

Use Parameterized Query and Check Record Existence

Contents

The sample scenario.....	1
Create a parameterized query	1
Create an action to search database with parameters	3
Create an action use HasData as action condition.....	5
Create an action use NOT HasData as action condition	7
Assign actions to button	9
Test	10

The sample scenario

Such a scenario is stated in the forum “Existence in Database”:

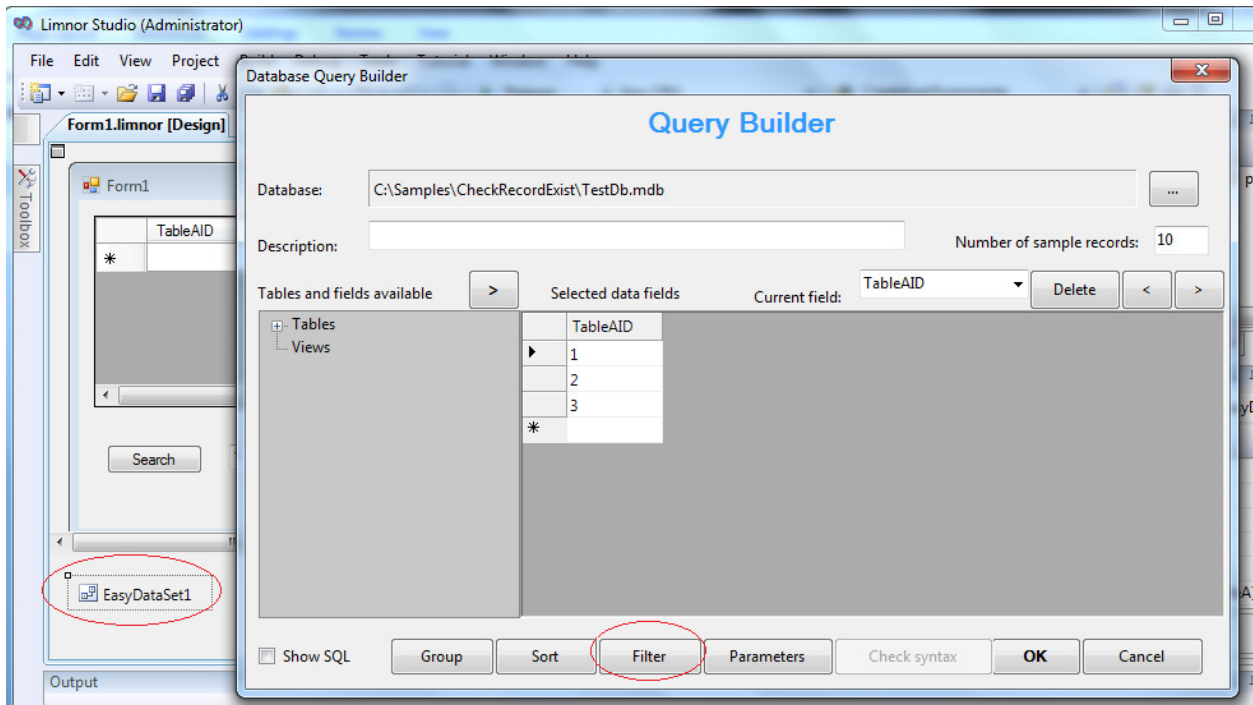
1. Let the user enter a record ID
2. If the ID does not match a record in the database then show a message box
3. If the ID matches a record in the database then close a form

It is assumed that you already went through the Users’ Guide for Database Applications:

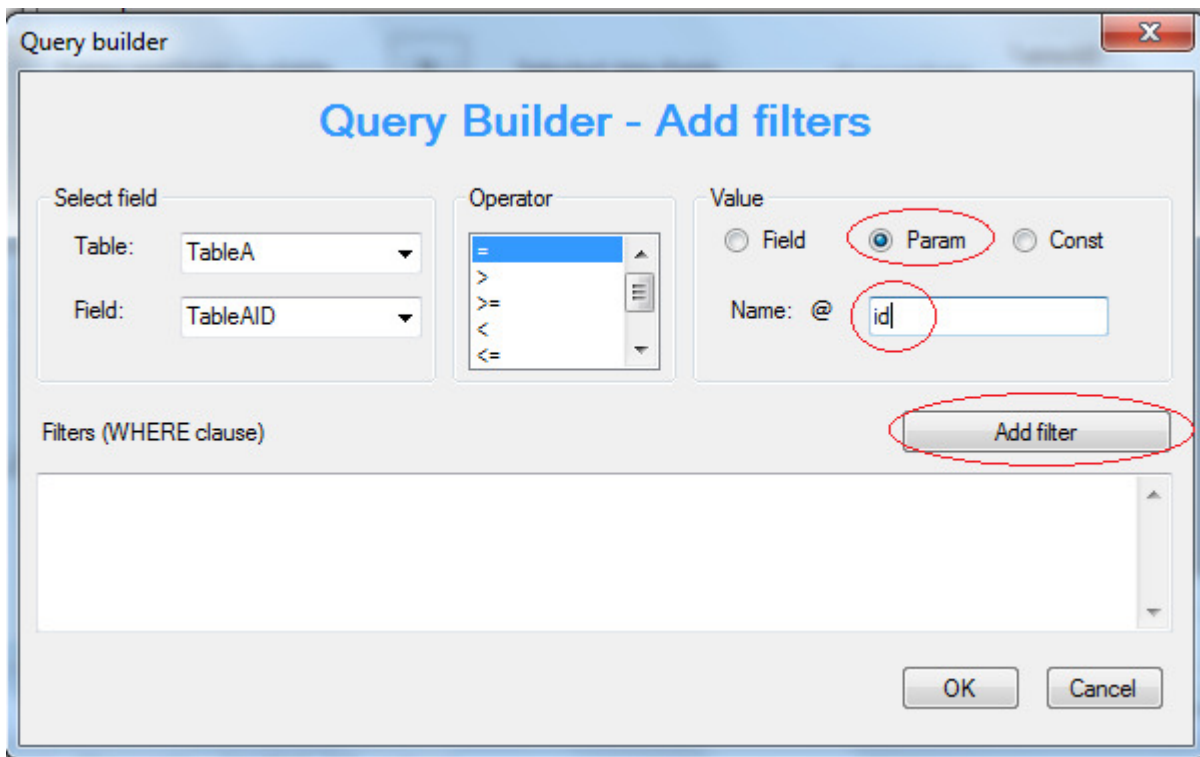
<http://www.limnor.com/support/Limnor%20Studio%20-%20User%20Guide%20-%20Part%20VI.pdf>

Create a parameterized query

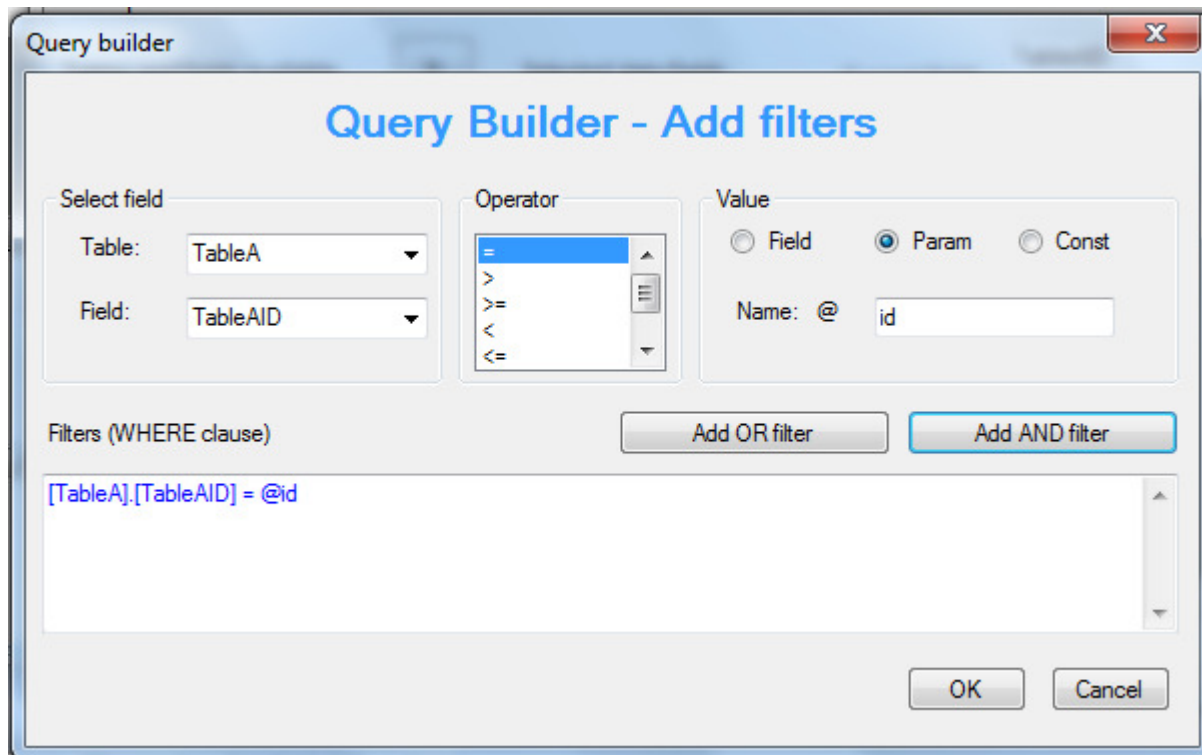
An EasyDataSet component is used to do the database search. Click Filter button to add filters to the query:



Create a parameter id to match TableAID field:

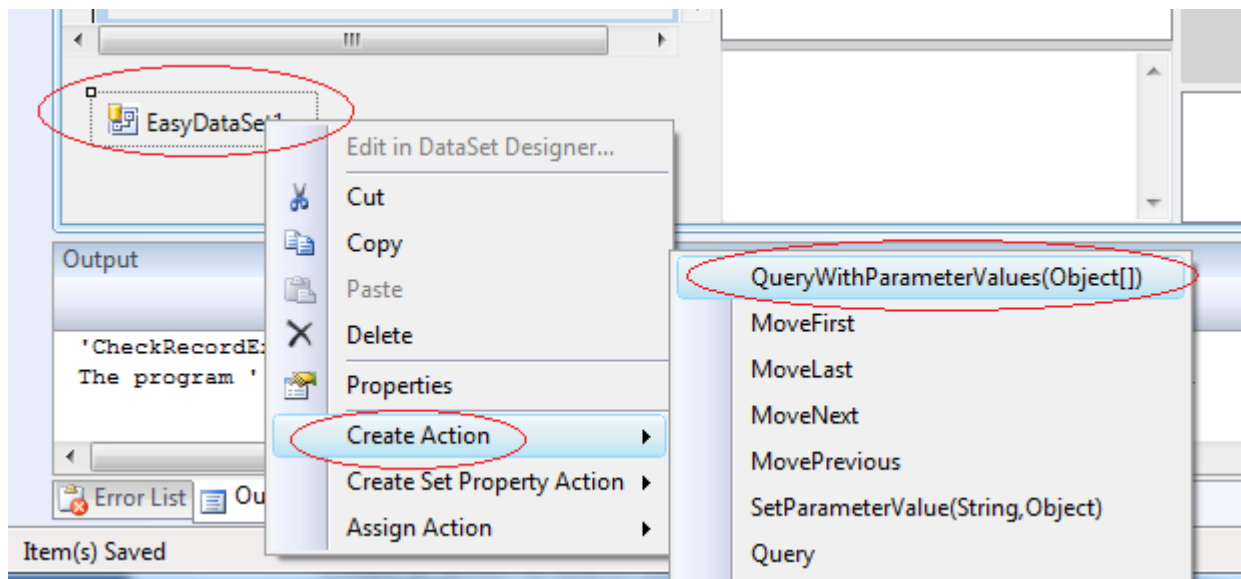


For this sample, we just need one filter:

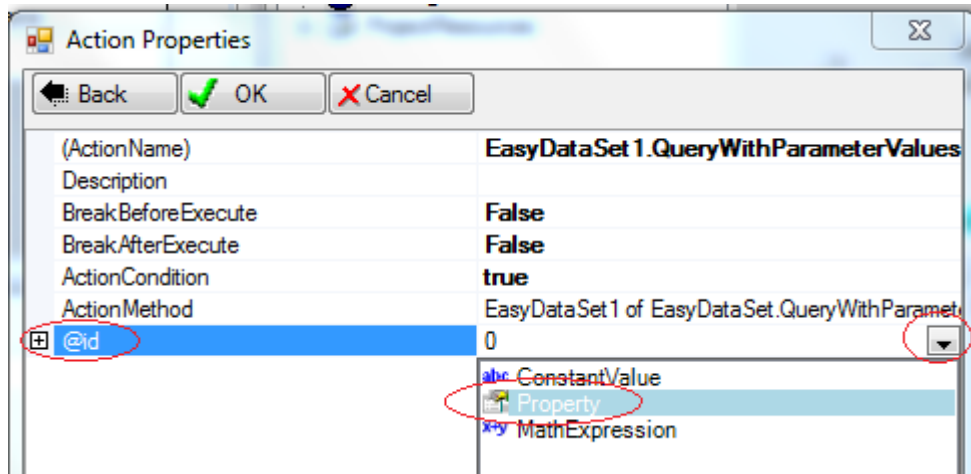


Create an action to search database with parameters

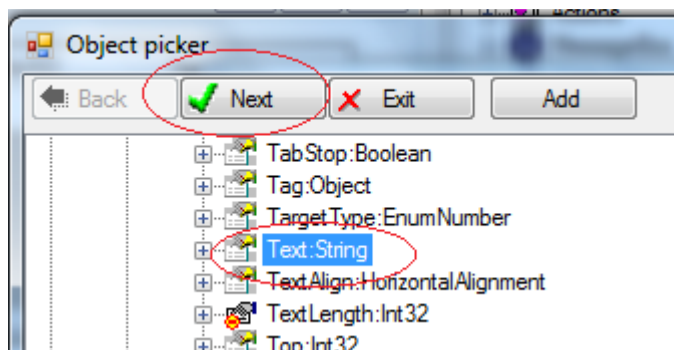
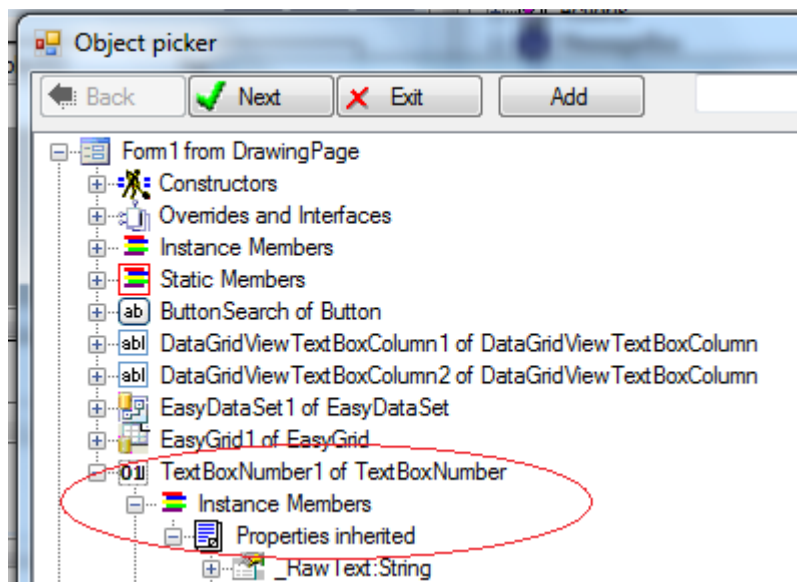
Right-click EasyDataSet1; choose Create Action; choose QueryWithParameterValues:



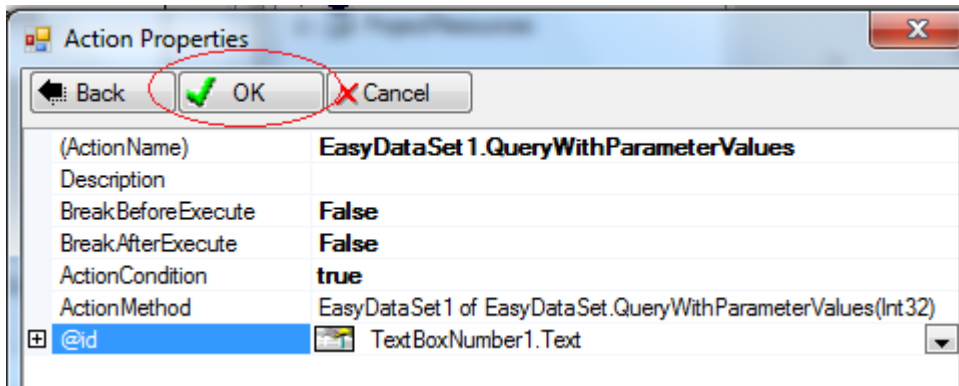
Select Property for parameter @id:



Select the Text property of the text box:

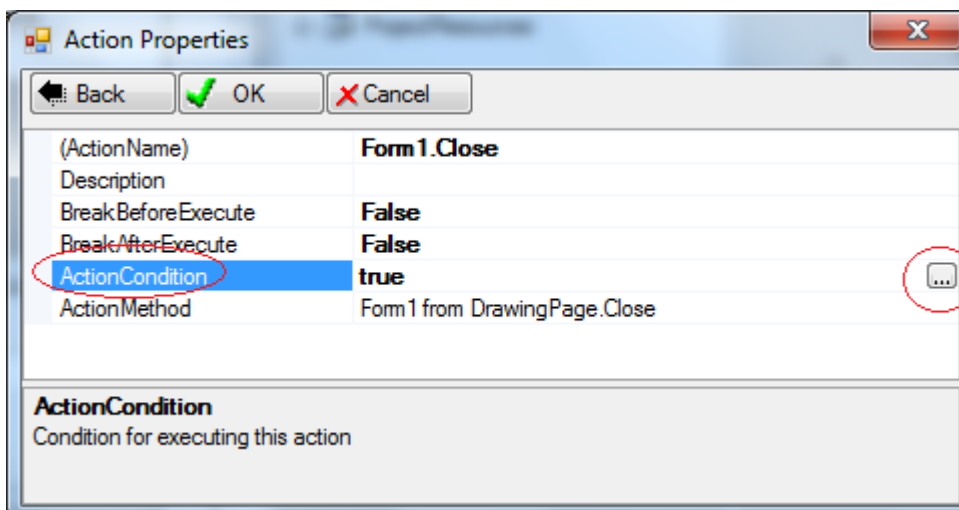


Click OK:

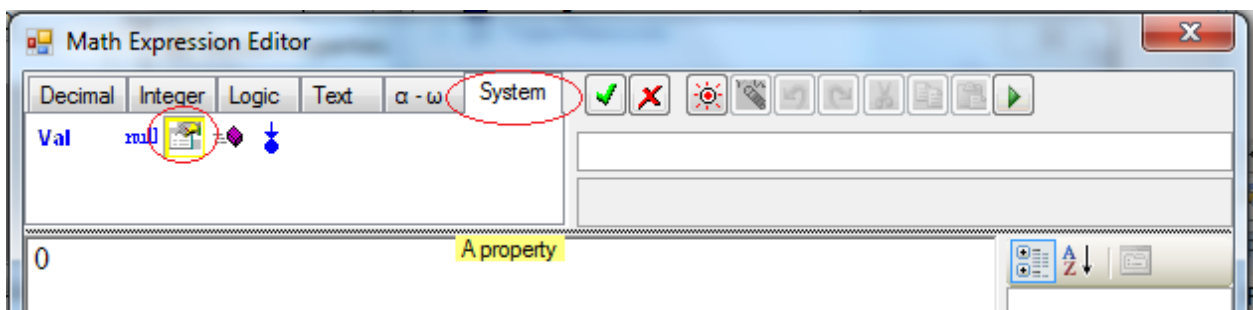


Create an action use HasData as action condition

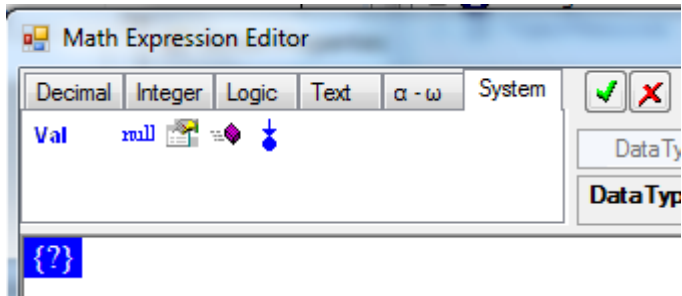
Create a Close action for a form, for simplicity, we use the current form. Set the action's ActionCondition property:



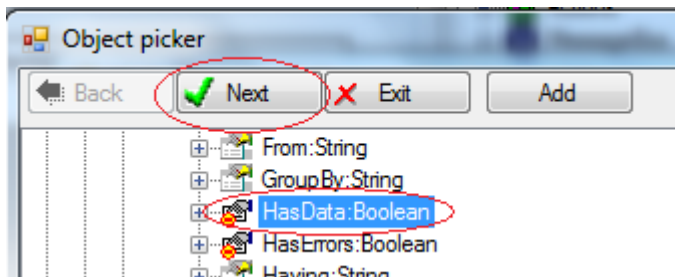
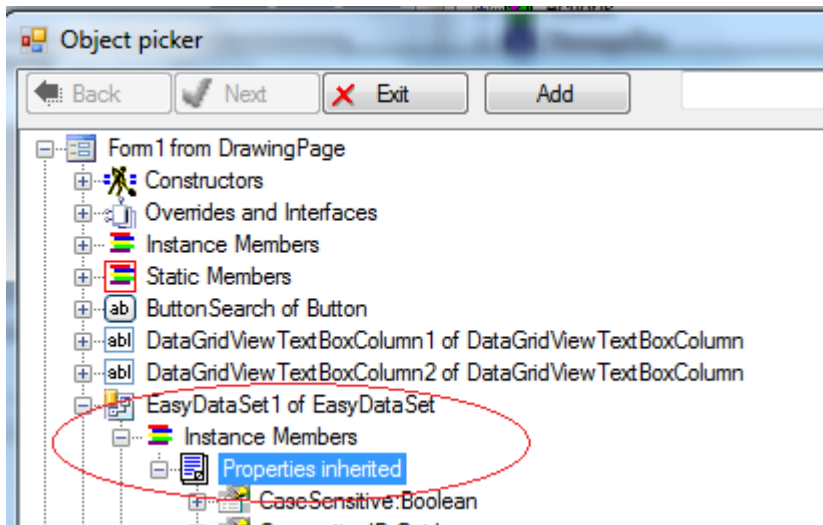
Click the Property icon:



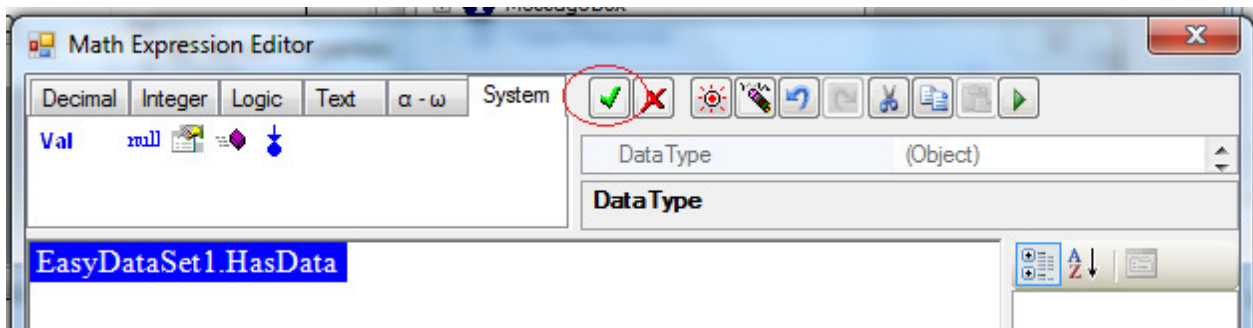
Double-click  to select a property:



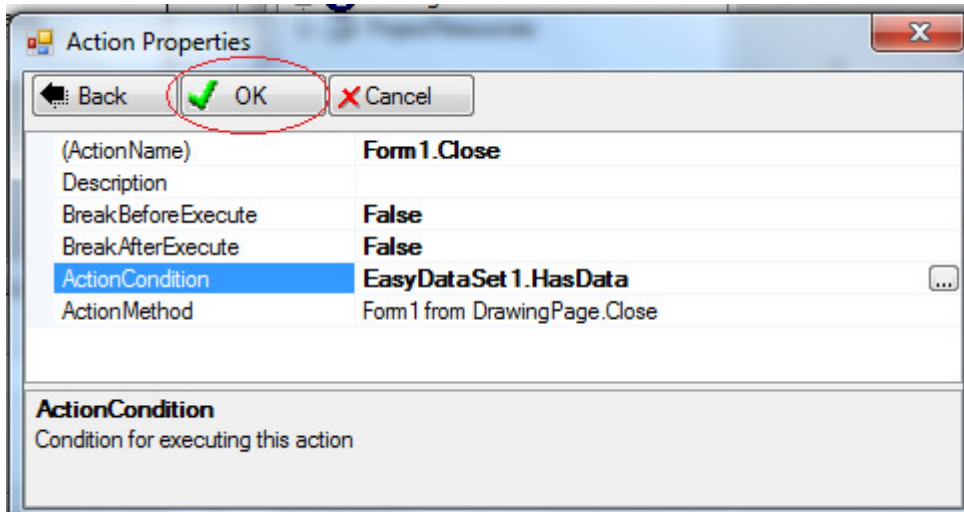
Select the HasData property of the EasyDataSet:



This is the action condition we want:

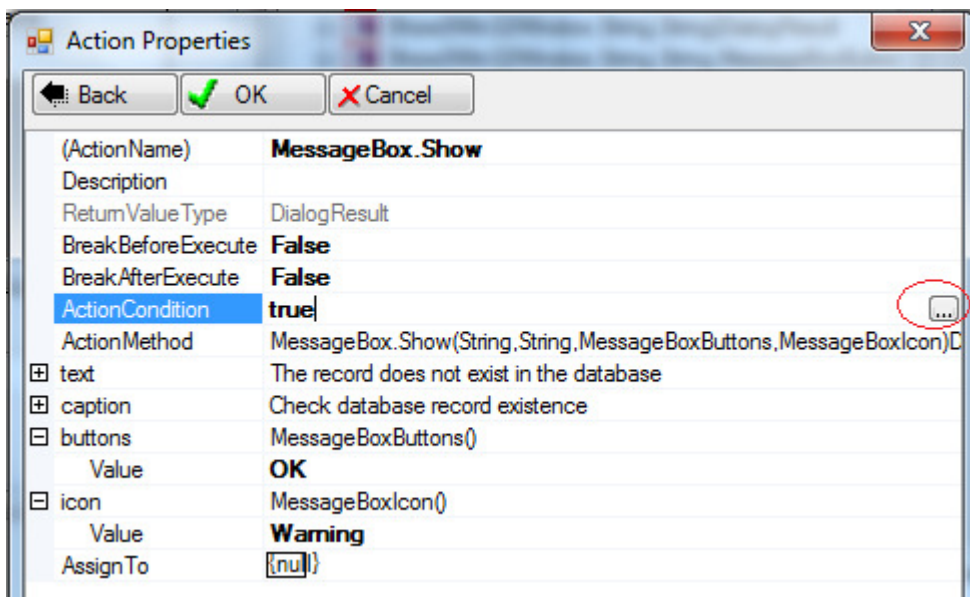


Click OK:

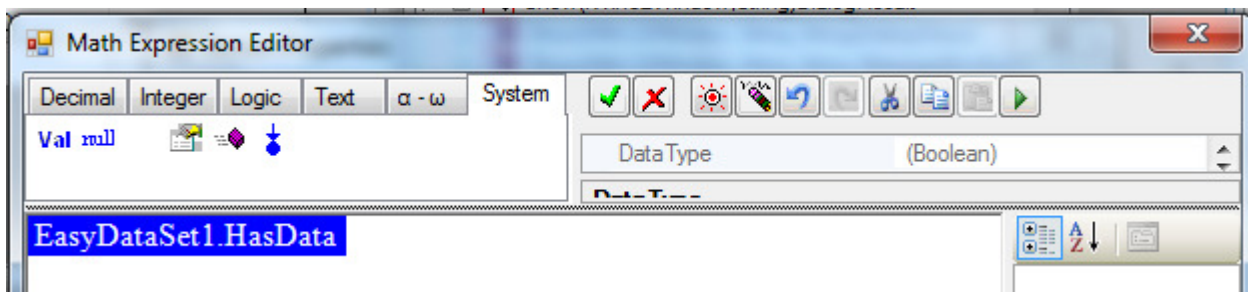


Create an action use NOT HasData as action condition

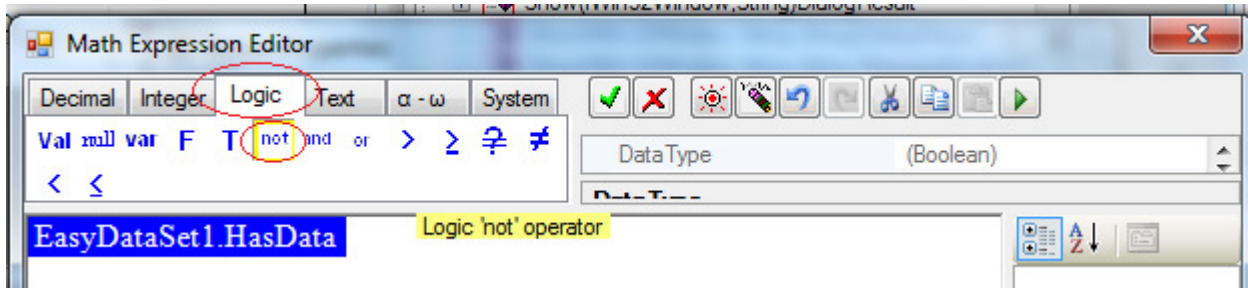
Use the MessageBox class to create an action to display message "record not found". Set its ActionCondition:



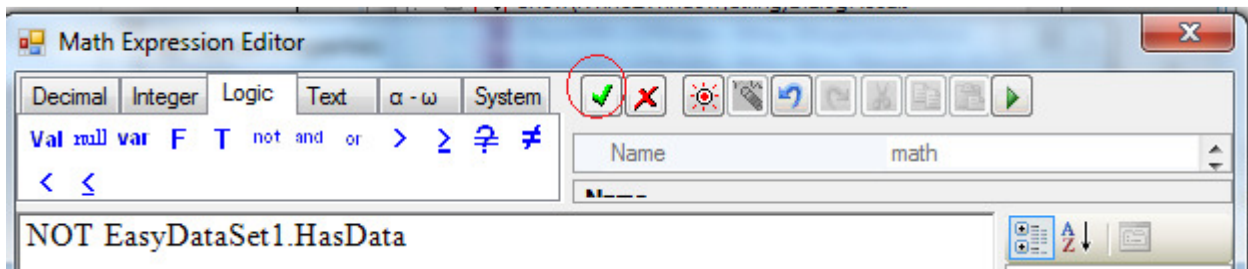
Use the HasData property of the EasyDataSet component, as we did in the last section:



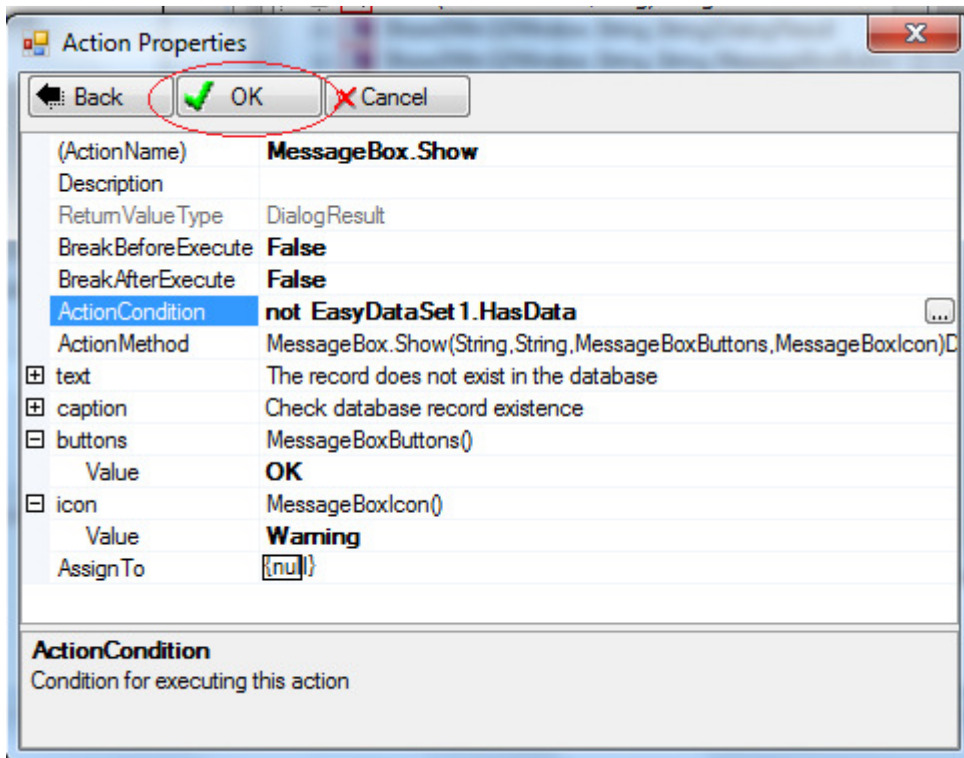
Click Logic tab and click "not" icon:



This is the action condition we want:

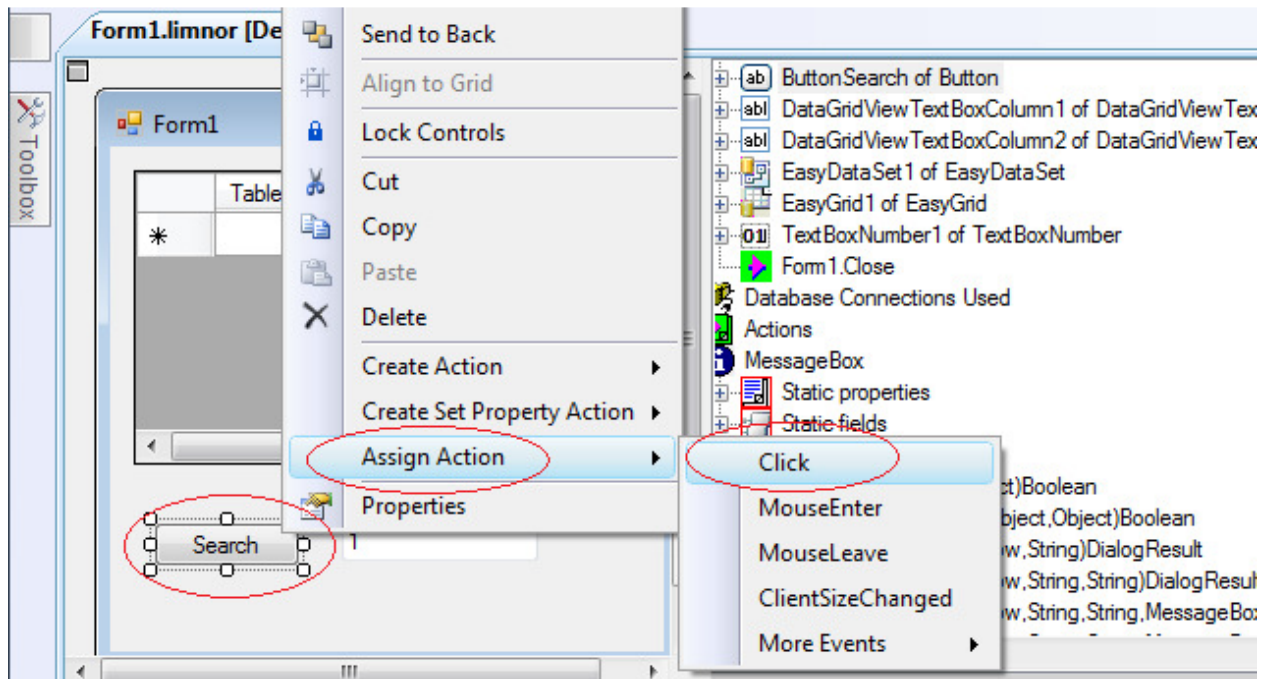


Click OK:

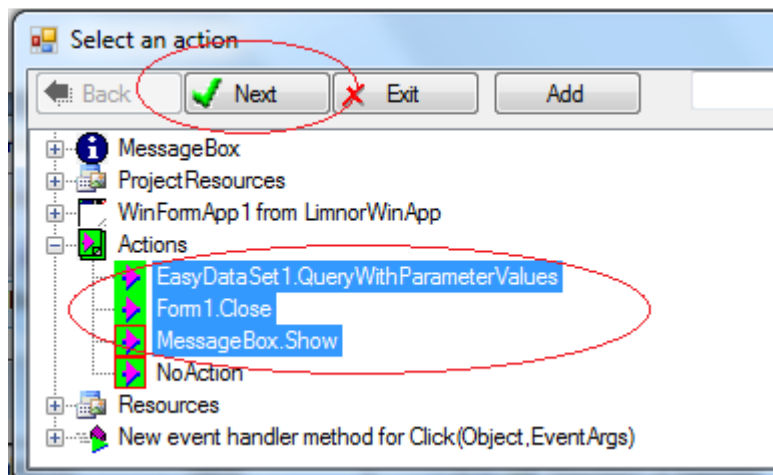


Assign actions to button

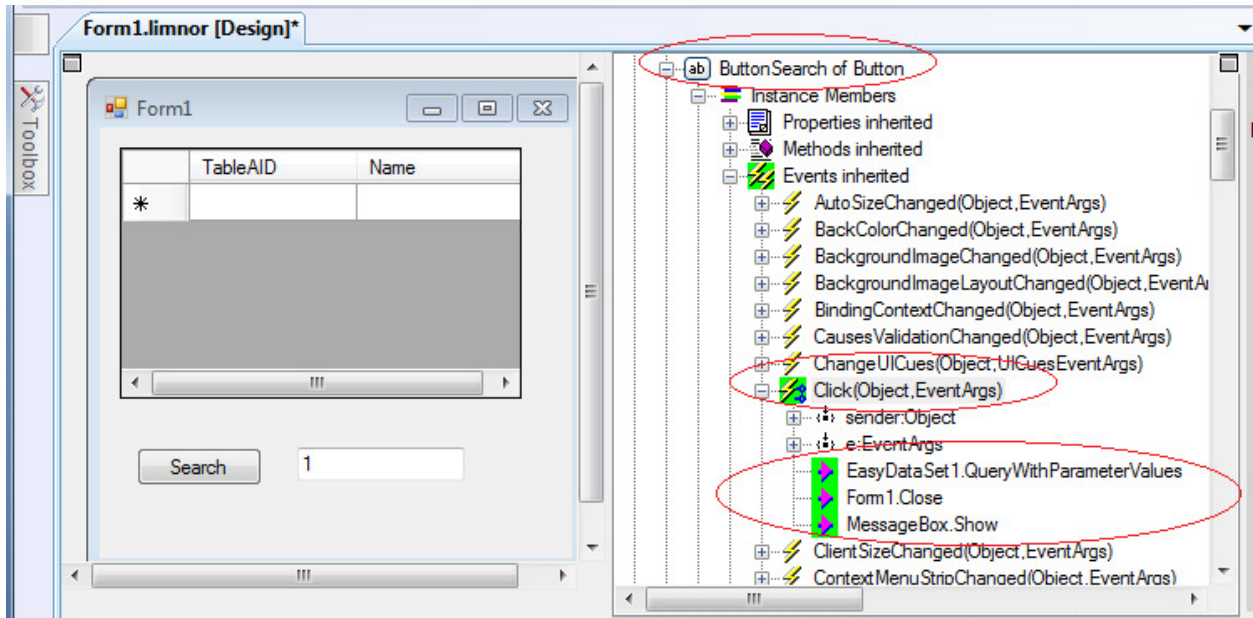
Right-click the button; choose "Assign Action"; choose Click:



Select the actions; click Next:



The actions appear under the Click event. Make sure that the action EasyDataSet1.QueryWithParameterValues should be the first action to be executed because the HasData property will has correct value only when the database query action is executed.

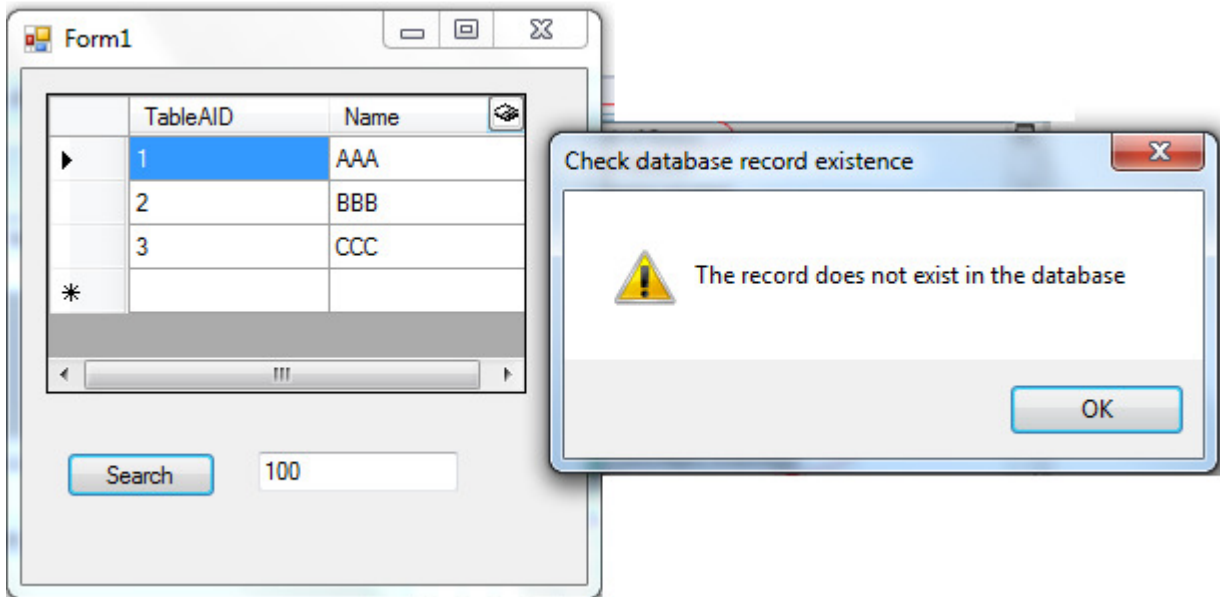


Test

We may test the sample now.

We use an EasyDataGrid to show what are in the database so that we may expect the test results.

We type 100 in the text box and click the Search button. We can see that "100" is not in the database. So, we expect to see the message box. And we do:



We type 2 into the text box. Because "2" is in the database we expect that the form will close. And it does close.

Form1

	TableAID	Name
▶	1	AAA
	2	BBB
	3	CCC
*		

Search 2

The image shows a software window titled "Form1" containing a table and a search interface. The table has two columns: "TableAID" and "Name". The first row has "1" in the "TableAID" column and "AAA" in the "Name" column. The second row has "2" in the "TableAID" column and "BBB" in the "Name" column. The third row has "3" in the "TableAID" column and "CCC" in the "Name" column. Below the table is a search bar with a "Search" button and a text input field containing the number "2". A red line connects the "2" in the search bar to the "2" in the table's "TableAID" column. The "2" in both the search bar and the table is circled in red.