

Data Entry Form

This sample shows the design of a data entry form for entering new records and modifying existing records. EasyDataSet and data-binding are used for database operations. It shows some design considerations. It assumes that you already familiar with visual programming in Limnor Studio.

Contents

Adding new records and modifying existing records	1
Use of dialogues.....	1
Form for adding new record	5
Create database query.....	5
Set data-binding	6
Data-validation: check record existence.....	7
Method for adding new record	8
Setting initial state.....	10
Form for modifying existing records	11
Create database query.....	11
Set data-binding	12
Search Method	12
Save modifications to database	13
Set form initial state	14

Adding new records and modifying existing records

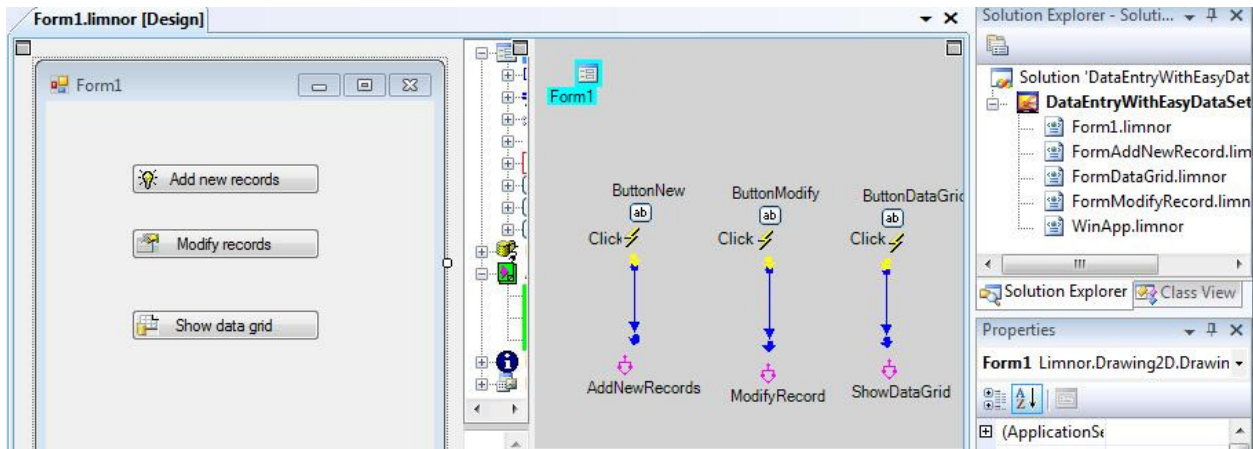
In this sample, we create two forms. One form is only for adding new records. One form is only for updating existing records. This separation of tasks makes it easier to consider the issues related to adding new records and modifying existing records.

Use of dialogues

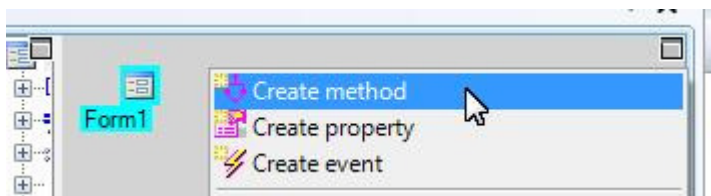
A form can be displayed using its Show action or ShowDialog action. A form becomes a dialogue box if the form is displayed using a ShowDialog action. The user cannot access other forms of the same application until the dialogue box is closed.

Using dialogues is easier in programming because there is not a need to manage the form instances.

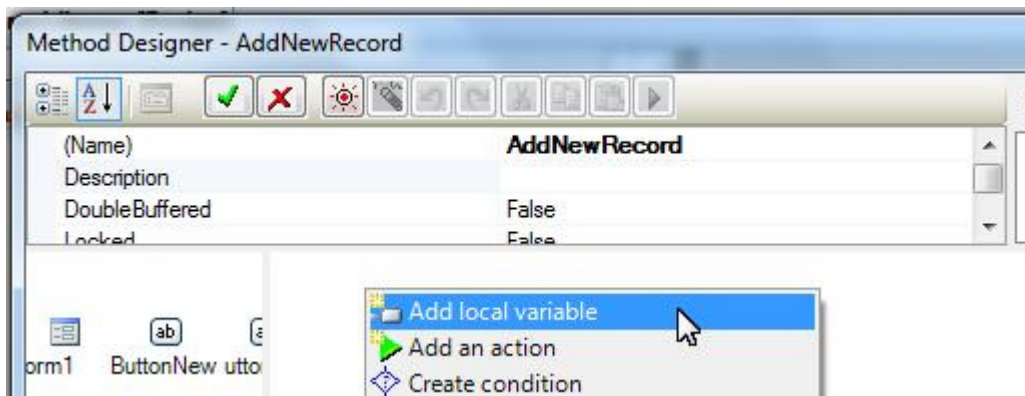
In this sample, the main form has some buttons. Each button is for opening a dialogue box.



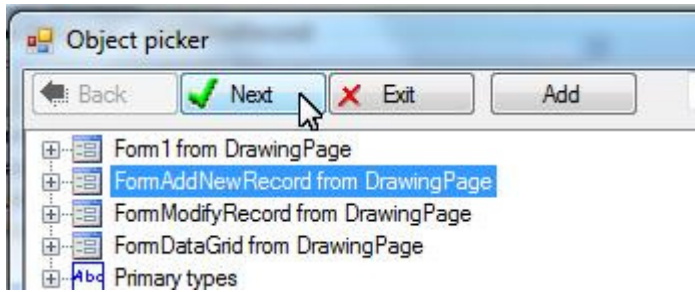
We'll show the programming of one dialogue box. One method is created for showing a dialogue box.



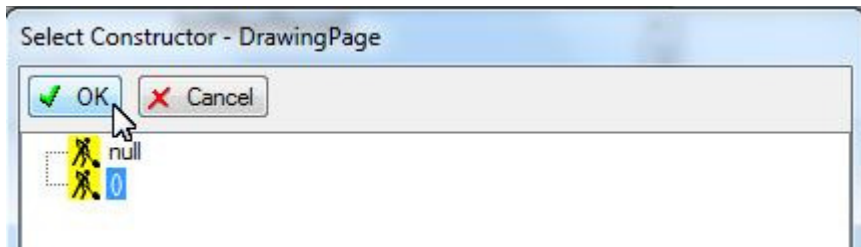
Create a variable of the Form to be used as the dialogue box:



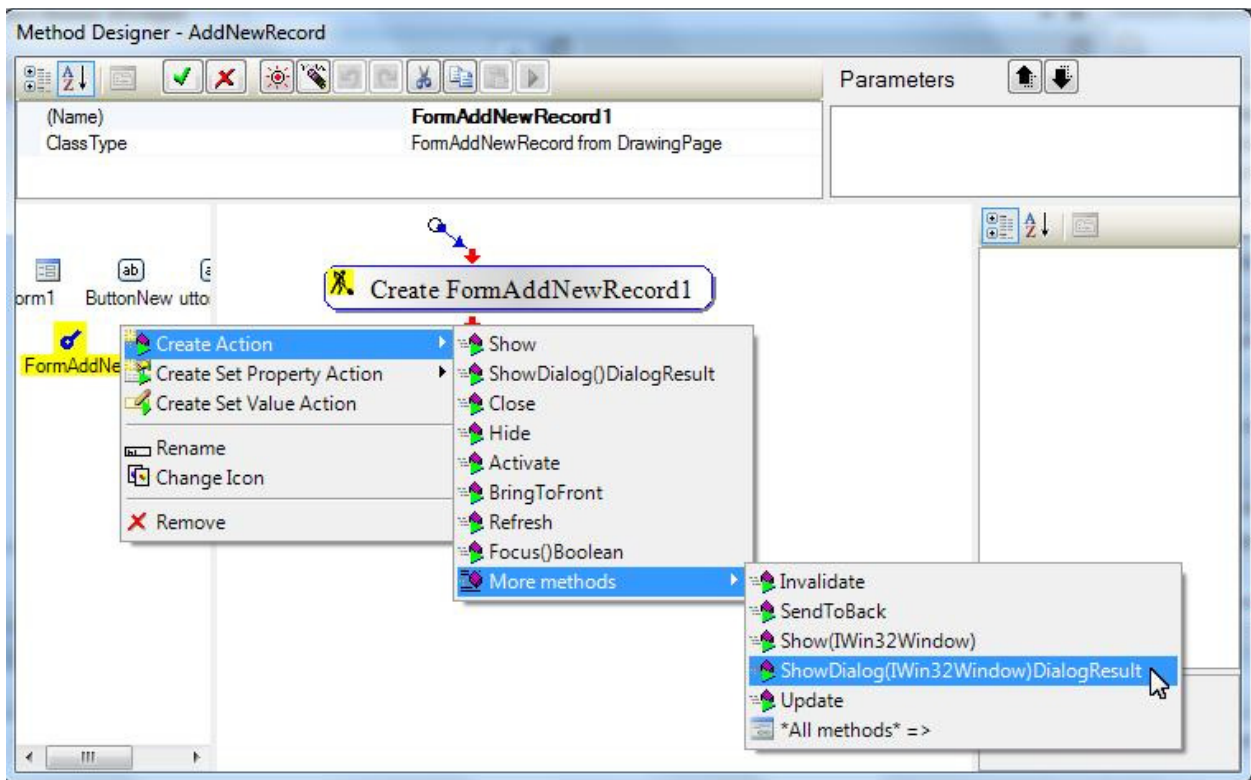
Select the form to be used as the dialogue box:



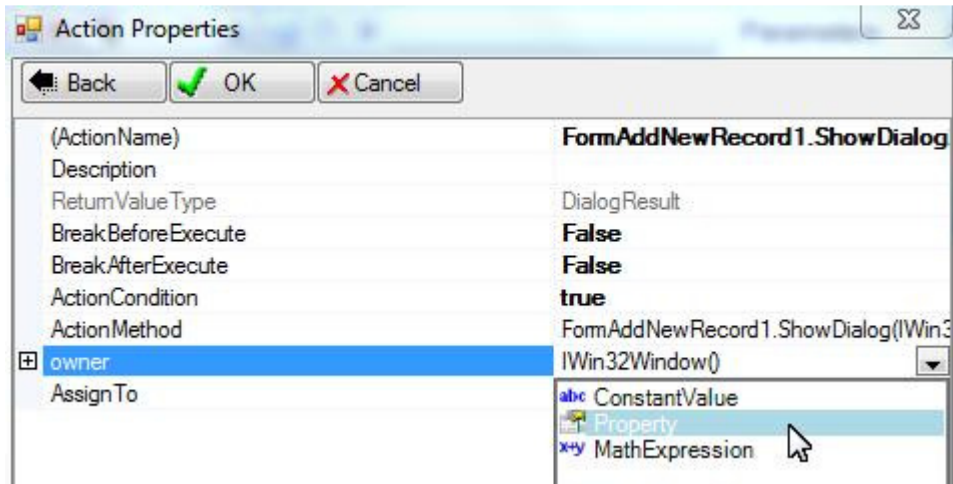
Select a construct to generate the form instance:



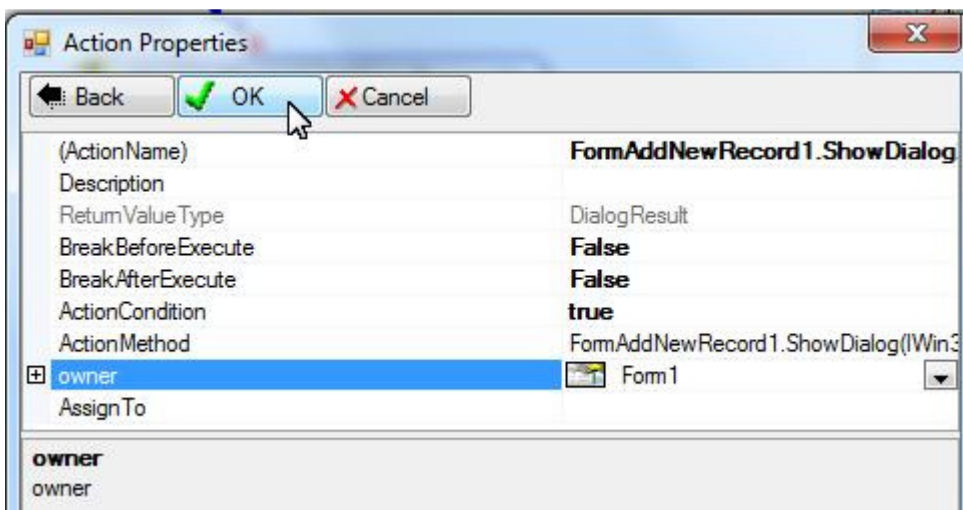
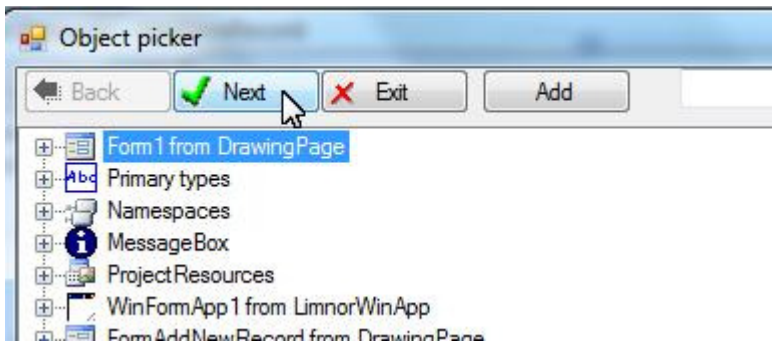
An action appears in the Action Pane. The action creates an instance of the form. The instance of the form appears in the variable pane. Right-click the form instance; choose "Create Action"; choose method "ShowDialog(IWin32Window)":



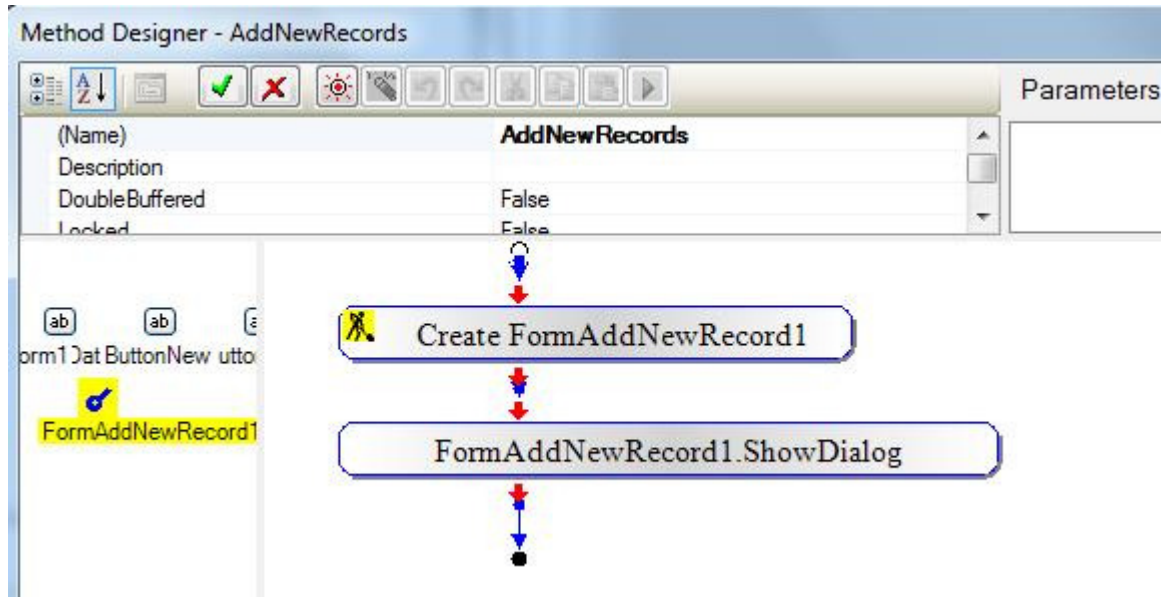
We selected method ShowDialog(IWin32Window), not ShowDialog(). ShowDialog(IWin32Window) allows us to specify owner form for the dialogue box. In the sample, we want to use Form1 as the owner form. Select "Property" to specify the owner form:



Select Form1 as the owner:



This is the method to show a dialogue box. It first creates an instance of the form; then it shows the form instance as a dialogue box:



To execute this method, create an action using it. Right-click the method, choose “Create Action”. Then assign the action to the Click event of a button.

Form for adding new record

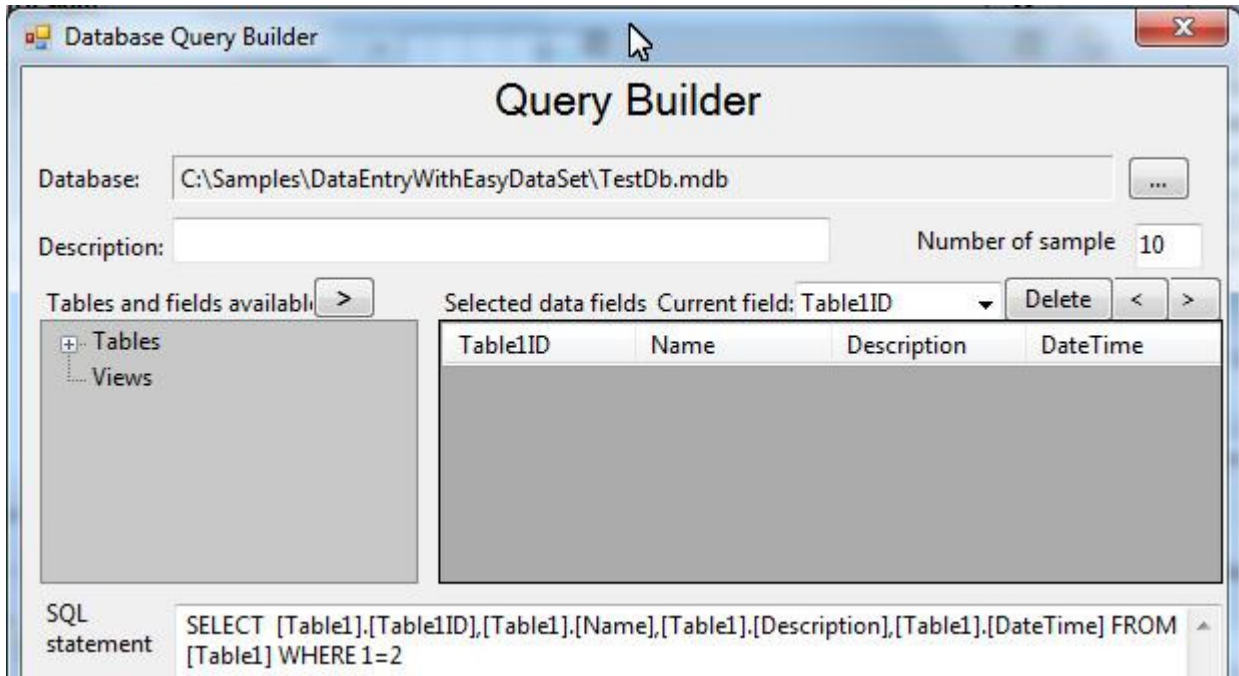
To use an EasyDataSet to insert new records, following basic steps are involved:

1. For the EasyDataSet, create a database query for linking to the database fields
2. Set data-binding to controls
3. Execute an AddNewRecord action to create a new record in memory and allow the user to enter data for the new record
4. Validate the data the user enters and execute an Update action to save the new record in database.

We’ll go into details on each step.

Create database query

For an EasyDataSet to modify data in the database, its fields must include an auto number or fields forming a unique index. In this sample, Table1ID is an auto number.

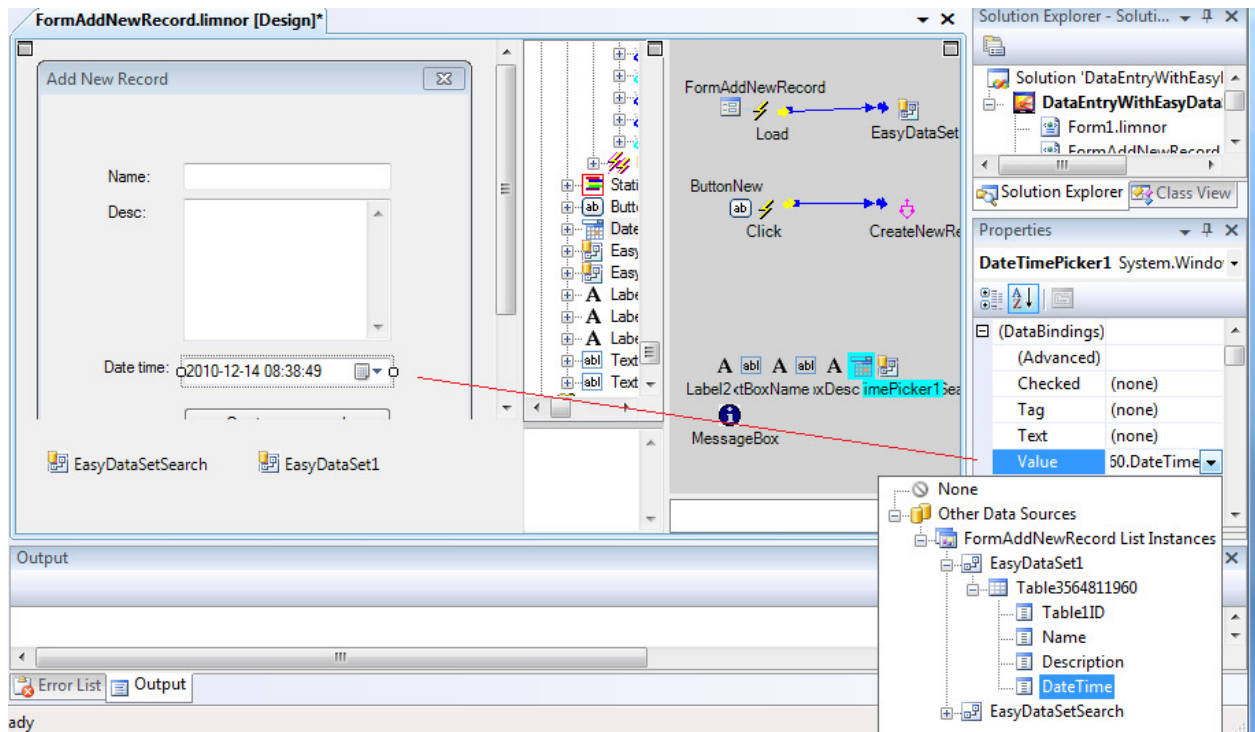


When designing a database query, in most cases we need to use filters to limit the records to be fetched from the database because usually we do not want fetch all records.

In this case we do not want the query to return any record because this form is only used for adding new records, not for displaying existing records. To prevent the query from fetching any records we used a special filter: "1=2". No records will satisfy this condition and thus no records will be fetched by this query.

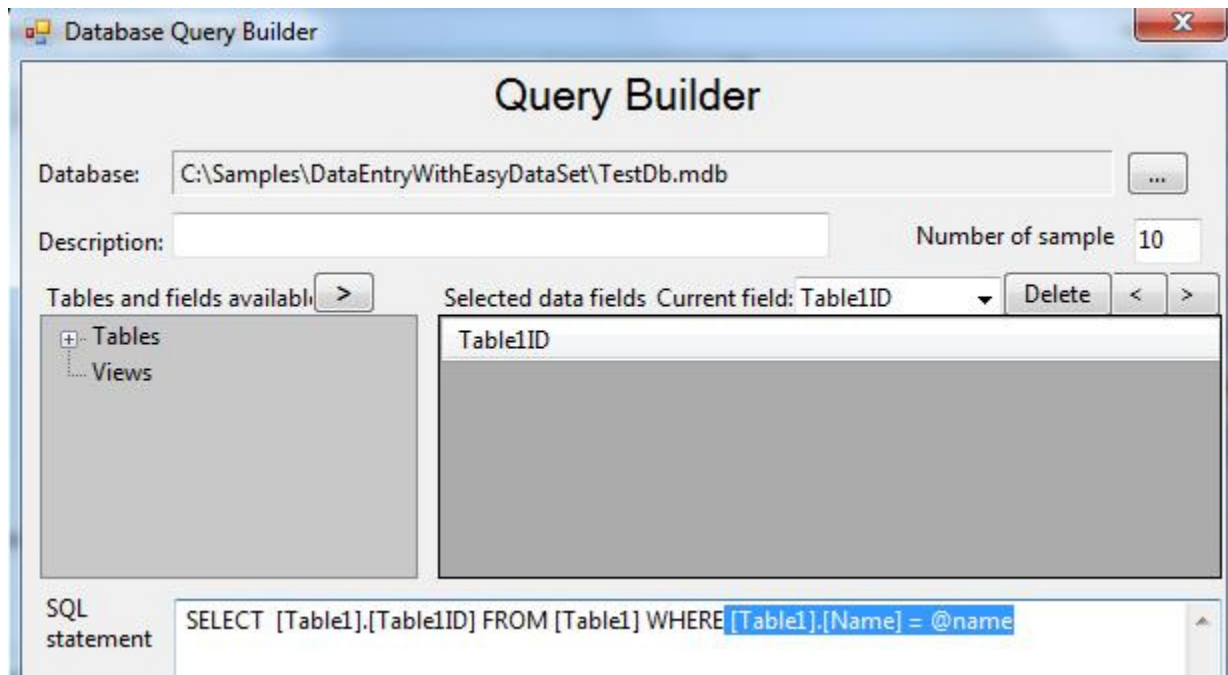
Set data-binding

For a text box, data-binding is on its Text property. For a datetime picker, data-binding usually is on its Value property:

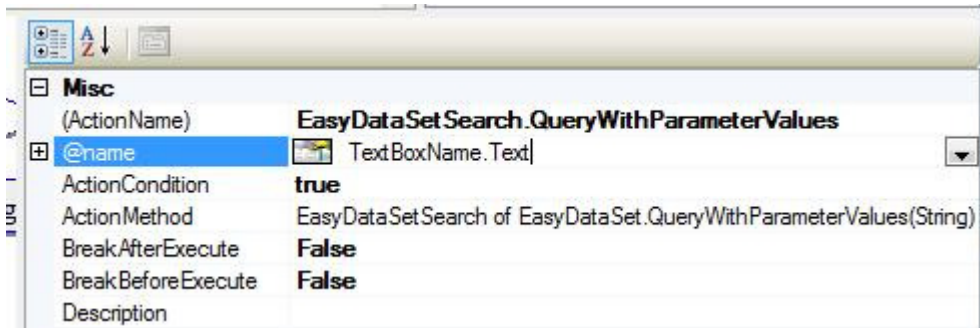


Data-validation: check record existence

Suppose the Name field is a unique field. We need to check the existence of it before creating a new record. We may use another EasyDataSet to do this checking. We name it as EasyDataSetSearch. What field it returns does not matter. The important thing is that it has a filter "[Table1].[Name] = @name".



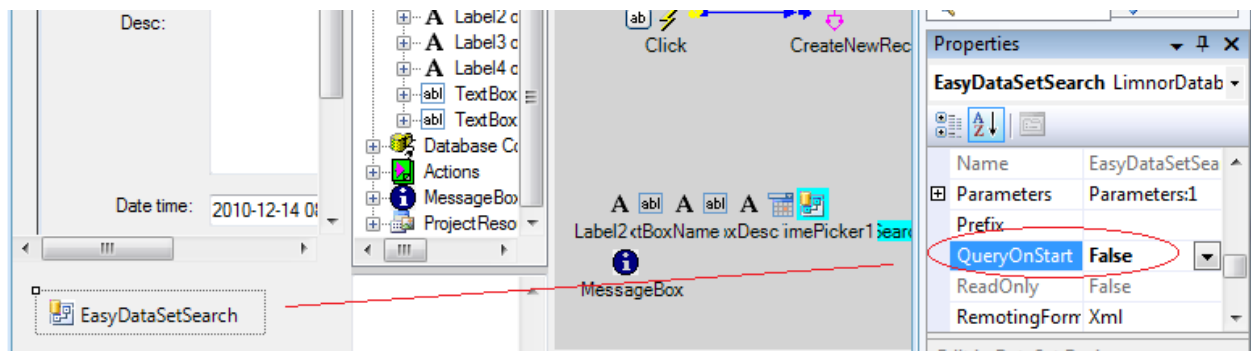
We may create a QueryWithParameterValues action to search the database using the Text property of the text box for the Name field:



After executing QueryWithParameterValues the Hasdata property of EasyDataSetSearch indicates whether the Name exists in the database.

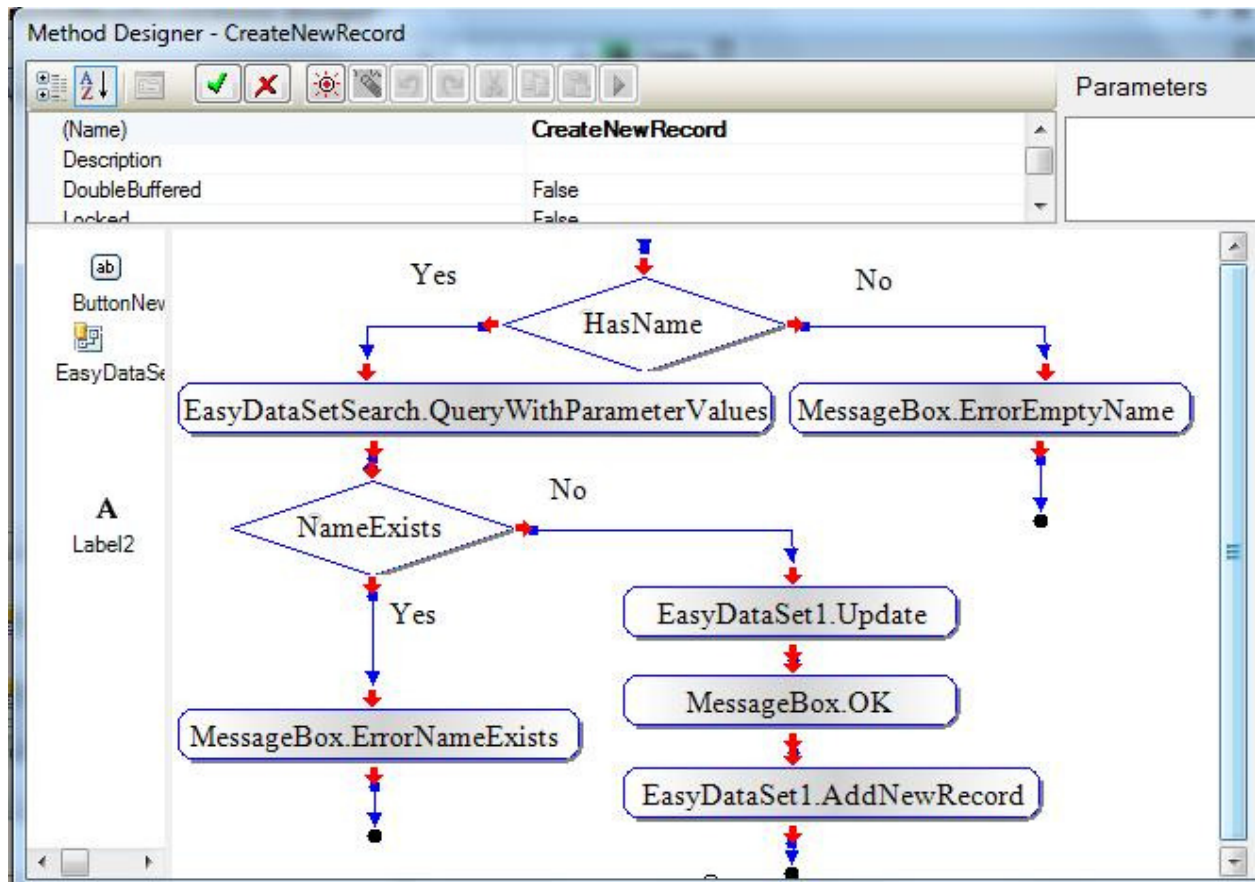
The QueryOnStart property should be False for EasyDataSetSearch because we do not want it to query the database when the form is loaded. We only want it to do query when action QueryWithParameterValues is executed.

Note that if there are data-bindings on an EasyDataSet then QueryOnStart should be True because data-binding needs data definition from database.



Method for adding new record

We use a new method to include the actions need for adding a new record to the database.

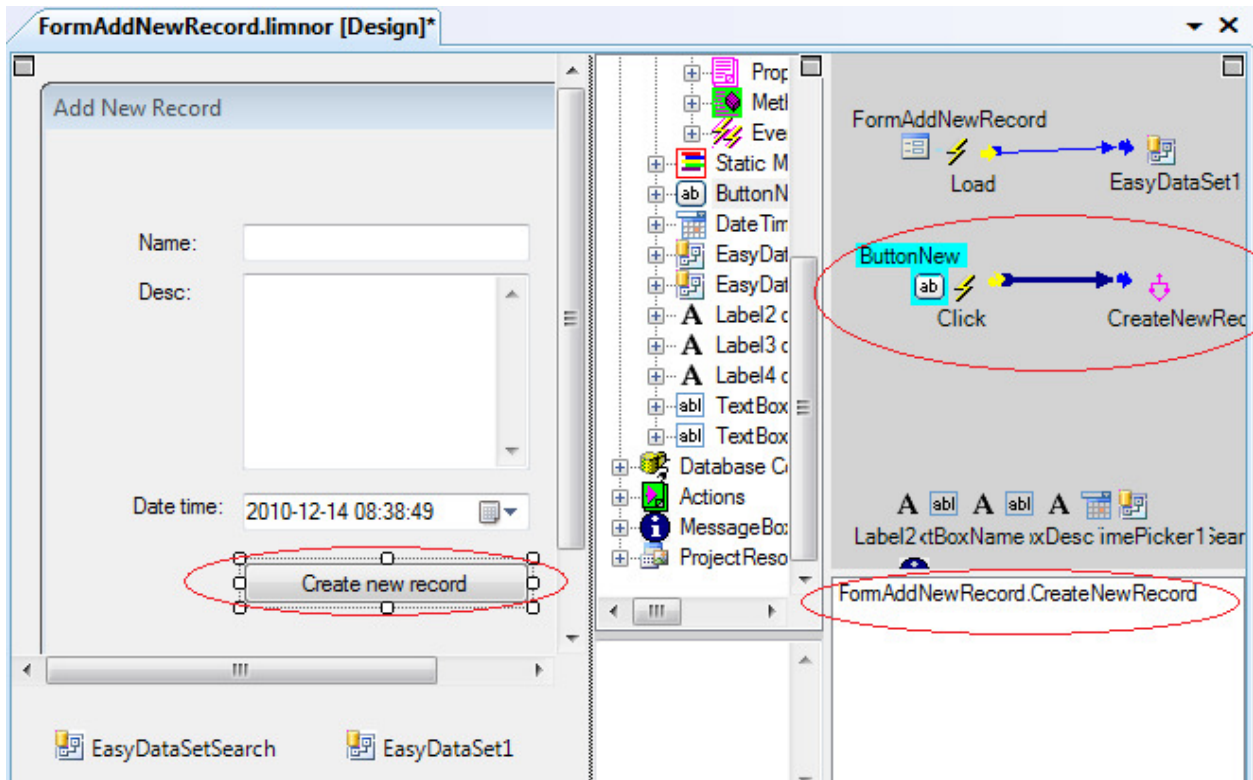


The first action is to check if the Text property of the text box for the Name field is empty. If it is empty then a message box appears saying that the Name cannot be empty. The method finishes.

If the Name is not empty then QueryWithParameterValues is executed to check if the name exists. If the name exists then a message box appears saying that the name exists and the method finishes.

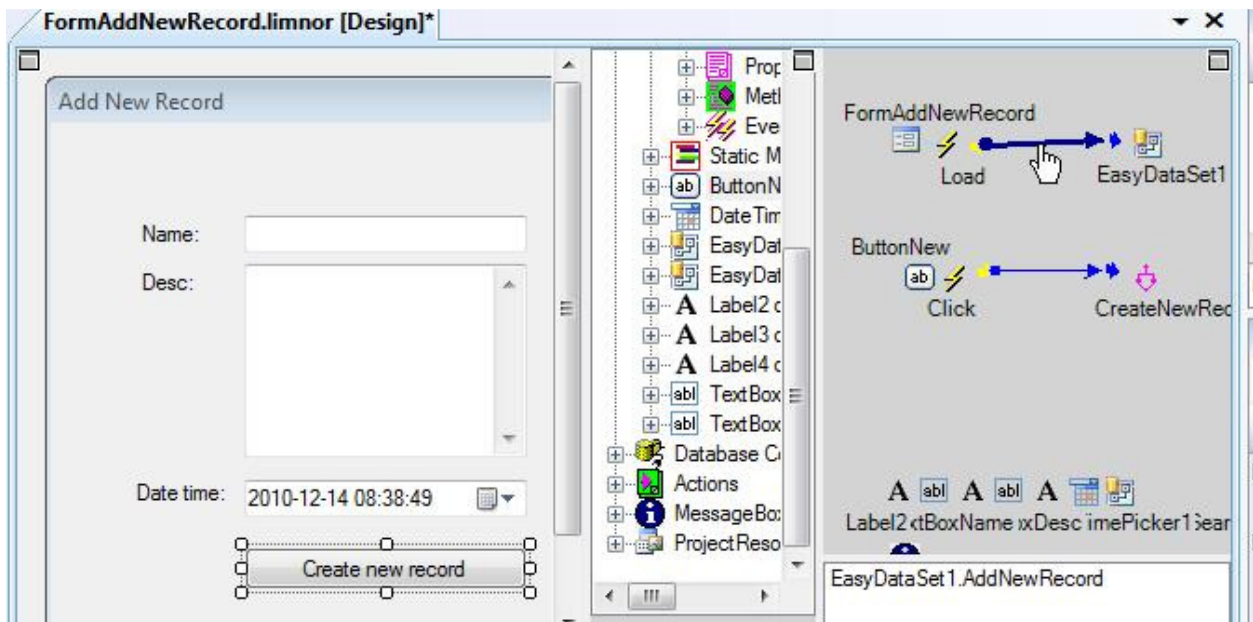
If the name entered by the user does not exist then Update action is executed to create the record in the database. A message box appears saying that the new record is created. The last action is AddNewRecord. This action creates a new record in memory and allows the user to enter another new record.

An action is created using this method. The action is assigned to the button.



Setting initial state

When the form is loaded we want it to be ready for the user to enter data for a new record. This can be done by assigning the AddNewRecord action to the Load event of the form:



The Load event occurs when the form is fully loaded. At the event executing AddNewRecord action puts the form in a state of entering data for a new record.

Form for modifying existing records

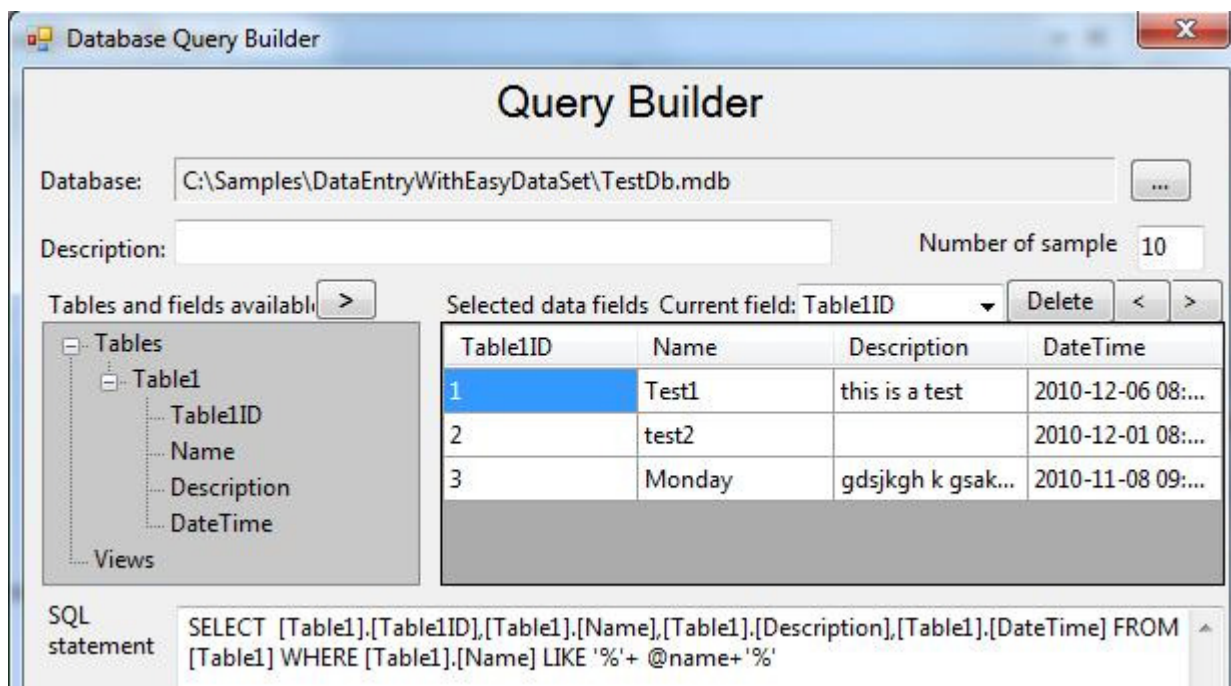
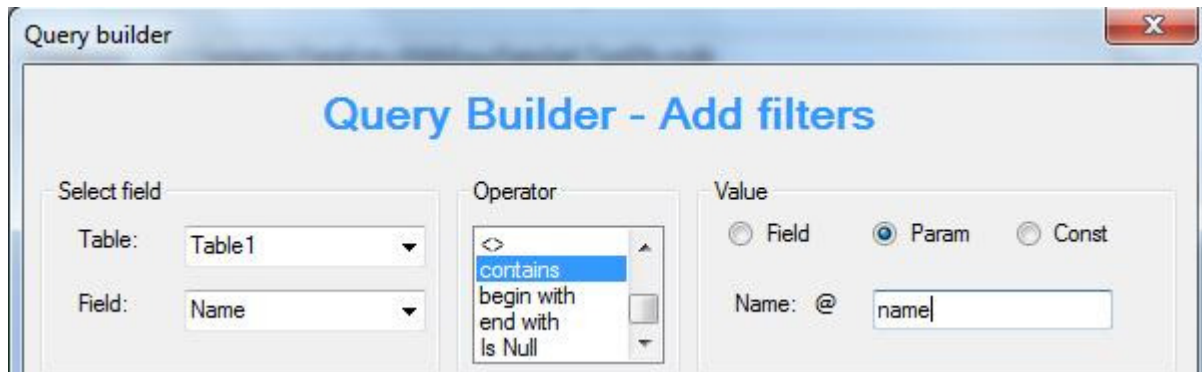
Modifying an existing record is easy:

1. Create a query to search the database for the records
2. Use data-binding to show the record to be modified
3. Execute Update action to save modifications to the database.

Create database query

For an EasyDataSet to modify data in the database, its fields must include an auto number or fields forming a unique index. In this sample, Table1ID is an auto number.

For this sample, we use “contains” filter to do the search. Any search text contained in the Name field matches the search condition

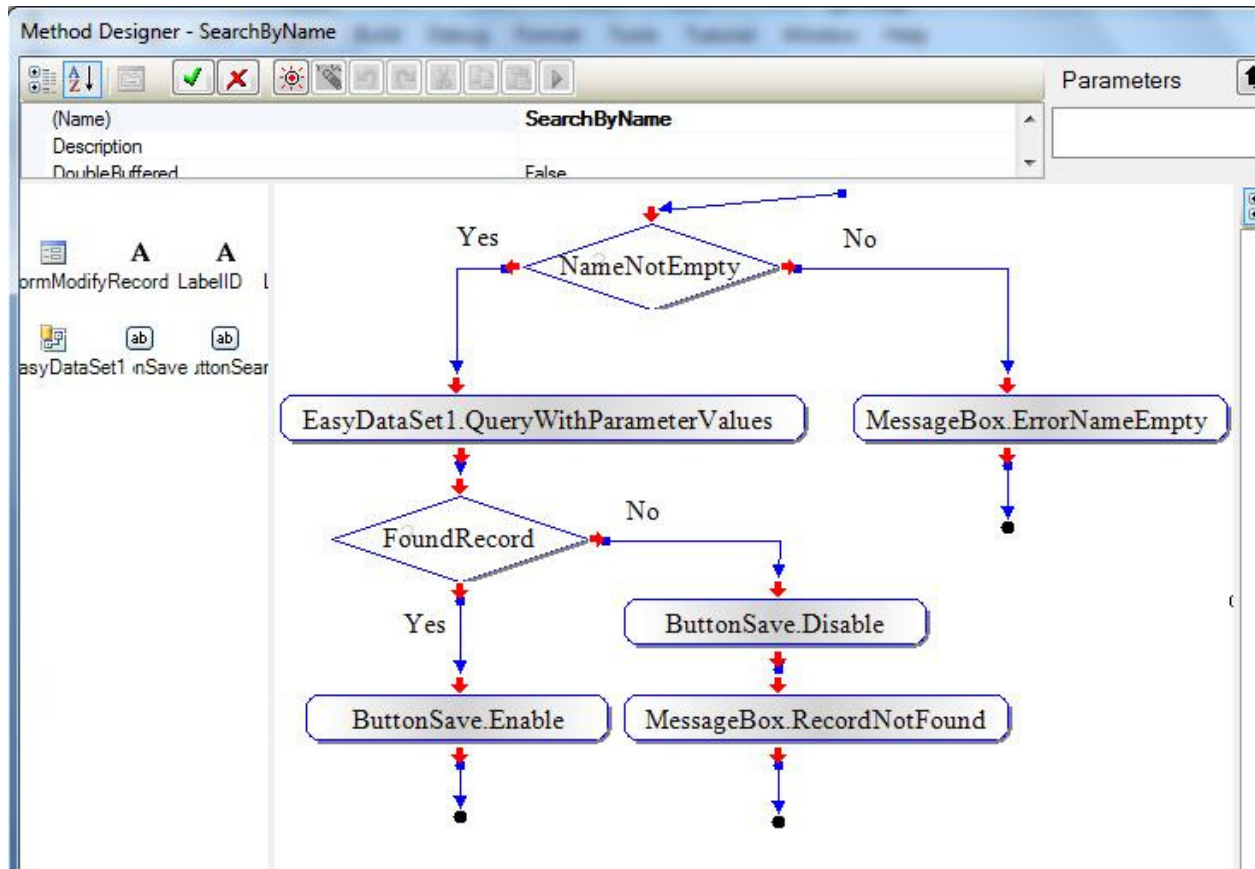


Set data-binding

It is the same as we did for adding new records.

Search Method

We use a new method to search for record to be modified and put the form in a proper state.

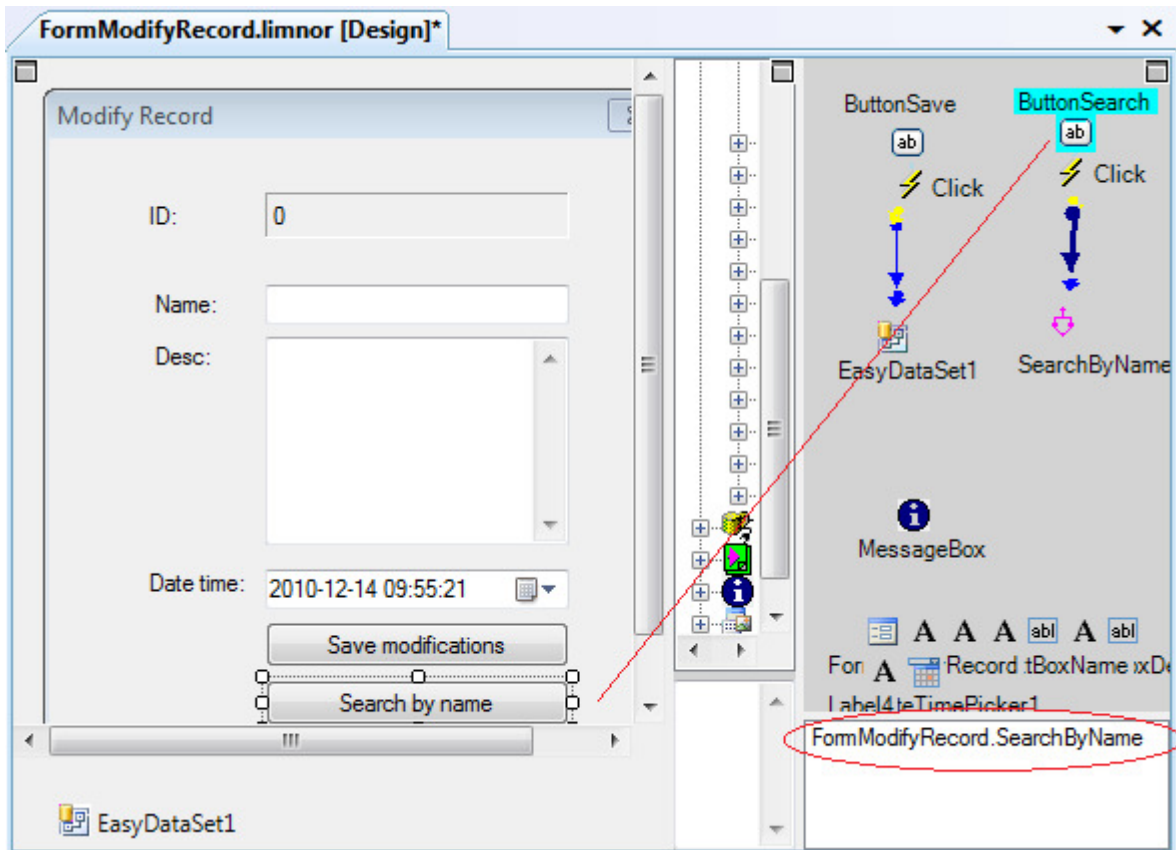


The first action is to check if the Text property of the text box for the Name field is empty. If it is empty then a message box appears saying that the Name cannot be empty. The method finishes.

If the Name is not empty then QueryWithParameterValues is executed to search for the record. If the record cannot be found then the Save button is disabled to prevent the user from clicking it. A message box appears saying that no record is found and the method finishes.

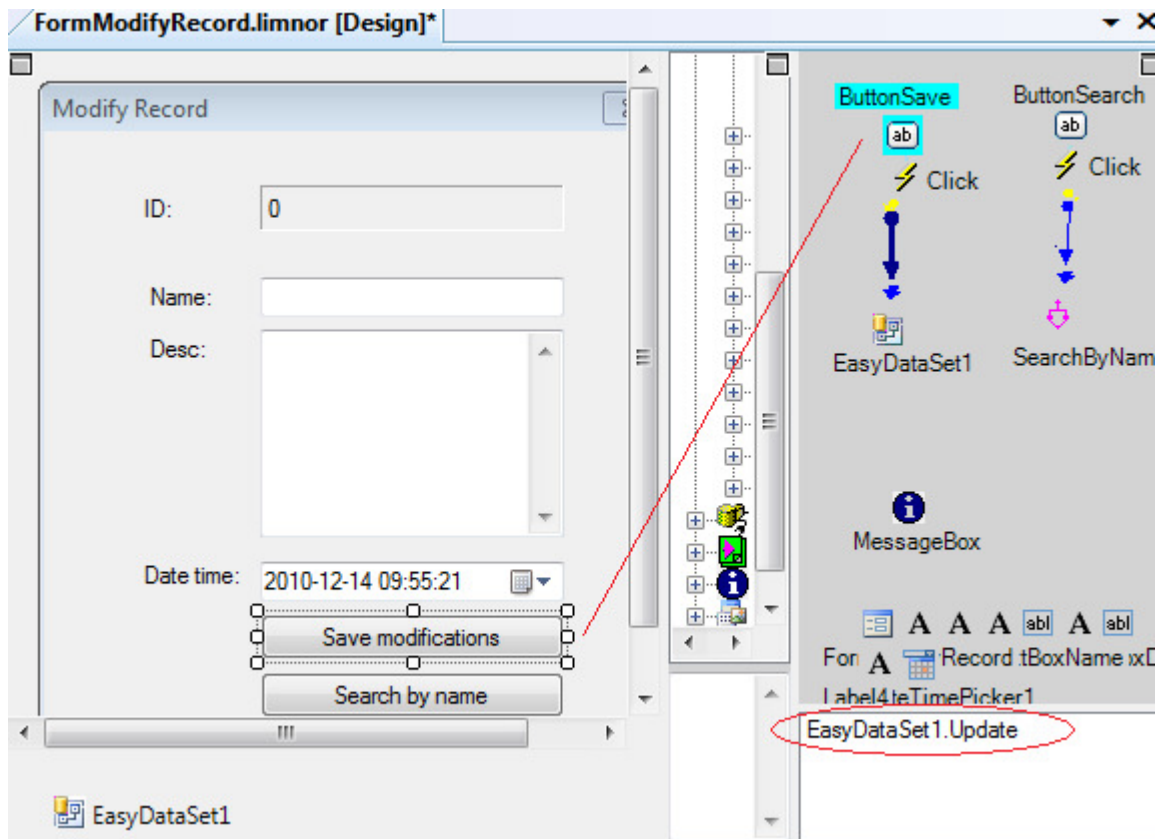
If a record is found then the Save button is enabled. At this time the record found is displayed in the controls bound to the EasyDataSet.

We may create an action using this method and assign the action to a button:



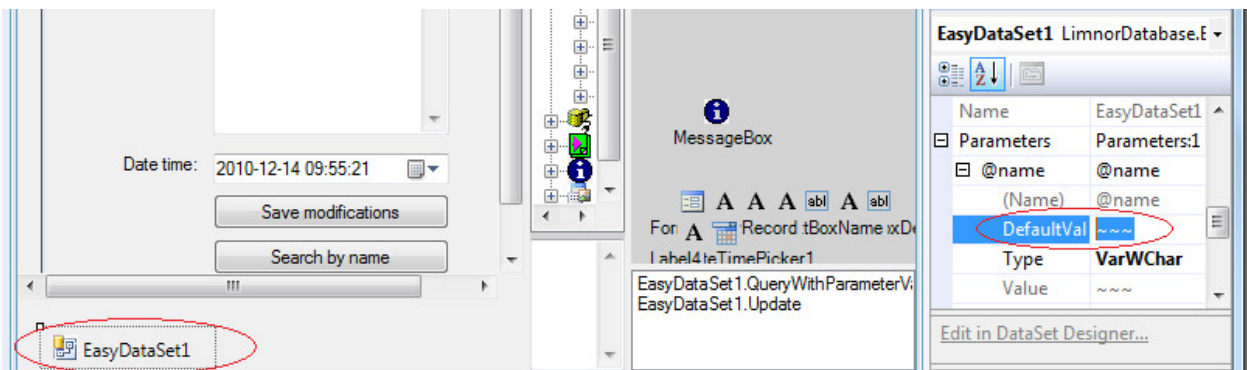
Save modifications to database

We may create an Update action and assign the action to a button.



Set form initial state

When the form is loaded we do not want it to display a record. We may set the default value for the search parameter to an invalid text for Name:



Because we purposely make that no record is found when the form is loaded, we need to set the Enabled property of the Save button to False.

