

# Runtime Database Query Filters

## 1. Introduction

This document describes making of a sample application QueryFilter.ezp.

In many situations you may want to query a database based on user selections. This is to build query filters based on user selections.

Performers DataTable and DataQuery have following methods for building filters at runtime:

RemoveFilters – This method removes all filters so that WHERE clause of the SQL SELECT statement will become empty. Usually you use this method to prepare for building new filters.

AddFilterAsAND – This method adds a new filter. It uses logic AND to add the new filter to existing filters.

AddFilterAsOR – This method adds a new filter. It uses logic OR to add the new filter to existing filters.

Performer QueryFilter helps you to build complex filters. You may use AddFilter and RemoveFilter methods in response to user clicks on selection buttons. Its Filters property is the filters constructed and can be used by DataTable and DataQuery via AddFilterAsAND or AddFilterAsOR methods.

The sample application QueryFilter.ezp shows you how to use the above features.

It is assumed that you already know how to use databases in Limnor applications. If you need more information, please download following tutorial <http://www.limnor.com/downloads/Lesson9.doc>

## 2. Database Used

A simple database table is used to illustrate the concept. It just contains one table with 3 records:

Customers								
CustomerID	CompanyName	ContactName	Address	City	State	Country	ZipCode	email
1	West Side Trading Corporation	Paul Adams	123 Main Street	Vancouver	BC	Canada	V5E 3W2	info@westSite.com
2	Sunrise computer services	Mike Lee	321 Broadway	Seattle	WA	USA	76903	sales@sunrise.com
3	Pacific Shipping Corporation	Tom Bush	789 35th Street	Vancouver	WA	USA	87933	info@ps.com

A DataTable performer is used to show the data:

Customers								
CompanyName	ContactName	Address	City	State	Country	ZipCode	email	
Sunrise computer services	Mike Lee	321 Broadway	Seattle	WA	USA	76903	sales@sunrise.com	
West Side Trading Corpor	Paul Adams	123 Main Stre	Vancouver	BC	Canada	V5E 3W2	info@westSite.com	
Pacific Shipping Corporati	Tom Bush	789 35th Stree	Vancouver	WA	USA	87933	info@ps.com	

### 3. User Selection Options

We give following options to users:

1. Select cities. All customers for the cities the user selected will be listed. For example, if Seattle is selected then the first customer will be displayed; if Seattle and Vancouver are both selected then all 3 customers will be displayed.
2. Select countries. All customers for the countries the user selected will be listed. The logic is the same as for selecting cities.
3. If both cities and countries are selected then customers displayed by the DataTable should satisfy both country and city selections, not just country selection and not just city selection. For example, if Vancouver and USA are selected then only the 3<sup>rd</sup> customer will be displayed. The 2<sup>nd</sup> customer will not be displayed because its country is not USA.

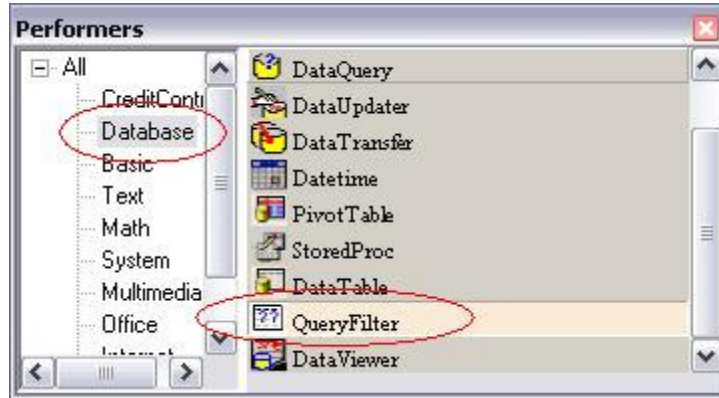
The above logics are our business requirements for this sample application.

We use CheckBox performers to build user interface to let the user to do country and city selections.

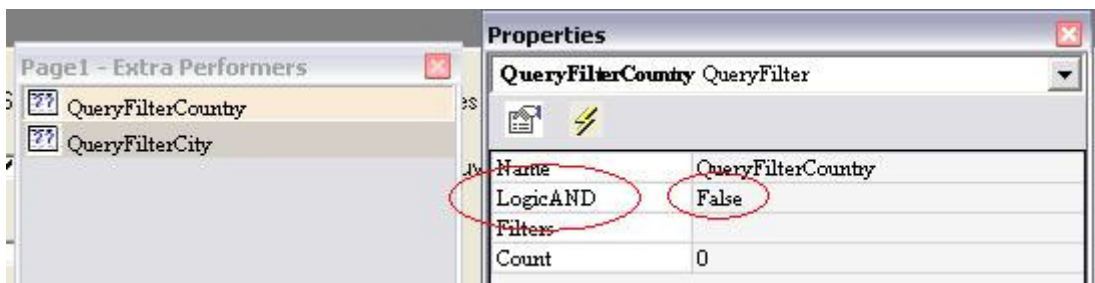
Select Countries	Select Cities
<input checked="" type="checkbox"/> USA	<input type="checkbox"/> Vancouver
<input type="checkbox"/> Canada	<input checked="" type="checkbox"/> Seattle

### 4. Build Filters

We use two QueryFilter performers to build filters; one for country selections and one for city selections.

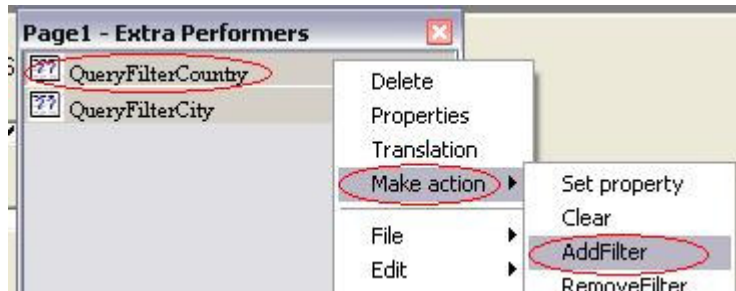


According to our business requirements, we need to use OR logic for all selected countries. So we need to set LogicAND property of QueryFilterCountry performer to False:



With the same reason we need to set LogicAND property of QueryFilterCity performer to False.

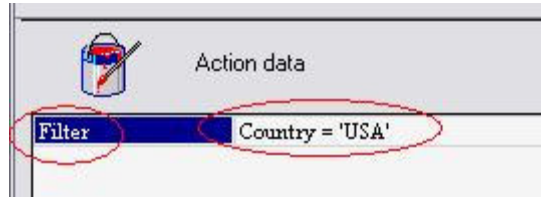
When the user selects USA, we need to add a filter for Country='USA'. Let's make such an action. Right-click on QueryFilterCountry, choose "Make action"; choose "AddFilter":



Give an action name, say, QueryFilterCountry.AddFilterUSA:

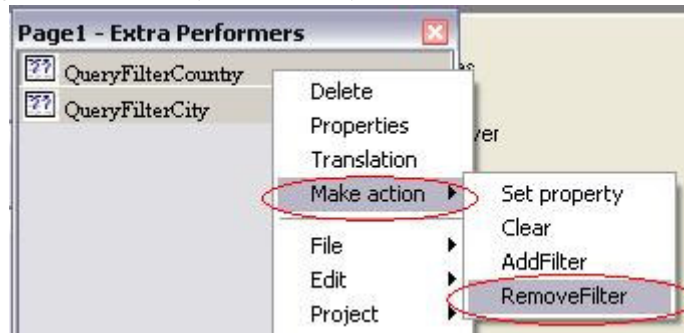


For Filter parameter, type in Country='USA'



Now we can finish making this action.

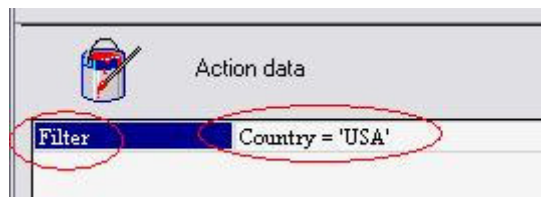
If the user unselect USA, we need to make an action to remove filter `Country='USA'`. Let's make an action to do that. Right-click on `QueryFilterCountry`, choose "Make action"; choose "RemoveFilter":



Give an action name, say, `QueryFilterCountry.RemoveFilterUSA`:



For Filter parameter, type in `Country='USA'`



**Warning:** the filter typed in must be exactly the same as you typed for action `QueryFilterCountry.AddFilterUSA`, including spaces.

For this reason, it is a good idea to avoid using spaces in your filters.

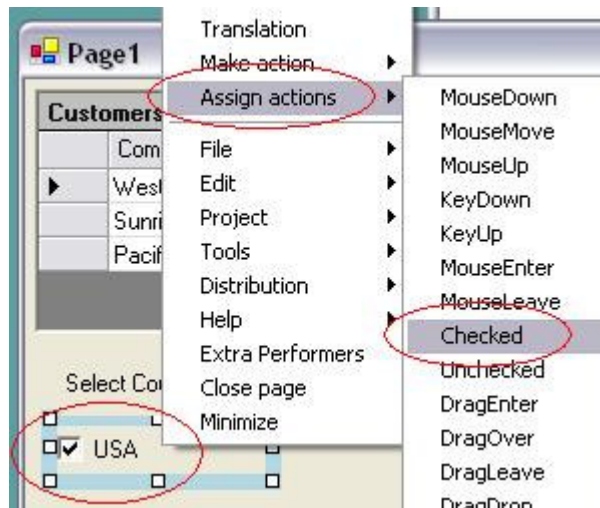
Now we have an action to add a filter when the user selects USA and an action to remove the filter when the user unselects USA.

We can make all actions needed for all city and country selections.

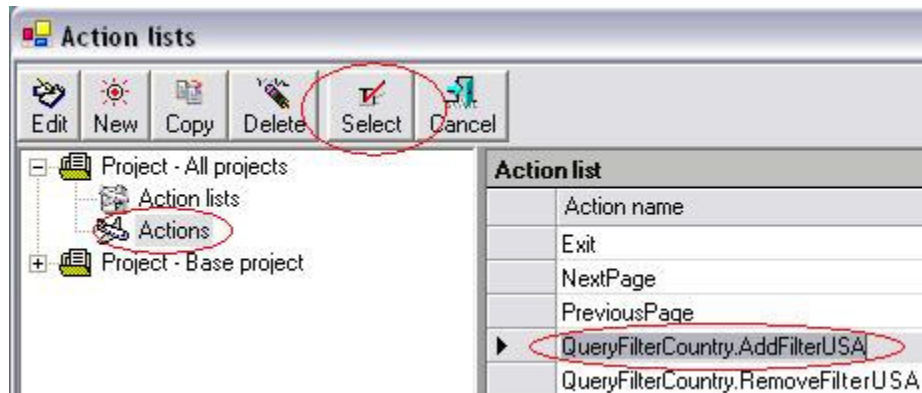
## 5. Build Filters Based On User Actions

In section 3 we build user interface components. In section 4 we build actions for building filters. In this section, we link the filter building actions to user actions. User actions are events fired by user interface components. In our case they are the Checked/Unchecked events of the CheckBox performers.

Right-click on the CheckBox for the USA selection; choose “Assign actions”; choose “Checked”:

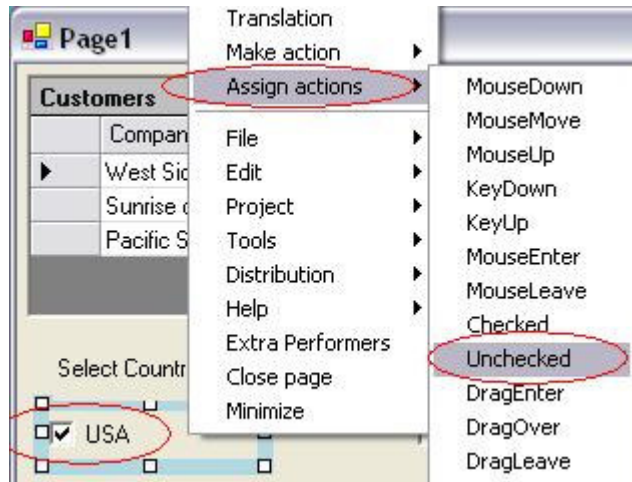


Select Actions; select QueryFilterCountry.AddFilterUSA, click “Select” button:

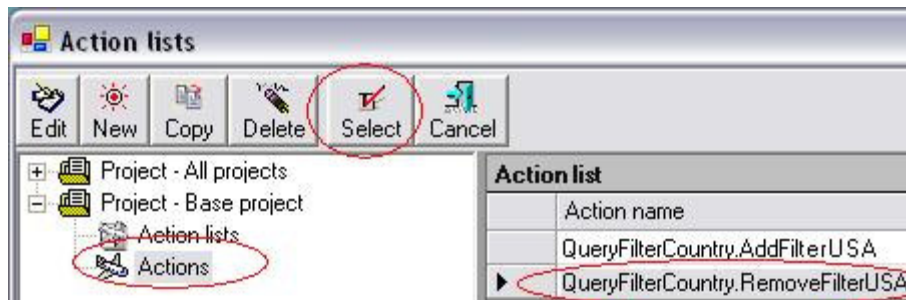


Now when the user selects country USA the filter for USA will be added.

Right-click on the CheckBox for the USA selection; choose “Assign actions”; choose “Unchecked”:



Select Actions; select QueryFilterCountry.RemoveFilterUSA, click “Select” button:



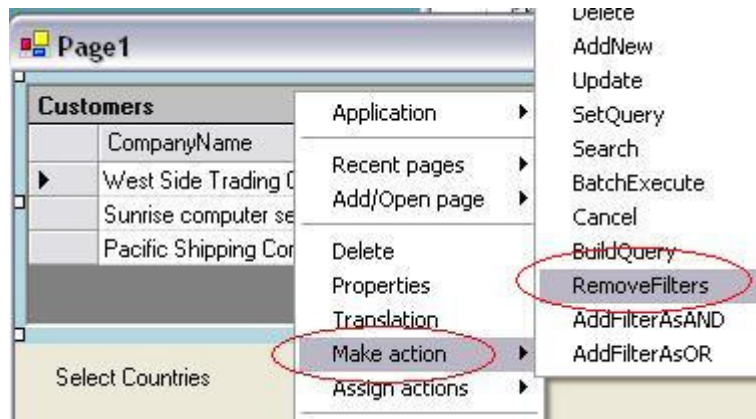
Now when the user unselects country USA the filter for USA will be removed.

We can do the above arrangements for all countries and cities.

## 6. Apply Filters

The filters for country selections are indicated by Filters property of performer QueryFilterCountry. The filters for city selections are indicated by Filters property of performer QueryFilterCity. We need to apply these filters to the DataTable performer in this sample application.

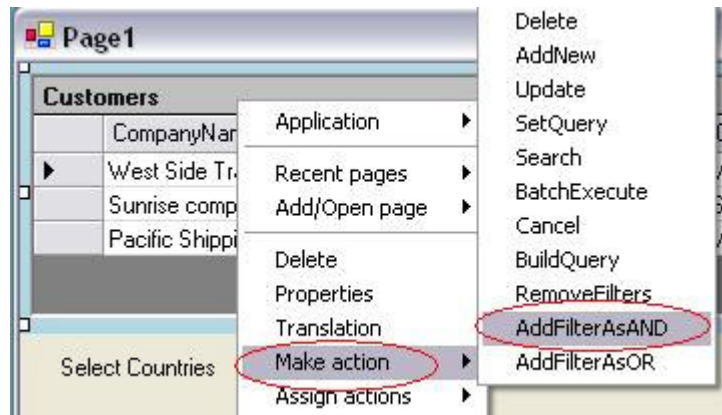
Right-click on the DataTable performer; choose “Make action”; choose “RemoveFilters”:



Give an action name, say, DataTable1.RemoveFilters.

This action is for preparing new filters for the DataTable by removing all existing filters.

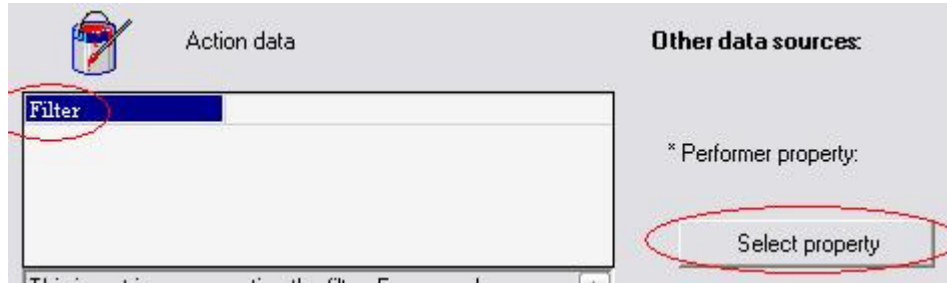
To create an action to apply country selection filters right-click on the DataTable performer; choose “Make action”; choose “AddFilterAsAND”:



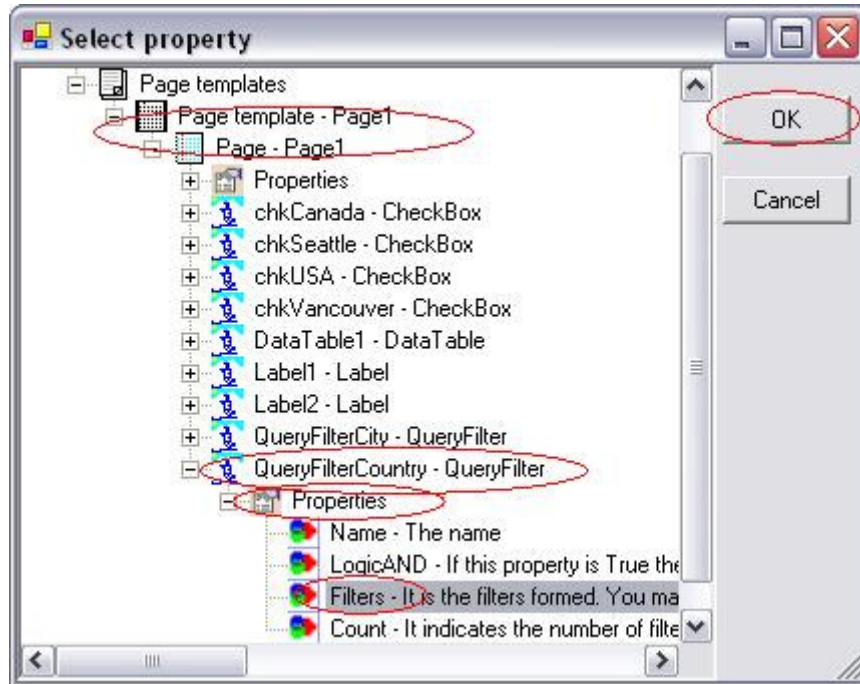
Give an action name, say, DataTable1.AddCountryFilters:



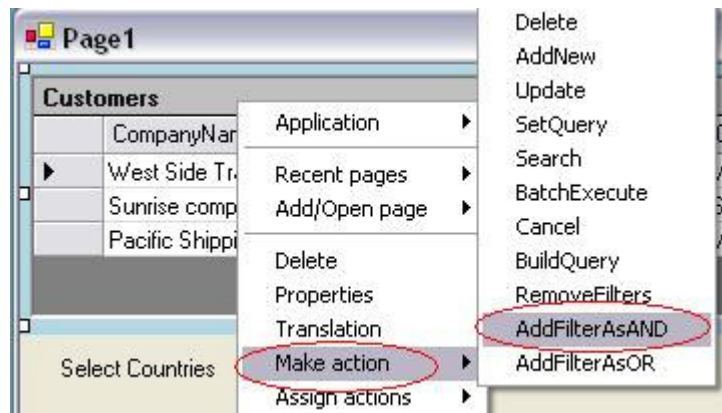
Click “Select property” button to select Filters property of the performer QueryFilterCountry:



Find the QueryFilterCountry performer, select its Filters property, click OK button:



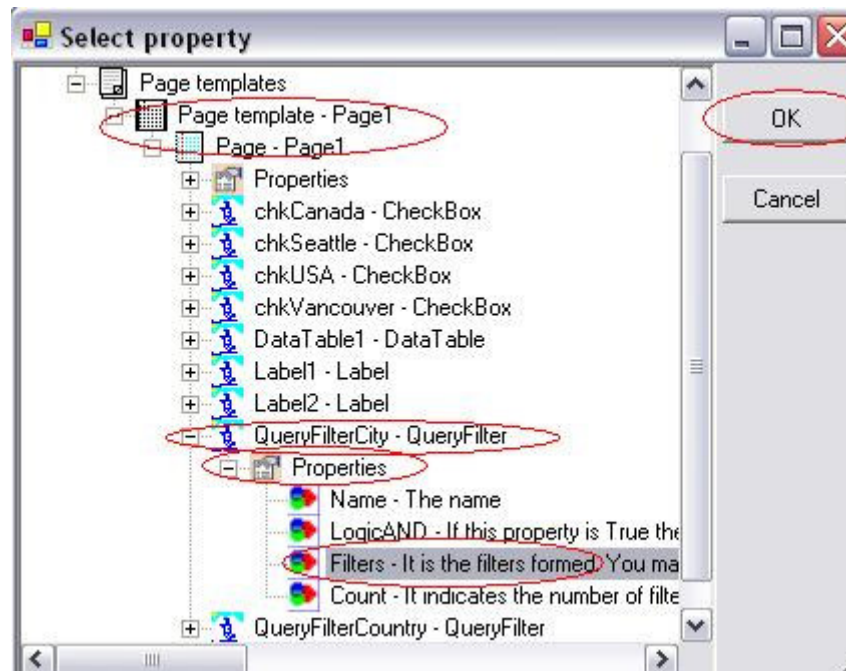
To create an action to apply city selection filters right-click on the DataTable performer; choose "Make action"; choose "AddFilterAsAND":



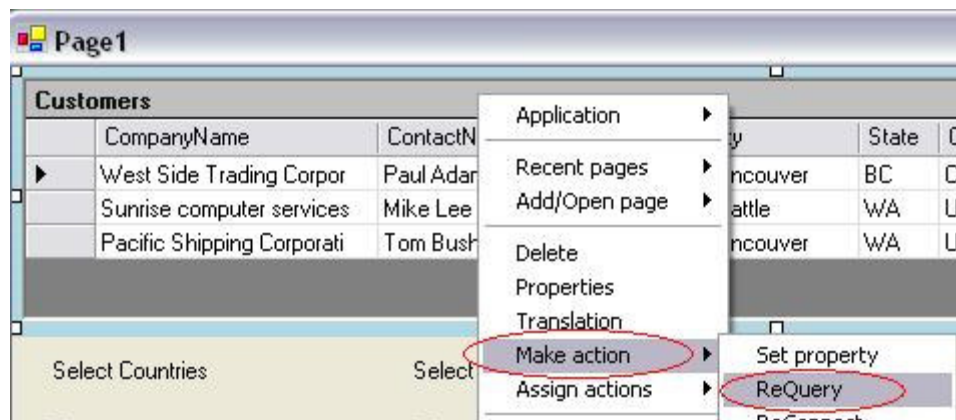
**Note:** here we use method AddFilterAsAND, not method AddFilterAsOR. It is because our business requirement. See section 3, item 3. We need to use logic AND between city selections and country selections.

Give an action name, say, DataTable1.AddCityFilters.

Click “Select property” button to select Filters property of the performer QueryFilterCity:

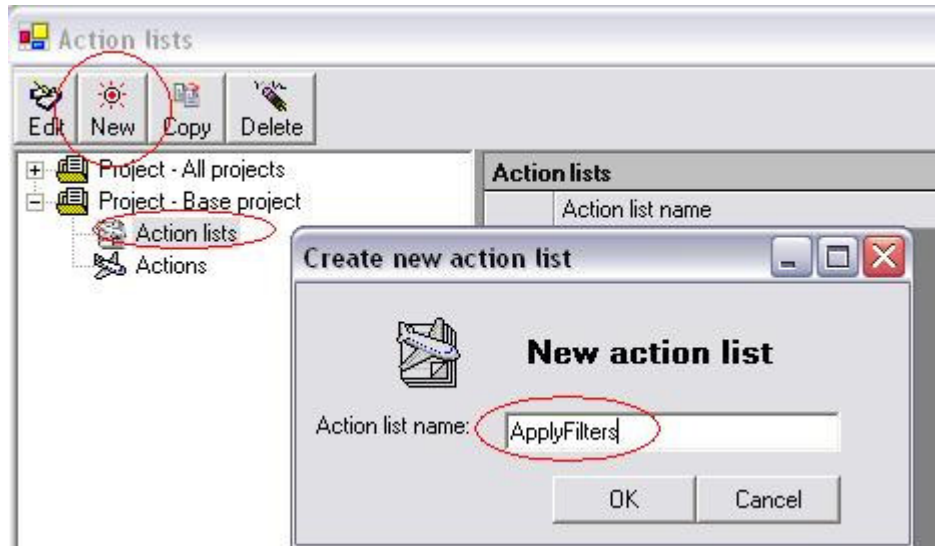


After applying filters, we need to connect to the database to fetch the data. This can be done by creating an action using ReQuery method. Right-click on the Data Table; choose “Make action”; choose “ReQuery”:

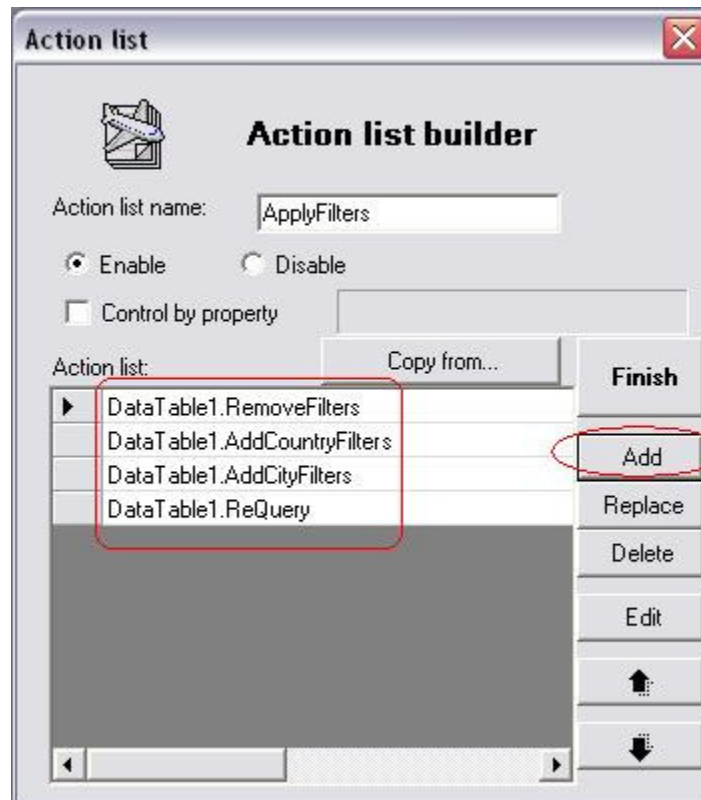


Accept default action name: DataTable1.ReQuery.

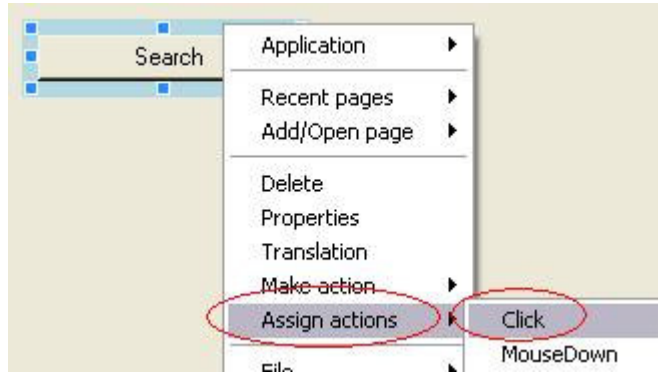
We have all actions needed. We may create an action list to include all these actions so that they can be executed one by one. Open Actions window, select Action Lists, click New button, give a name for the new action list, say, ApplyFilters:



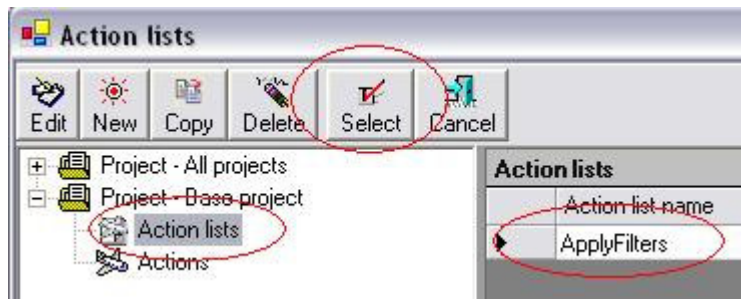
Use Add button to add the actions we want to include in this action list:



Now we have all actions ready. The question is when to execute action list ApplyFilters. In this sample, we use a button to do it. When the user clicks the button, ApplyFilters is executed: Create a button, right-click on it, choose “Assign actions”, choose “Click”:



Select Action Lists; select ApplyFilters; click Select button:



We are done.

Press F2 to run it. Check/uncheck selections, click Search button and watch what data it gives you.

