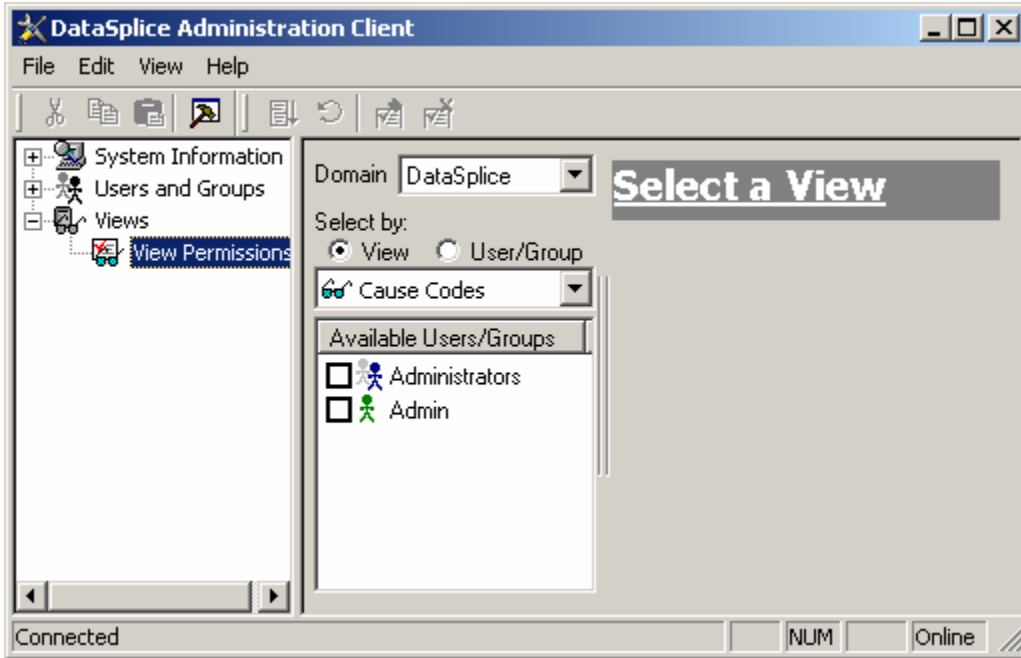
The event action will appear in the view's event list:

Connection	Data	Fields	Attributes	View Criteria	Relationships	Events	Preview
Action						Event	
.NET Event Handler Plugin::InventoryModule.SetPhysicalCount						Instead of Update	
.NET Event Handler Plugin::InventoryModule.ReconcilePhysicalCo...						Instead of Update	




## View Permissions

Expanding the **Views** section reveals the **View Permissions**  listing. Clicking this button brings up the **View Permissions** screen. This screen allows you to see and alter access to particular users, groups, and views. The screen looks like this:




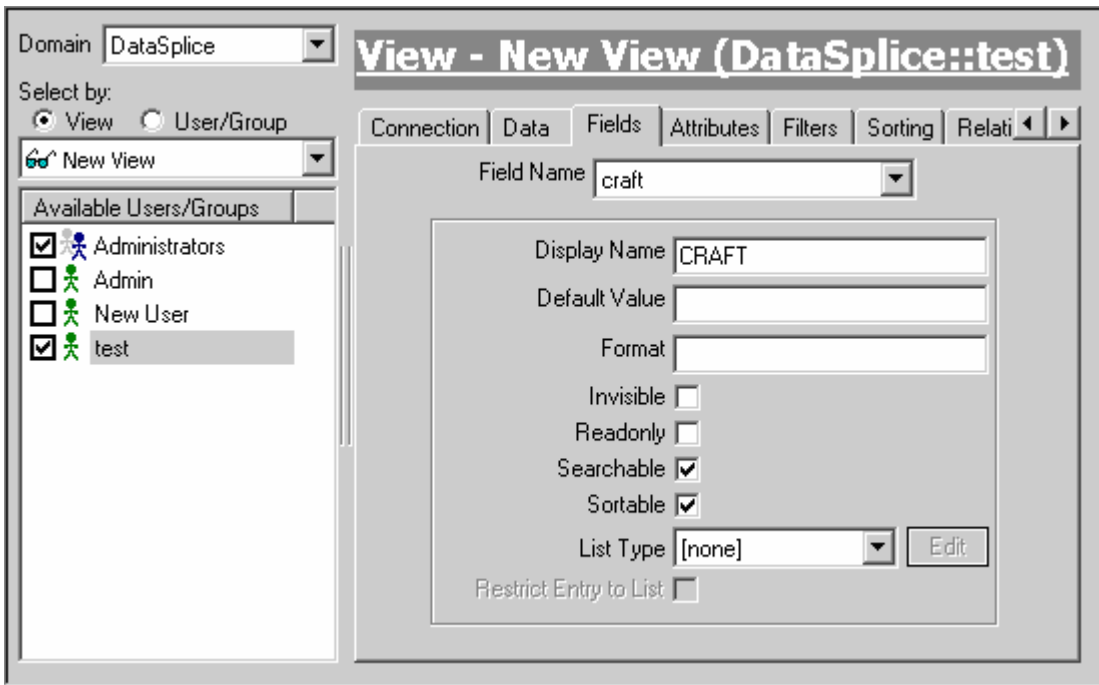
Group membership generally determines access to, and permissions for, views. Permissions may also be set for individuals, with access rights that differ from the rest of the group.

Default permissions are set in the **Views** section under the **Connection** tab. These permissions (and therefore access to the view) can be extended to the user, or they may be modified on a per-user or per-group basis. Once you click on a user or group, the **Add Access** button  will become active. You can use this button, or simply place a check in the checkbox next to the user's name, to provide extended access to that user or group.


**Note:** When view permissions are altered for a specific user, those permissions take precedence over the standard view permissions. For example, if you grant *Update* permissions to a specific user and then grant *Insert* permissions to the standard view permissions for that view, when that user logs in he/she will have *Update* permissions but not *Insert* permissions. Any changes to the standard permissions for that view will not be reflected in the user's permissions. To reset a user's permissions for a particular view so they match this new standard view permission, you must remove the permissions for that user and then reassign them. You can do this by unchecking the box next to their name, then rechecking it. The user's permissions will now be reset to whatever is now the standard view permissions for that view.

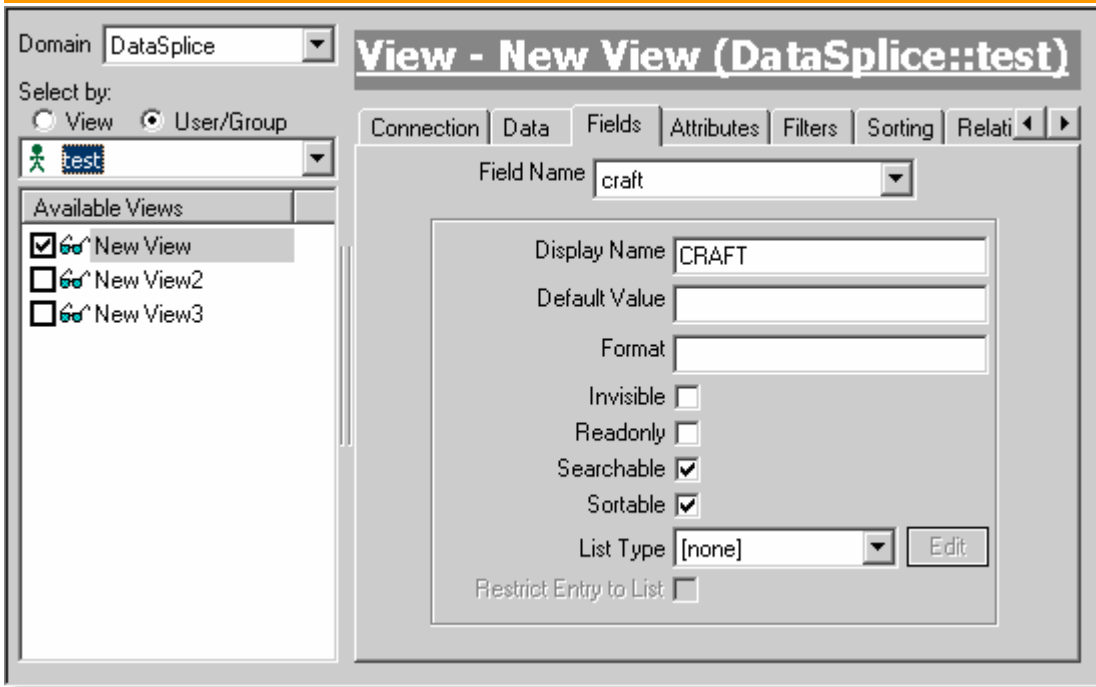
**To grant permission by view:**

1. Expand the **Views** section in the left pane.
2. Select the **View Permissions** section.
3. Choose the **Select by: View** radio button.
4. Select the view of interest from the dropdown list.
5. Check or uncheck each **User Name** or **Group Name** to indicate whether they should have permission or not.
6. Click the **Update** button  in the toolbar.



**To grant permissions by user or group:**

1. Expand the **Views** section in the left pane.
2. Select the **View Permissions** section.
3. Choose the **Select by: User/Group** radio button.
4. Select the **User or Group** of interest from the dropdown list.
5. Check or uncheck each view to indicate whether permission is allowed.
6. Click the **Update** button  in the toolbar.



## Overriding Defaults

Permissions can be overridden by highlighting a view or user with permissions to that view, then changing the configuration of the view as necessary. The only items that may not be changed are the SQL statement and the type of list. Items that appear in the list may be overridden.

The following are some examples of permission overrides. This is not an exhaustive list, but rather some illustrative examples.

### Overriding Access Permissions

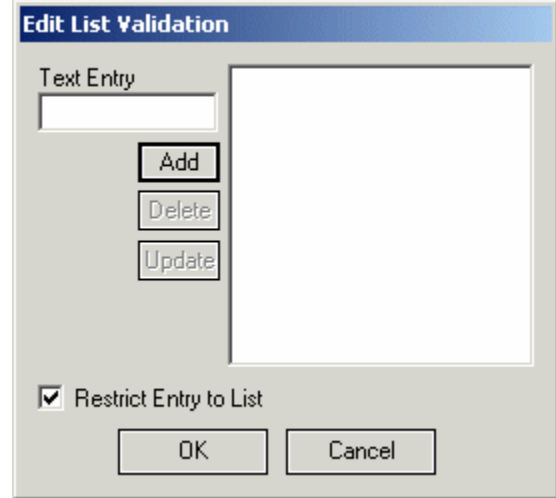
A supervisor may need to change a description of a work order, and this access may not be desired for a worker. In this case, the default permissions may be set to make the description field a **Readonly** in the **Field** tab. Then, the administrator can open the view in the **View Permissions** section and open the view for the **Supervisor** group and deselect the **Readonly** checkbox in the **Fields** tab, then set the default permissions on the **Connection** tab to allow updates.

### Overriding Visible Fields

Some fields may not be necessary for a particular group of workers. Someone stocking the inventory need not see the price of each item, as this may clutter the view with too many fields. A salesman using the same view will need the price field. The field can be enabled for the **Salesmen** group, and marked as invisible for the **Stock** group.

### Overriding List Items

Although list types are set by the initial configuration of the view, the items in the list may be changed to suit a user or group. A group based in one warehouse may not need to use the inventory in another warehouse, therefore their list of item descriptions can be limited to just their warehouse.

	<p><b>To override a list type value list for a field:</b></p> <ol style="list-style-type: none"> <li>1. Select the list entry in the <b>Field Validation</b> dropdown box.</li> <li>2. Click the <b>Edit</b> button.</li> <li>3. The <b>Edit List Validation</b> dialog box will open (as seen at left).</li> <li>4. In the <b>Edit List Validation</b> dialog, add entries to the list by typing the values into the <b>Text Entry</b> box in the upper left corner and clicking the <b>Add</b> button.</li> <li>5. Click the <b>OK</b> button when finished adding entries to the list.</li> <li>6. The list entries will appear to the user in the same order in which they are added.</li> <li>7. Select the <b>Restrict Entry to List</b> checkbox to prevent the user from typing data into the field and forcing them to select a value from the list provided.</li> </ol>
---	---

The static list can display additional information to the right of the list of values. To create a two column list, separate the value from the additional information with the pipe character:

**Example:** “1 | 1st Value”

### Overriding Filters

The data which appears as part of the view may be changed per user or group by adding or altering the filters which appear in the **Filters** tab. This may be used to limit the inventory that appears to only that which is in a particular storeroom, or may limit work orders to those assigned to a particular supervisor.

### **Overriding Display Name**

Display names may be altered for a specific user or group by editing the **Display Name** field in the **Fields** tab. The staff that places orders may refer to an item by a UPS number, while the staff that pulls the inventory may refer to an Item number. This is the same number in both cases, but it has a different name for each application. The **Display Name** can be changed in this case to avoid confusion.

## Reference

In addition to the material previously mentioned in this manual, there are some other reference topics that are not specific to working with the DataSplice Administration Client, but are helpful for users at all skill levels. These topics will be discussed in the following sections:

- **ODBC**
- **SQL**
- **Format Strings**

## ODBC

Open Database Connectivity (ODBC) is a widely accepted application programming interface (API) for database access. Databases use proprietary data access schemes that require the use of different database languages. ODBC eliminates the need to create code specific to each database, instead providing one standard set of calls. The calls are translated to the appropriate language by the ODBC drivers. Utilizing ODBC, databases are referred to by their Data Source Name (DSN).

**Note:** If you have established your ODBC data source but it does not show up in DataSplice, please be sure to disconnect and reconnect to the DataSplice Administration Client. Your data source should then be listed.

### Acquiring ODBC

ODBC is required for DataSplice and is standard on many installations of Windows. To verify that ODBC is installed, open the **Control Panel** (Windows 95, 98, NT, and ME) or **Administrative Tools** in the **Control Panel** (Windows 2000). If **ODBC** appears in either location, it is already installed. ODBC is available to download for Windows from Microsoft. <http://www.microsoft.com/data/download.htm>.

### Data Sources

**The three types of data sources:**

#### User DSN

An ODBC **User Data Source Name** stores information about how to connect to the indicated data provider. A User data source is local to a computer and is accessible only by the current user.

#### System DSN

An ODBC **System Data Source Name** stores information about how to connect to the indicated data provider. A System data source is local to a computer but is not user-dedicated. Any user can access a system DSN. In most cases, this is the preferred type of data source for use with DataSplice.

#### File DSN

An ODBC **File Data Source Name** allows you to connect to a data provider. A File data source need not be user-dedicated or local to a computer.

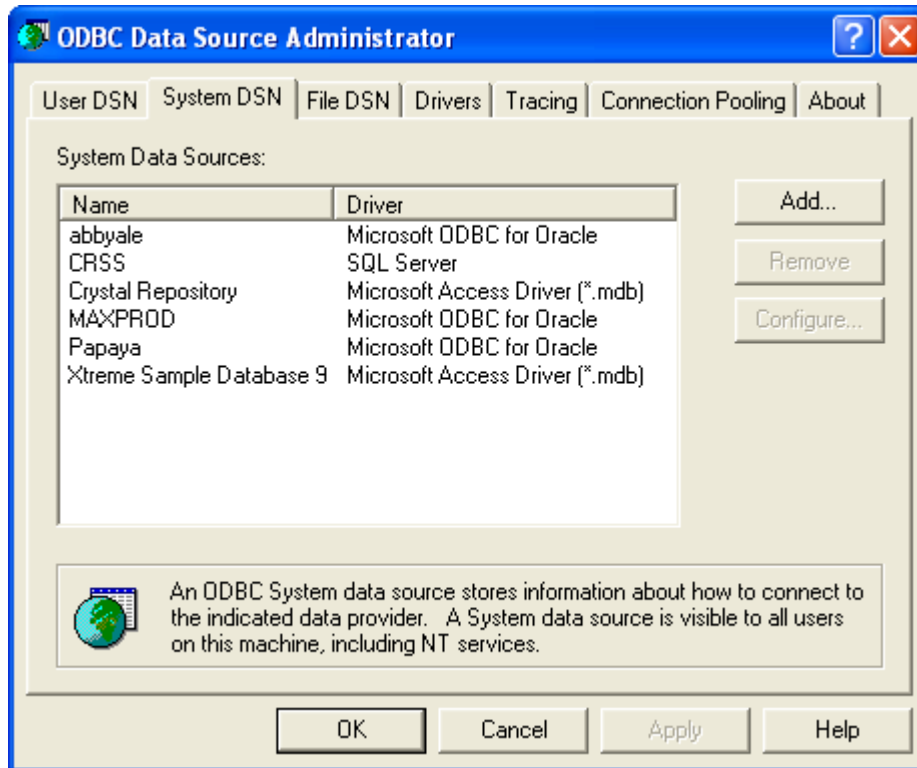
**Note:** DataSplice requires that each DataSplice Server have at least one DSN set for all databases utilized by DataSplice.

### Creating a Data Source in Microsoft ODBC Data Source Administrator

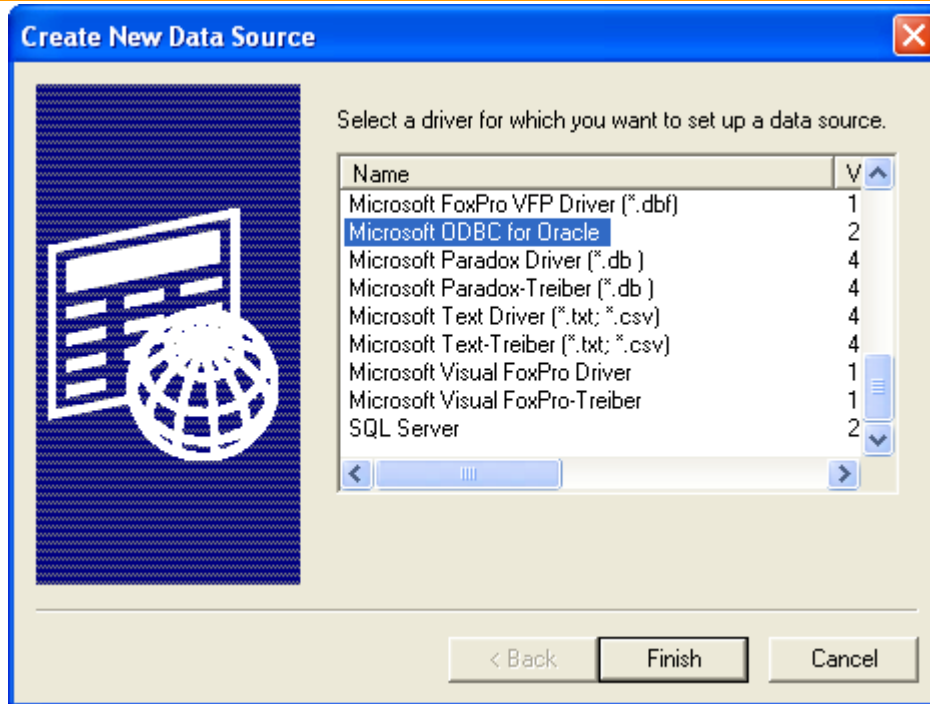
Before creating an ODBC data source, be sure the database server is configured. For more information regarding configuration of the database, consult the documentation that was provided with the particular database.

**To create an ODBC data source:**

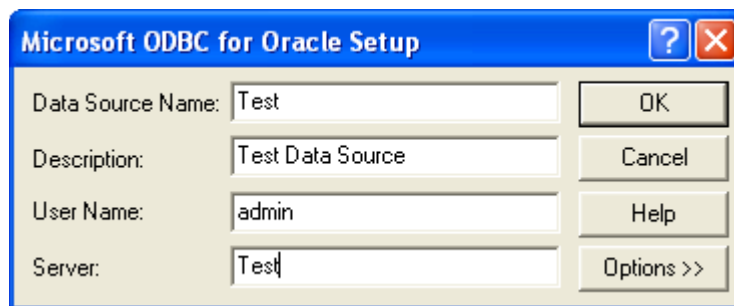
1. Launch the ODBC Data Source Administrator by selecting **Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC)**.
2. Select the appropriate DSN tab. In most cases, you will want to use a **System DSN**:



3. Click **Add** to begin creating the data source. This will open the **Create New Data Source** dialog box:



4. Select the appropriate driver for the data source, and then click **Finish**. A dialog box will appear to allow you to configure the data source. Each data source driver has a different installation interface. Consult the documentation provided with the data source for more information about how to complete the fields.



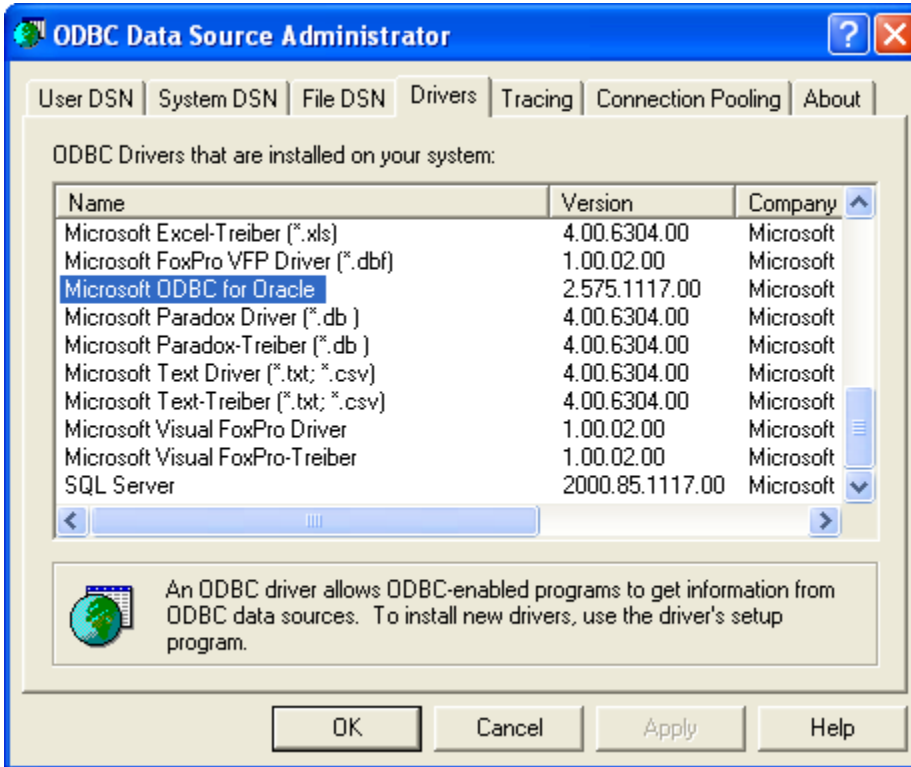
5. After you have entered the appropriate information for your data source, click **OK** to finish. You can make modifications to the data source later by highlighting the data source from the list on the **System DSN** tab and then clicking **Configure**.

## Drivers

Databases use proprietary data access schemes that require the use of different database languages. ODBC provides an interface that standardizes the syntax necessary to communicate with ODBC-compliant databases. The ODBC drivers are responsible for translating the syntax into the database-specific language, which allows one set of code to access many different databases. Each type of database utilized must have its own ODBC driver installed.

## Installing ODBC Drivers

In order to add or remove an ODBC driver, the setup utility that is specific to the particular driver and database must be run. The **Drivers** tab in the ODBC Data Source Administrator only displays information about the drivers already installed. *Drivers may not be installed from this location.*

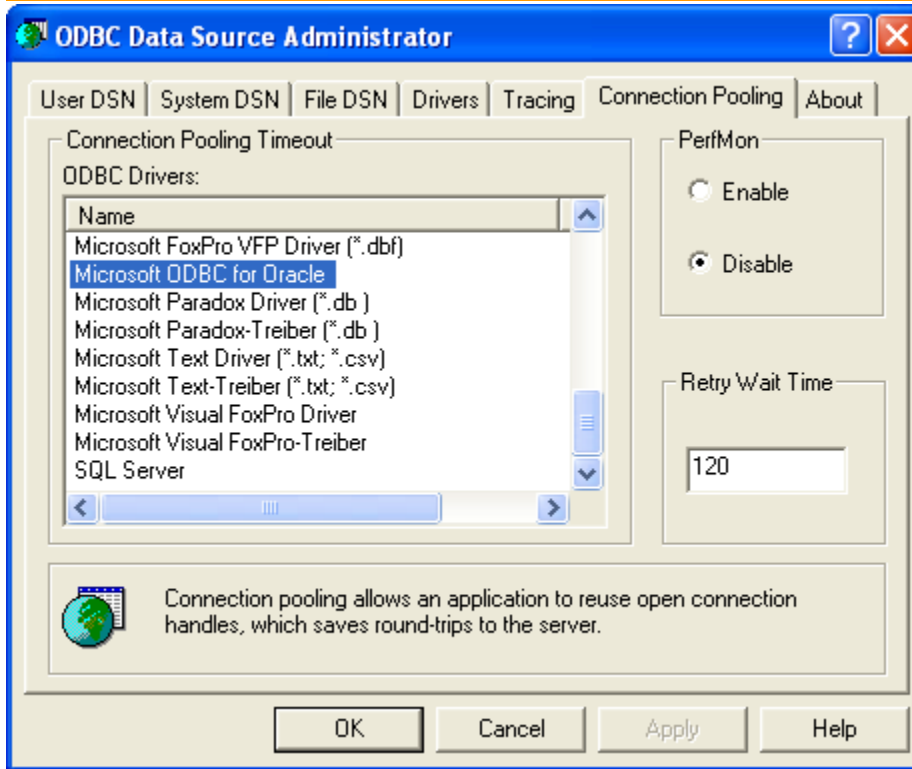


## Connection Pooling

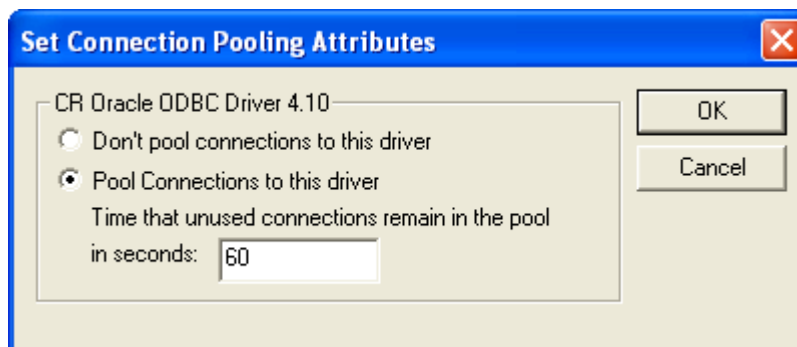
When an application needs a particular driver, it first requests a connection handle with which to identify the driver and then requests that the Driver Manager load the driver. Connection pooling places these handles in a pool so that they may be reused without having to re-establish a connection, thus saving round-trips to the server and improving performance.

### Connection Pooling Tab

The **Connection Pooling** tab sets the connection pooling timeout, retry wait time, and enables or disables performance monitoring.



- **Connection Pooling Timeout** – This specifies, in seconds, the time that unused connections will remain in the pool. To set the connection pooling timeout, double click the appropriate driver. The **Set Connection Pooling Attributes** dialog box appears:



- **PerfMon** – Enable or disable performance monitoring by selecting the appropriate radio button for this option.
- **Retry Wait Time** – This specifies, in seconds, the interval at which the ODBC Driver Manager retries to make a connection to the database server.

# SQL

## Accessing Information through DataSplice

The DataSplice Administration Client application uses SQL queries to determine which records, and which fields in those records, will be included in a particular view. First, it uses SQL in the **Data** tab of the **Views** section. This determines which *fields* will be included in the view. Second, it uses SQL in the **Filters** tab of the **Views** section. This determines which *records* will be included in the view. The DataSplice Remote Client application uses SQL for filters as well. However, the Remote Client filter is applied only to the view of that particular user, not all users with access to that view.

SQL stands for "Structured Query Language" and is often pronounced "sequel" instead of "es-que-el". SQL is a standardized, non-procedural language used to communicate with a SQL-compliant database in order to read, write, and delete information from the database.

## Basic SQL Queries

### SELECT

A **SELECT** query is used to extract information from the database. In its simplest form, it states what data is to be extracted and from which table the data should be extracted.

#### Example:

Information regarding a customer named Nelson is needed. Assuming the database table (called *CUSTOMERS*) looks like this:

CUST_ID	FIRST_NAME	LAST_NAME	LAST_PURCHASE
1	Tom	Servo	December 5, 1996
4	Mike	Nelson	January 3, 1999
22	Joel	Robinson	October 1, 1998

The query would look like this:

```
SELECT * FROM CUSTOMERS
WHERE LAST_NAME = "Nelson";
```

And would return these results:

CUST_ID	FIRST_NAME	LAST_NAME	LAST_PURCHASE
4	Mike	Nelson	January 3, 1999

Queries can be broken down into several sections. The following query is looking for all customers with a last name beginning with "Nel". The numbers are not part of the actual query. Rather, they are present to illustrate what each part of the query performs:

1. SELECT FIRST\_NAME
2. FROM CUSTOMERS
3. WHERE LAST\_NAME
4. LIKE "Nel\*"

<b>1</b>	<b>SELECT</b>	Tells the system what type of query will be run. The field after it ( <i>FIRST_NAME</i> ) tells it what will be viewed.
<b>2</b>	<b>FROM</b>	Chooses the database table to be examined.
<b>3</b>	<b>WHERE</b>	Sets the criteria, or parameters, for the data within the field.
<b>4</b>	<b>LIKE</b>	Sets the conditions of the query, whether it is an exact or a close match. This could also be any of a number of logical operators.

## Logical Operators

Logical operators allow you to choose how you would like to compare information in your query.

=	Equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
<>	Not equal to
<b>LIKE</b>	Compares an item using a wildcard
<b>IS NULL</b>	Returns values which are null
<b>IS NOT NULL</b>	Returns values which are not null

**Note:** There are no spaces between the characters which compose “Not Equal”, “Less Than or Equal To”, and “Greater Than or Equal To”.

### Example:

Payroll needs to know who accrued overtime in the 12<sup>th</sup> week of the year. The query would look like this:

```
SELECT NAME FROM PAYROLL
WHERE HOURS_WORKED > 40 AND WEEK = 12;
```

This will return all of the employees who have worked more than 40 hours in week 12.

Conversely, payroll may need to know who did not accrue overtime. In that case, the query would look like this:

```
SELECT NAME FROM PAYROLL  
WHERE HOURS_WORKED <= 40 AND WEEK = 12;
```

This will provide the desired names.

## Complex SQL Queries

### Wildcards

Wildcards allow queries to be performed using partial information.

#### Example:

Data on a client is needed, but no one can remember the name of the client. Someone remembers the last name begins with “Ser”. The query would look like this:

```
SELECT * FROM CUSTOMERS  
WHERE LAST_NAME LIKE “Ser%”;
```

This will return all the data on any customers with a last name beginning with “Ser”, including “Servo”, the fellow being sought.

Notice the use of the comparison `LIKE`. Additionally, the wildcard used in this case, as in most cases, is the percent sign. Some databases use other wildcards, such as an asterisk. Check the documentation for the database to be used for information regarding which wildcard to use.

### Distinct

The keyword *DISTINCT*, when used in an SQL query, will return only one instance of each item searched for, regardless of how many times it appears in the database.

#### Example:

A company would like to know in how many states they have clients. The query would look like this:

```
SELECT DISTINCT STATE FROM CUSTOMERS
```

The query could return the following results, even if the company has 10,000 customers in each state, because `DISTINCT` has removed the duplicates:

<b>STATE</b>
<b>TX</b>
<b>CO</b>

<b>MN</b>
<b>IL</b>
<b>OH</b>

### **More Information**

SQL queries are capable of accessing information in multiple tables and performing operations on that information such as averaging. For more in-depth information regarding SQL queries, please go to <http://www.sql.org>.

## Format Strings

### Formatting Date and Time

To format a date or time, use the following table to determine the correct variable(s) to use. They may be used with other text, including punctuation.

The date format

*09/03/01*

would be entered into the format field as

*%m/%d/%y*

The date format

*Monday – September 03, 2001*

would be entered as

*%A - %B %d, %Y*

### Formatting Decimals

Decimals can be formatted using the *%.#f* where # is the number of decimal places appearing after the decimal point.

To display zero decimal places:

*%.0f*

To display one decimal place:

*%.1f*

And so on for any number of decimal places.

**Note for Oracle users:** ODBC reports Oracle integers as doubles. As such, Oracle integers should be formatted using the above notation.

## Formatting Options

<b>%a</b>	Abbreviated weekday name
<b>%A</b>	Full weekday name
<b>%b</b>	Abbreviated month name
<b>%B</b>	Full month name
<b>%c</b>	Data and time representation appropriate for locale
<b>%d</b>	Day of month as decimal number (01 – 31)
<b>%H</b>	Hour in 24-hour format (00 – 23)
<b>%I</b>	Hour in 12-hour format (01 – 12)
<b>%j</b>	Day of year as decimal number (001 – 366)
<b>%m</b>	Month as decimal number (01 – 12)
<b>%M</b>	Minute as decimal number (00 – 59)
<b>%p</b>	A.M./P.M. indicator for 12-hour clock
<b>%S</b>	Second as decimal number (00 – 59)
<b>%w</b>	Weekday as decimal number (0 – 6 Sunday is 0)
<b>%x</b>	Date representation for current locale
<b>%X</b>	Time representation for current locale
<b>%y</b>	Year without century, as decimal number (00 – 99)
<b>%Y</b>	Year with century, as decimal number
<b>%%</b>	Percent sign