

Image Data Binding

Contents

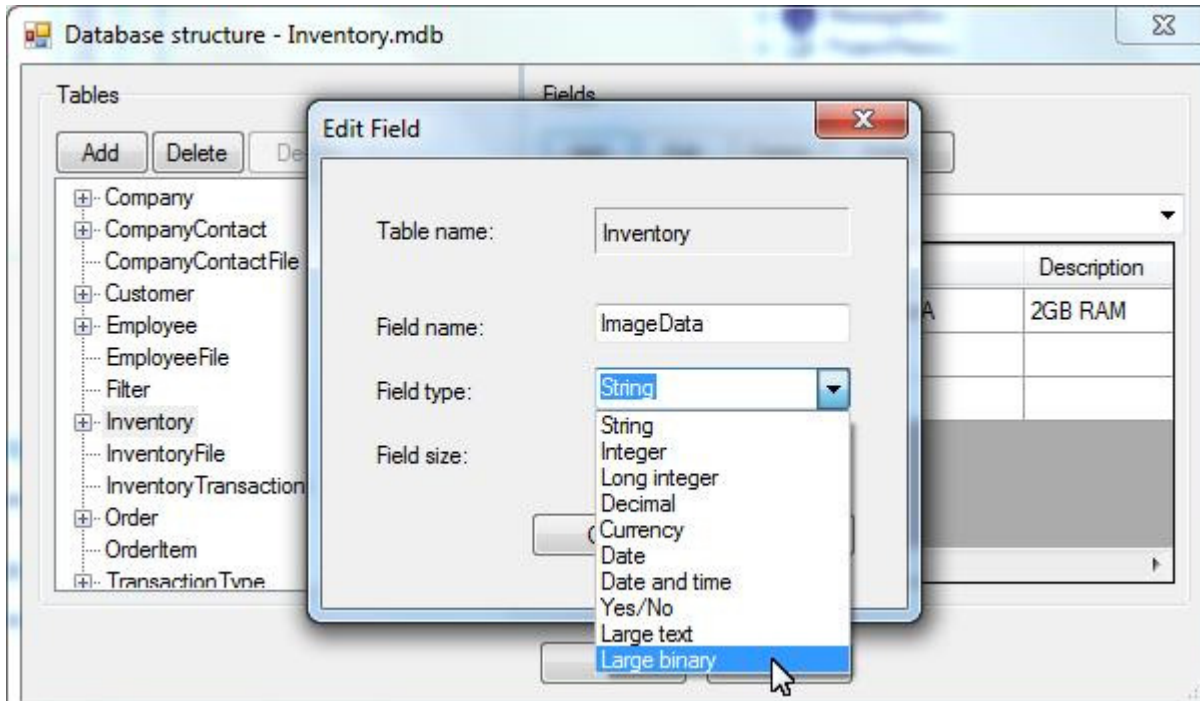
Save images in database.....	1
Data Binding.....	2
Update images	3
Create method to select image into the Picture Box	3
Execute SelectMethod when the Picture Box is clicked	13
Test	15

This document is for image data binding in Windows standalone applications. For image data binding in web applications, see <http://www.limnor.com/support/WebImageDataBinding.pdf>

Save images in database

An image needs large amount of storage space. Only binary variable length fields may hold images.

For Microsoft Access database, an OLE Object field should be used to save images. If you use Limnor Studio to create Access database then choose "Large binary" for a field for saving images:



For MySQL database, use LONGBLOB fields.

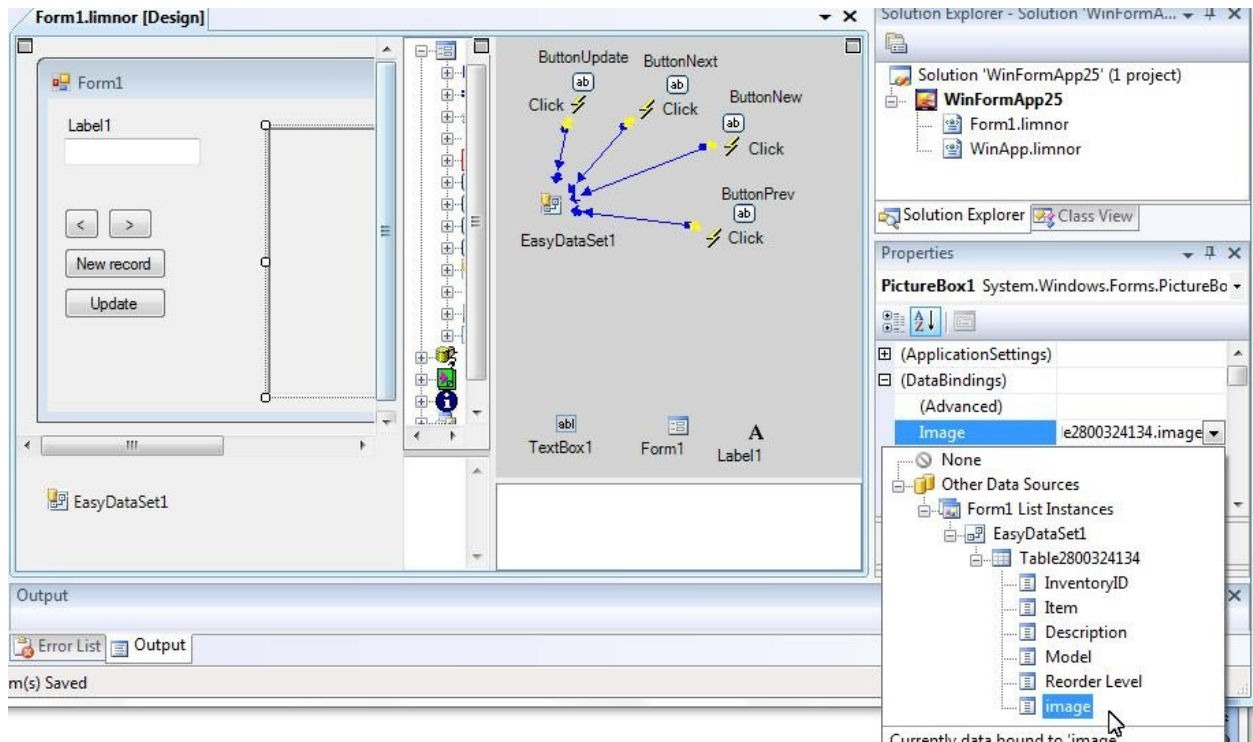
For other databases, try use LONGVARBINARY fields.

Data Binding

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

In this sample, we use an EasyDataSet to fetch data from a database; a Label and, a TextBox and a PictureBox are used to bind to 3 database fields respectively.

Here we just need to pay attention to the PictureBox data-binding. It is bound to the Image property of the PictureBox:



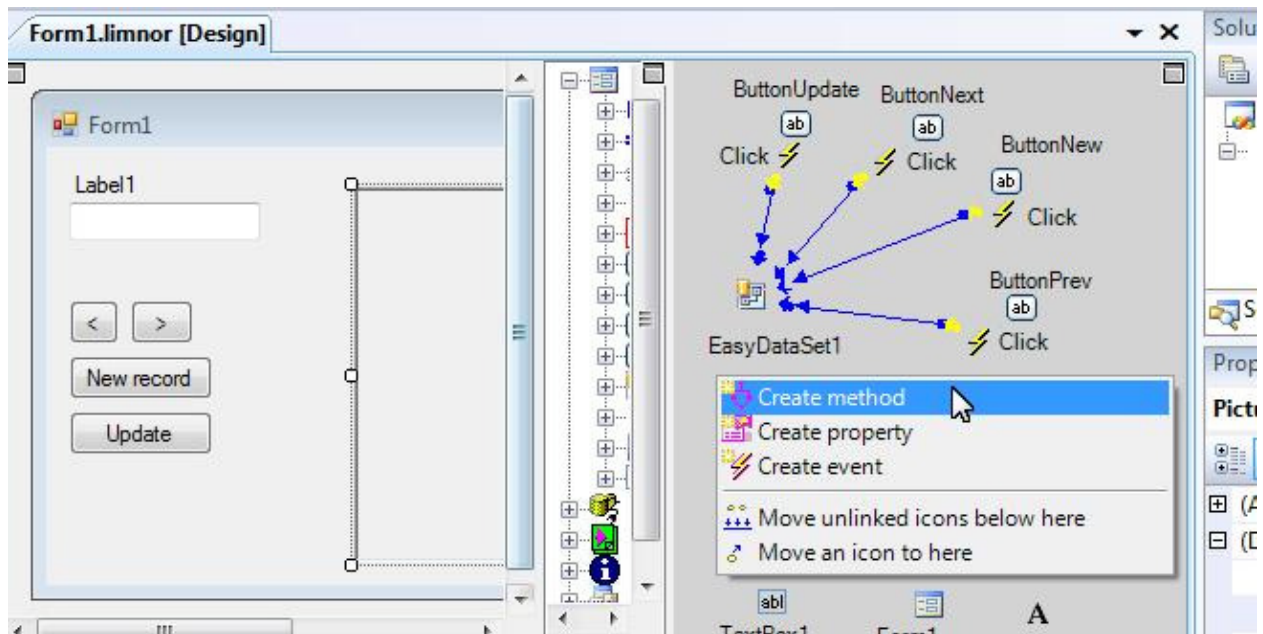
Update images

At runtime, your application may allow the user to select images from disks and save the images to databases.

In this sample, we create a SelectImage method. It uses the standard file selection dialogue box to select an image file. The image of the file is assigned to the Image property of the PictureBox. Since the Image property is bound to the database, when Update action of the EasyDataSet is executed, the image data is saved into the database.

Create method to select image into the Picture Box

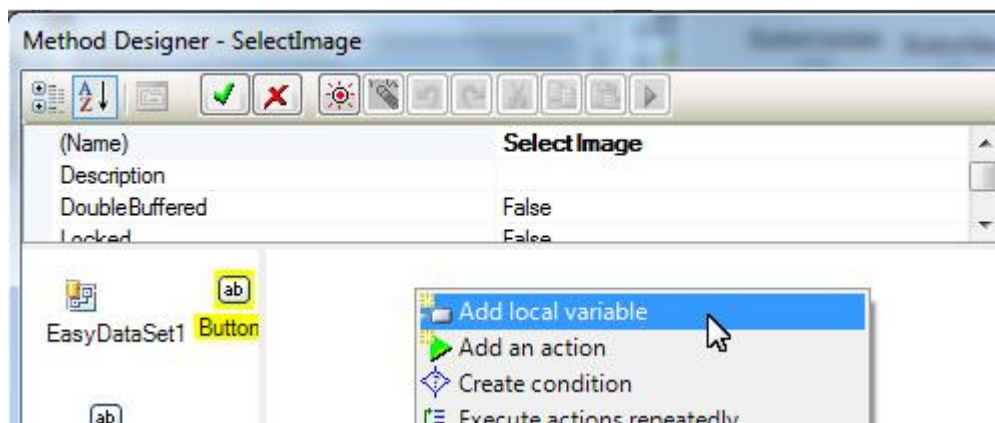
Right-click the Event Path; choose "Create method":



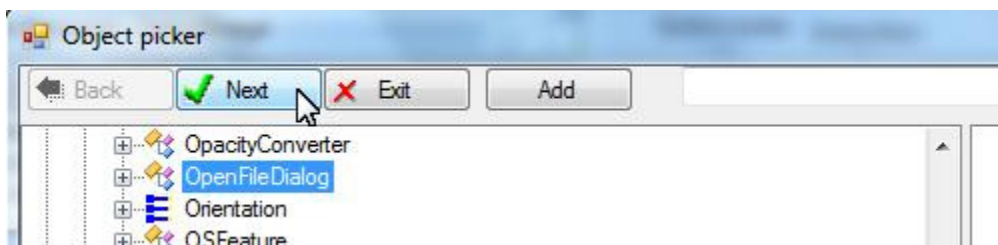
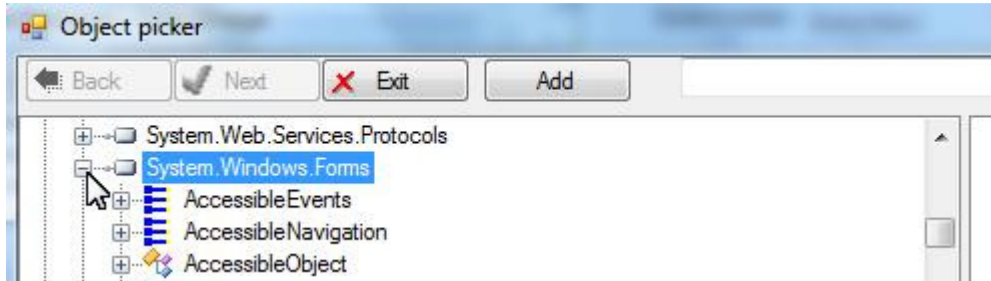
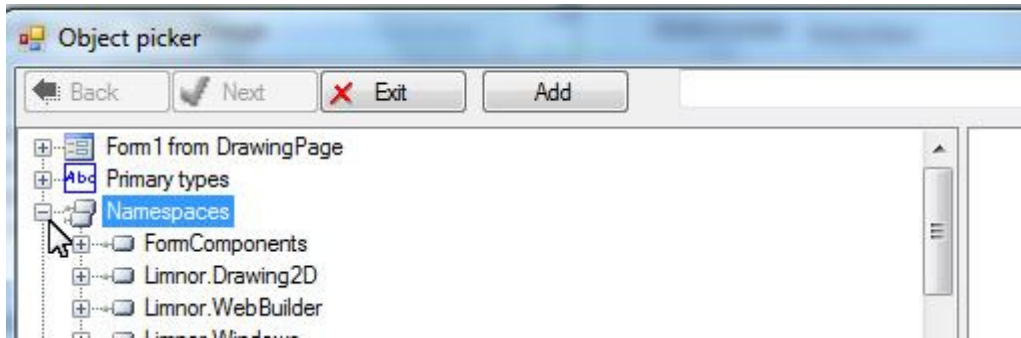
Rename the method to SelectImage:



To use the "open file" dialogue box, create a new local variable for the dialogue box:



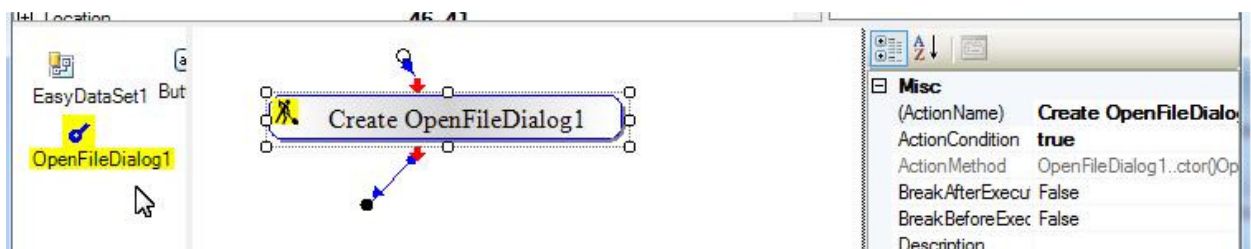
Select the OpenFileDialog class under System.Windows.Forms namespace:



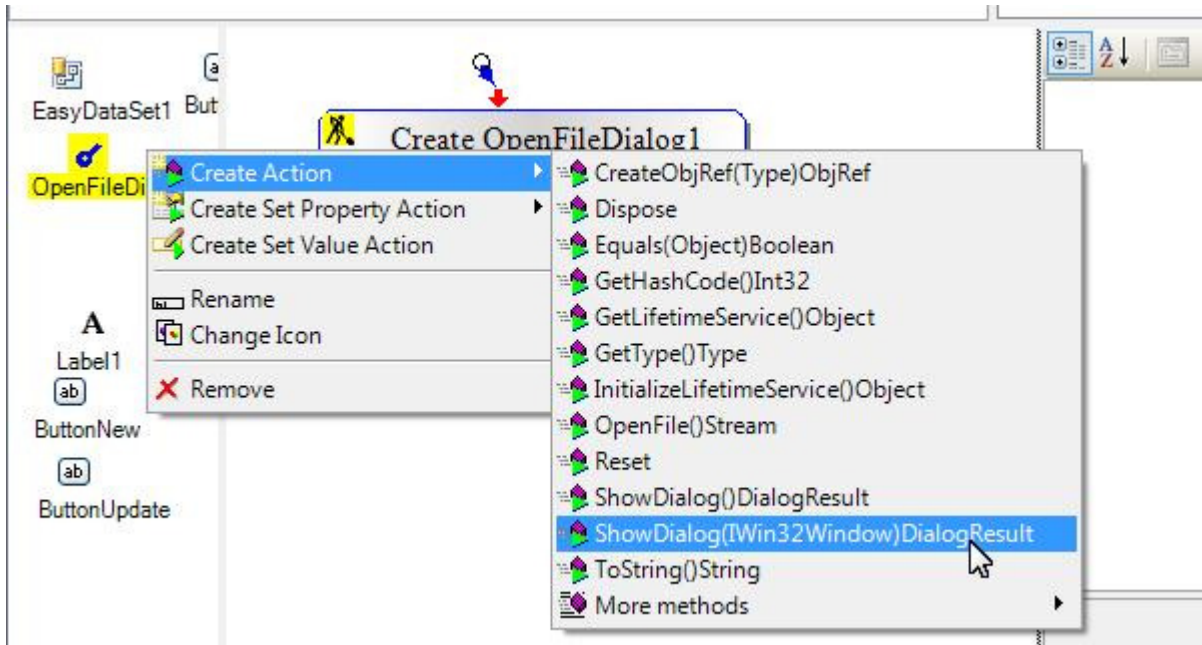
Select the constructor because we want to create an instance from the class:



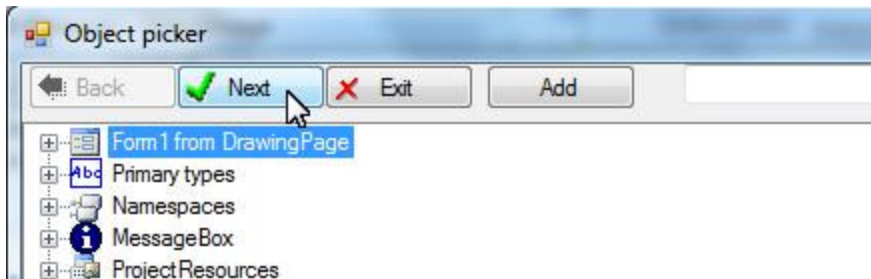
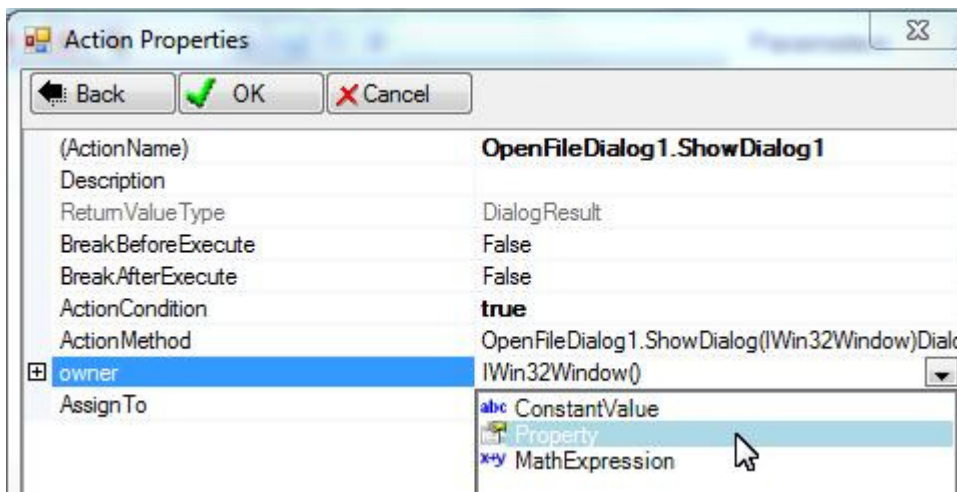
An action is created to create the dialogue box. The Dialogue instance appears in the Variable Pane:



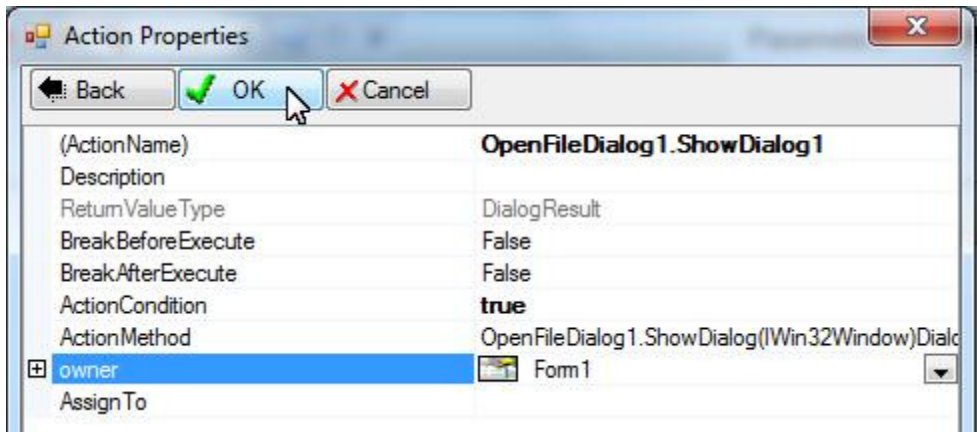
To display the dialogue box, we need to create a ShowDialog action. Right-click the dialogue variable; choose "Create Action"; choose "ShowDialog" method:



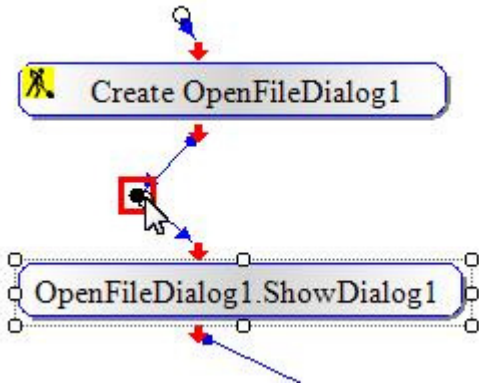
We may set the dialogue box owner to be the current form:



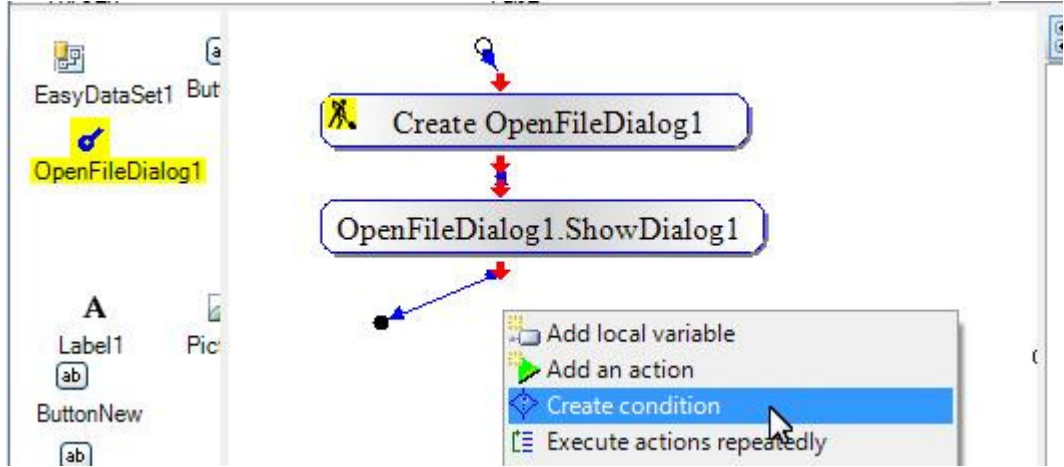
Click OK. The action appears in the Action Pane.



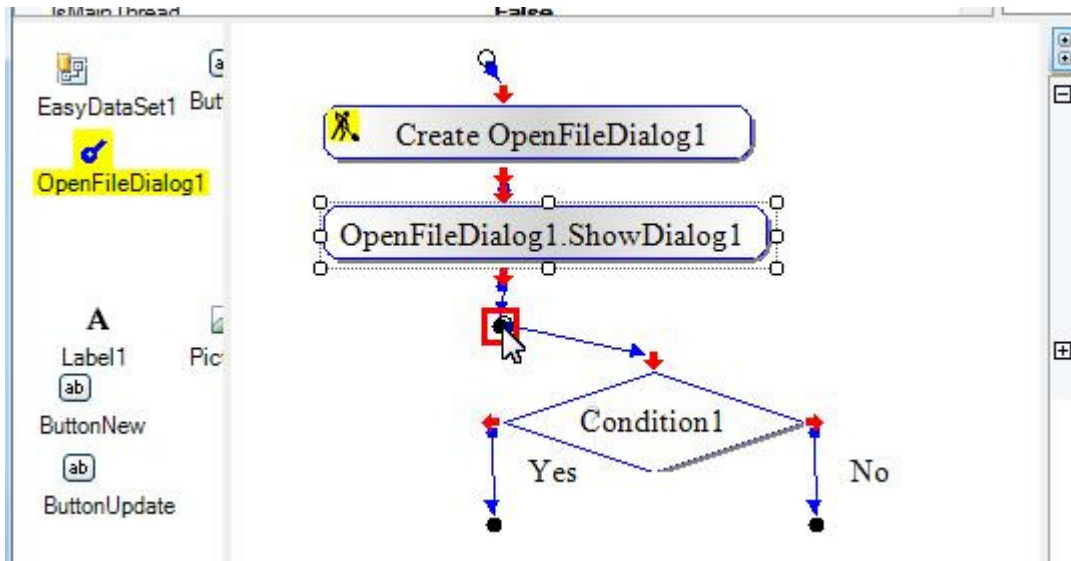
Link it to the existing action:



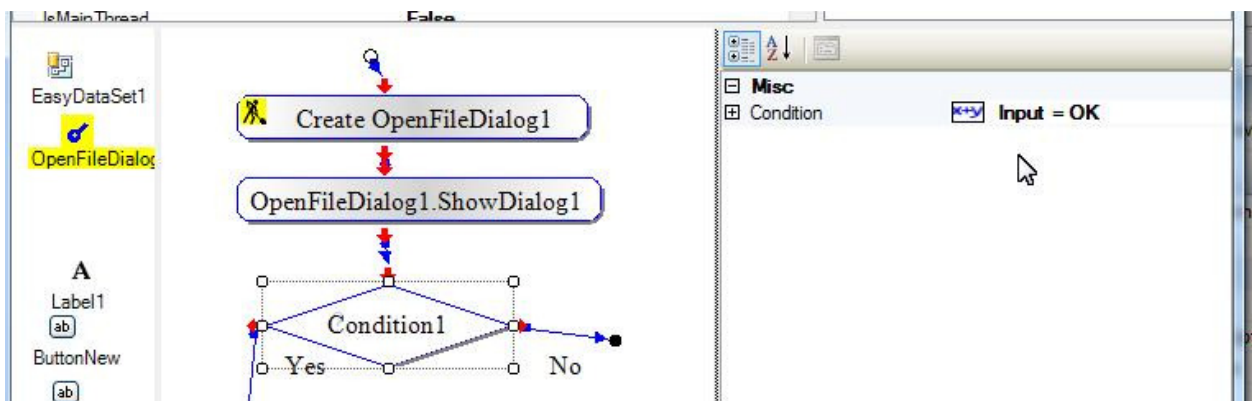
The ShowDialog action returns a value indicating how the dialogue box is closed: a file is selected or the dialogue box is cancelled. We may use a Condition action to accept the return value:



Link the in-port of the Condition action to the out-port of the ShowDialog action:



By doing this link, the return value of the ShowDialog action is passed to the Condition:



“Input = OK” means that the dialog is closed by selecting a file. If this is true then the actions linked to Yes-port of the Condition will be executed; otherwise the actions linked to the No-port of the Condition will be executed.

In this sample, we create an action to load the selected file into the PictureBox and link the action to the Yes-port of the Condition.

Right-click the PictureBox icon; choose “Create Set Property Action”; choose “Image” property:

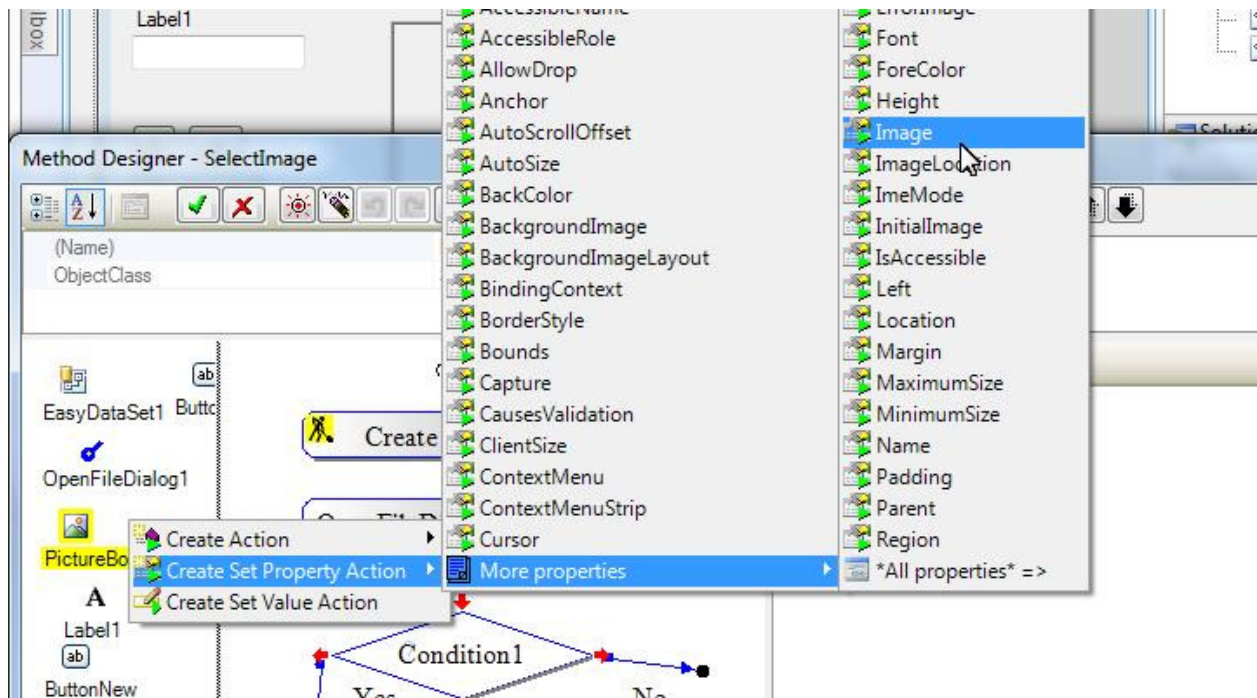
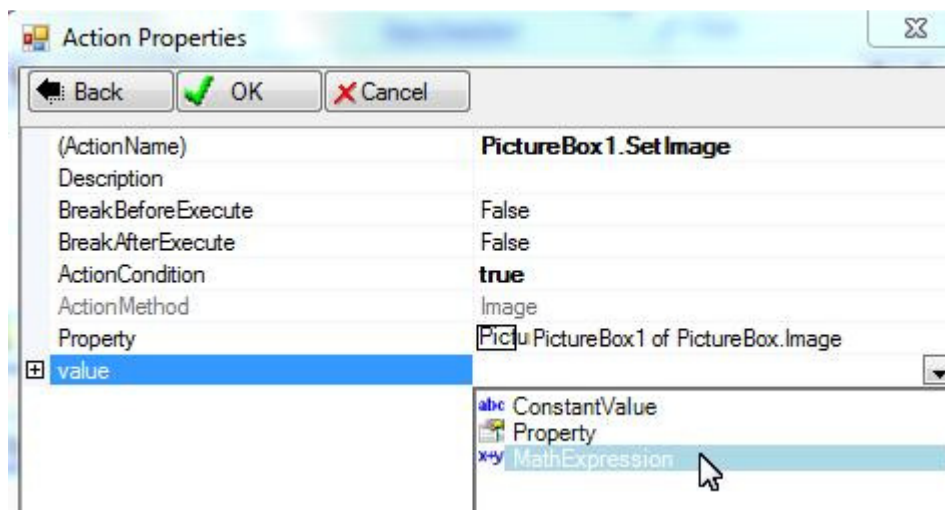
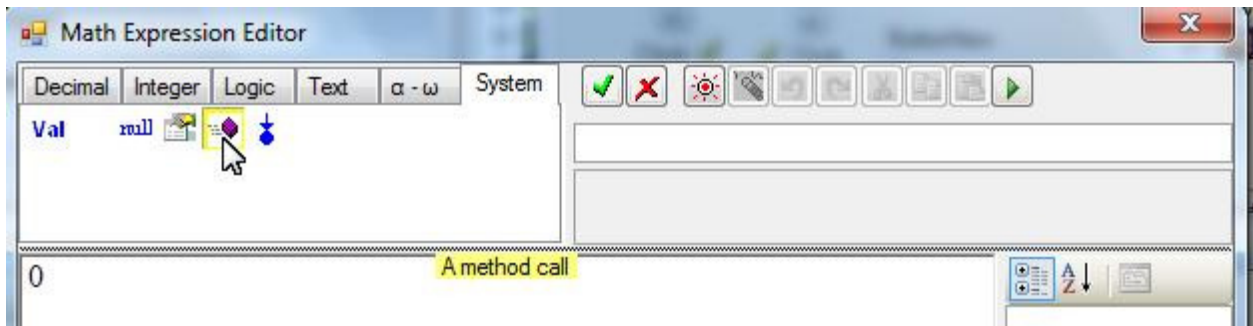


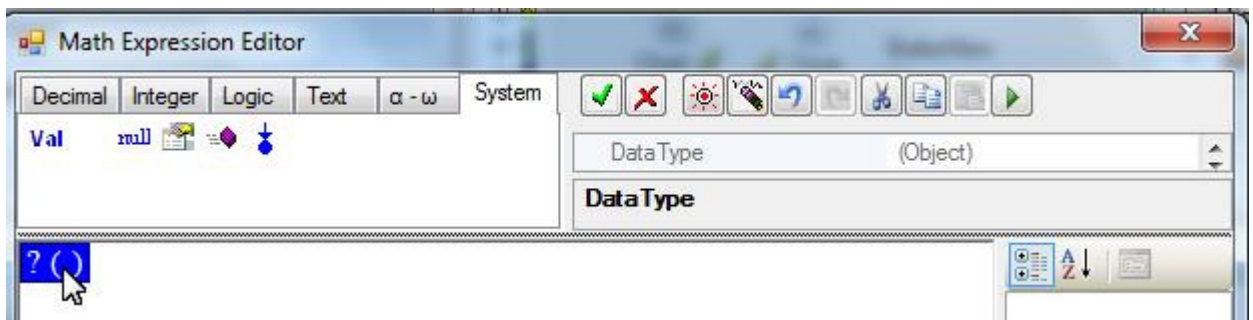
Image class in System.Drawing namespace has a FromFile method which can be used to load image from file. To call a method, choose “Math Expression”:



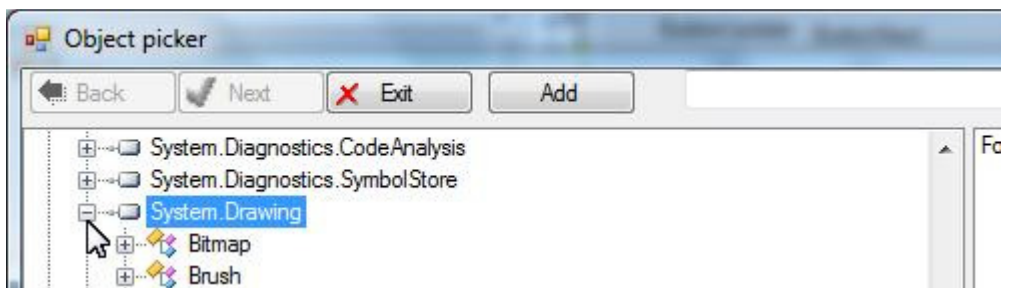
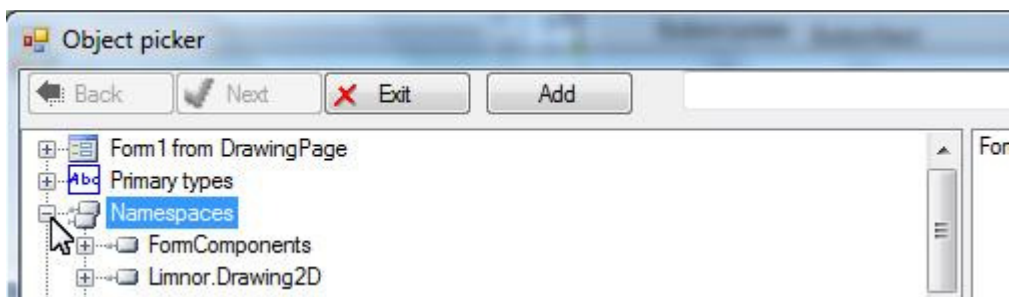
Select “method” icon:

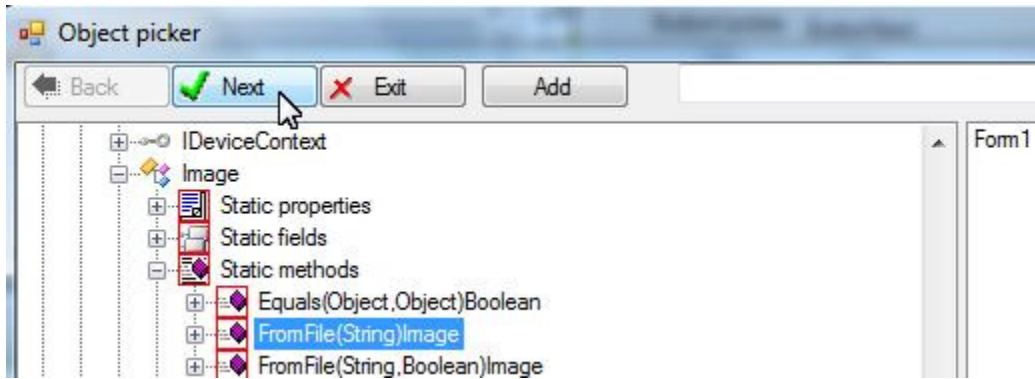


Double-click the method element:

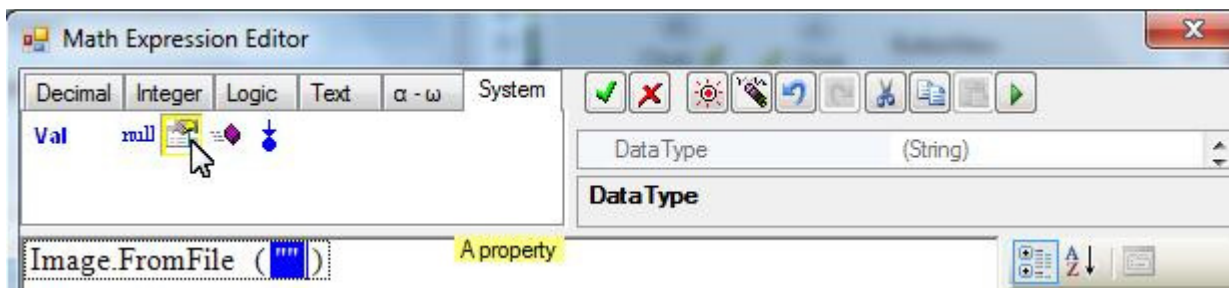


Select the FromFile method of the Image class:

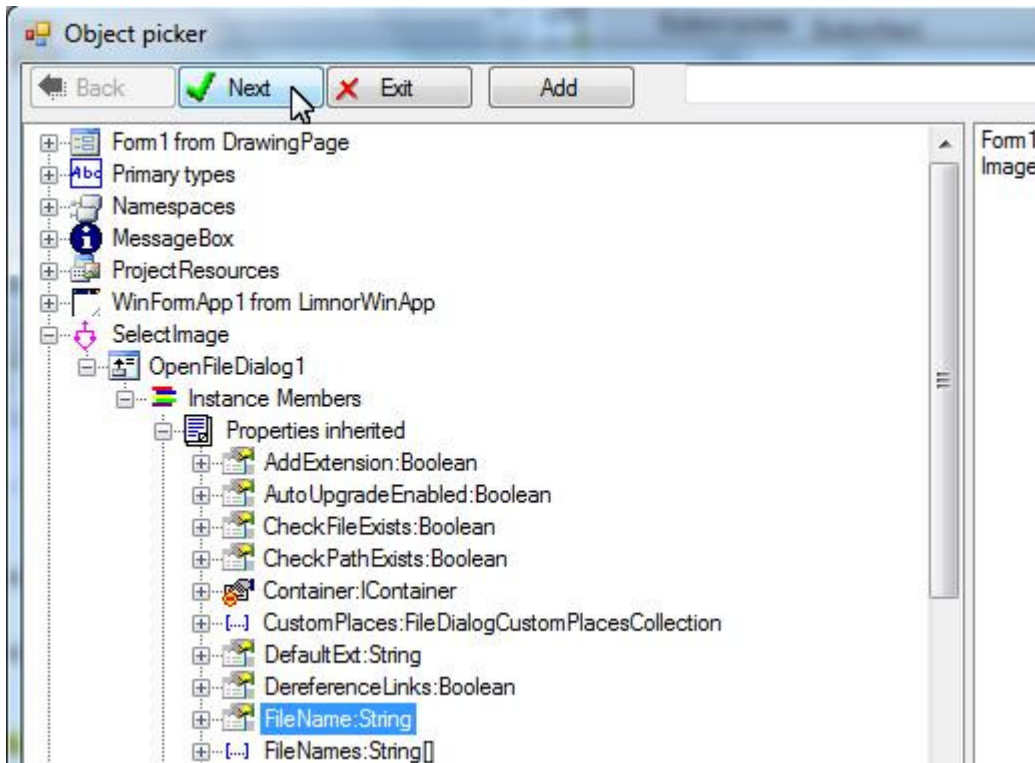




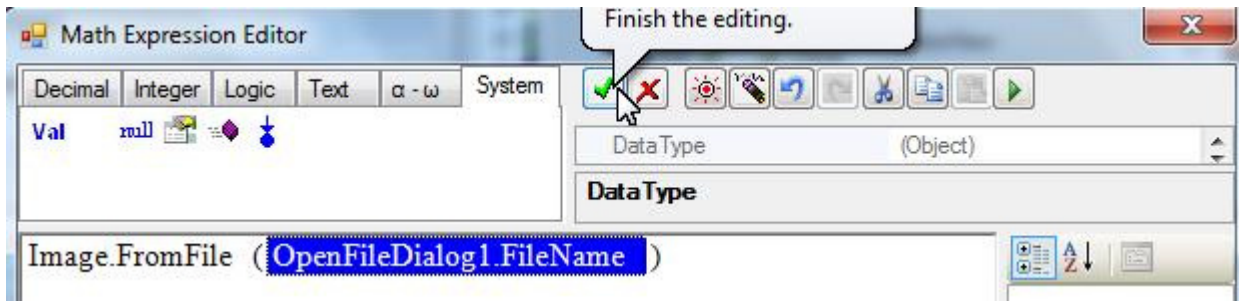
Select the parameter of the FromFile method; click Property icon because we want to use the selected file:



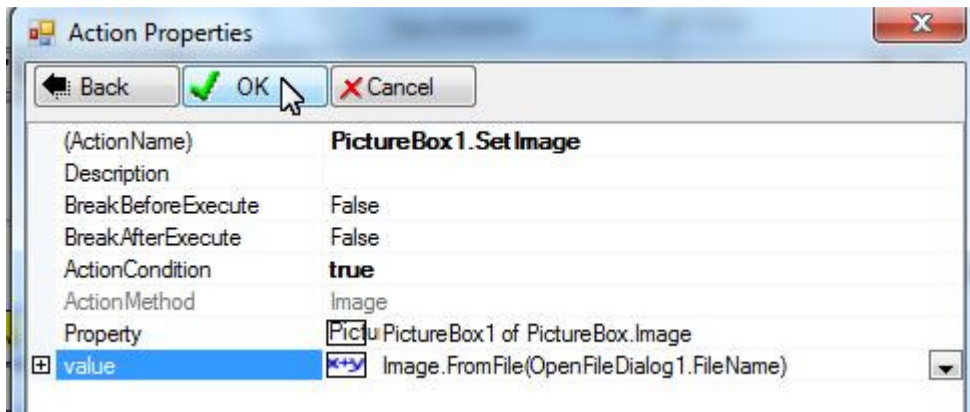
Select the Filename property of the dialog:



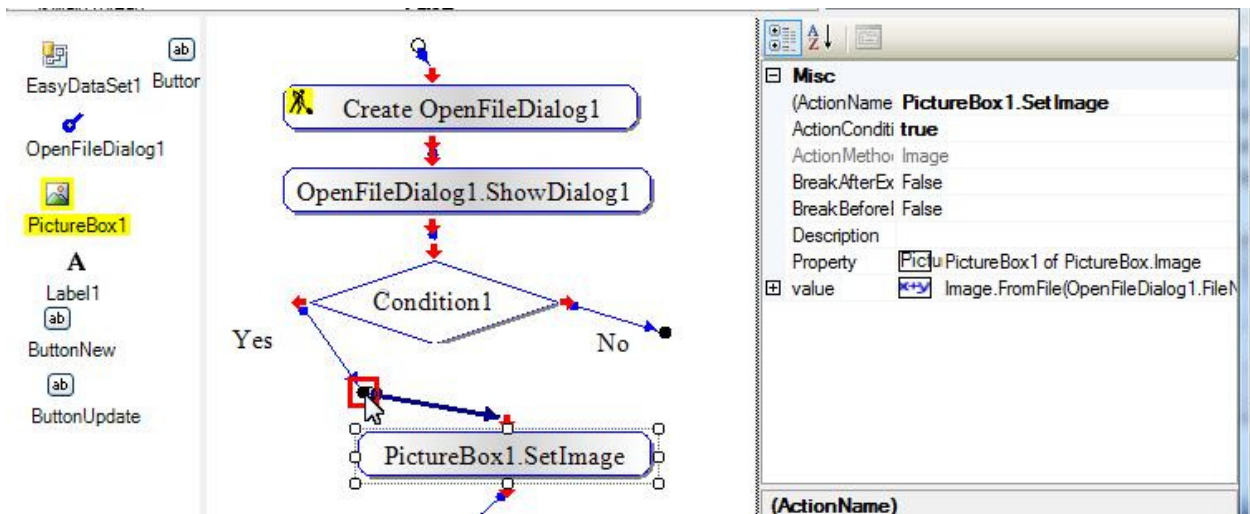
This expression loads the image from the file:



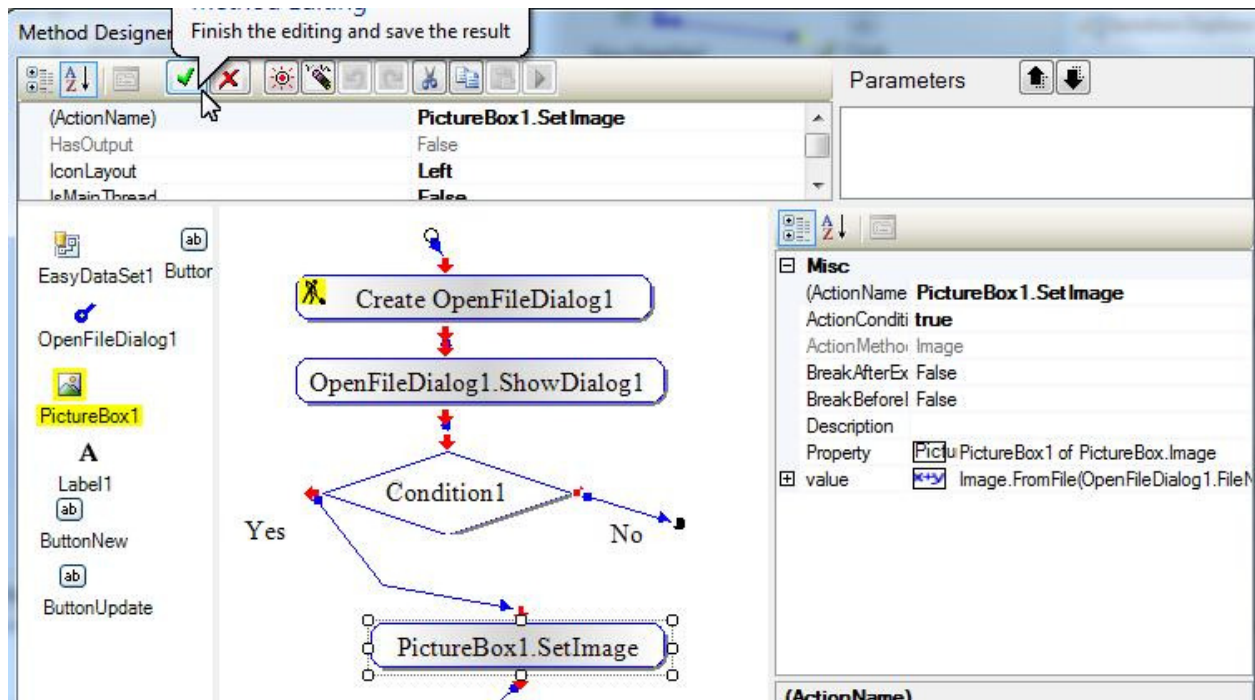
This action loads the image from selected file and assigns the image to the Image property of the Picture Box:



Link the action to the Yes-port of the Condition:

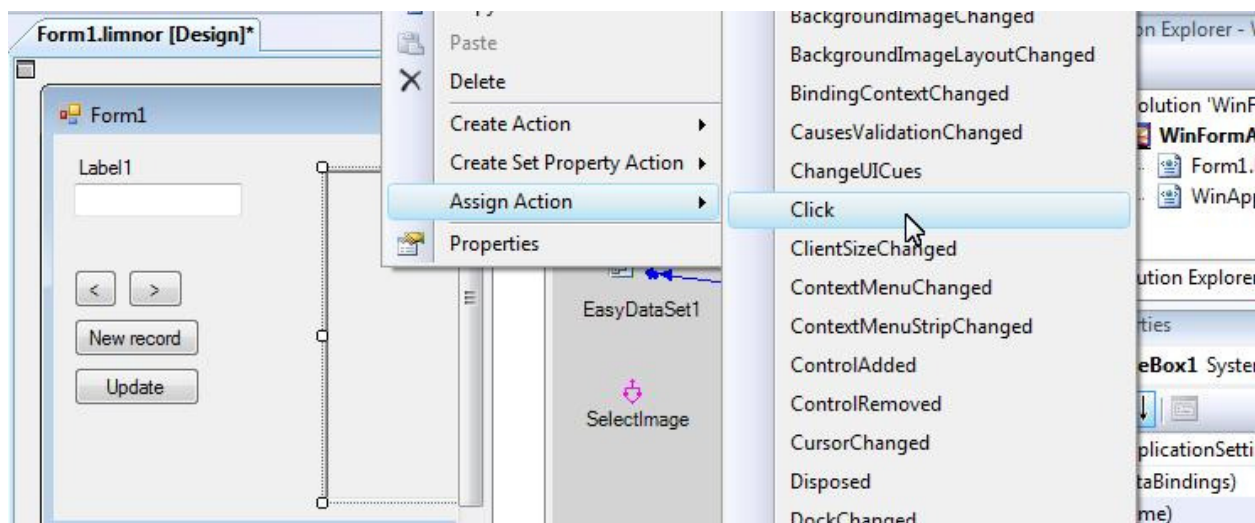


We are done creating this method.

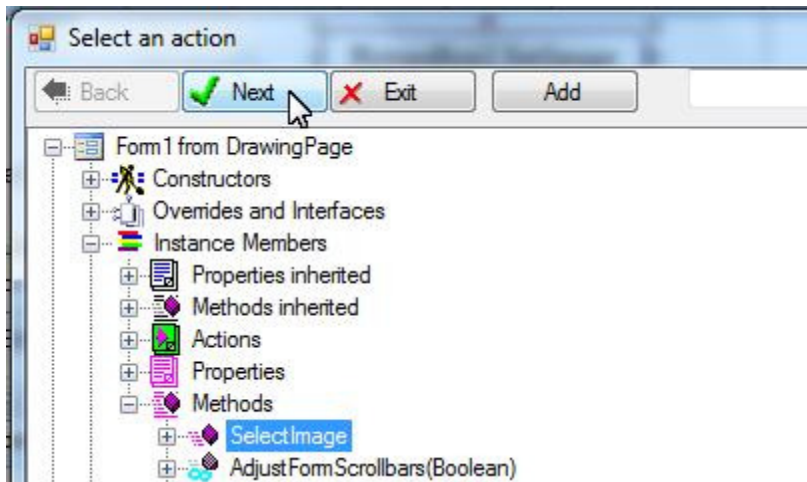


Execute SelectMethod when the Picture Box is clicked

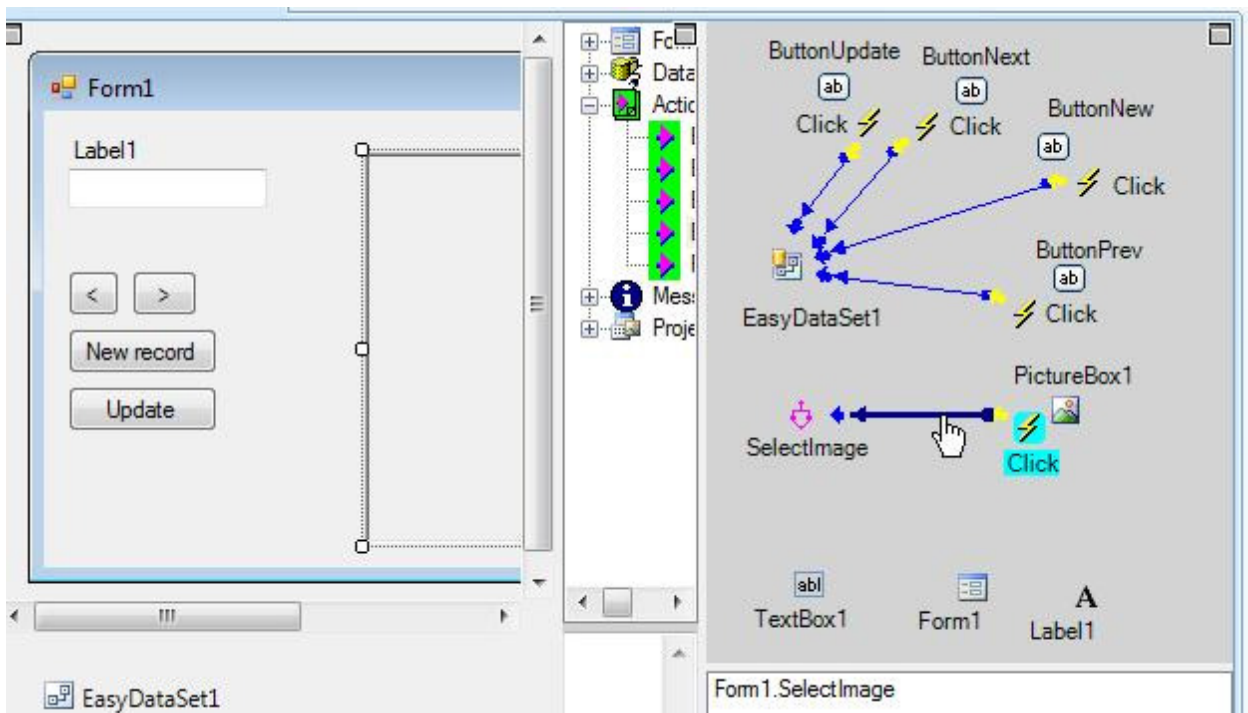
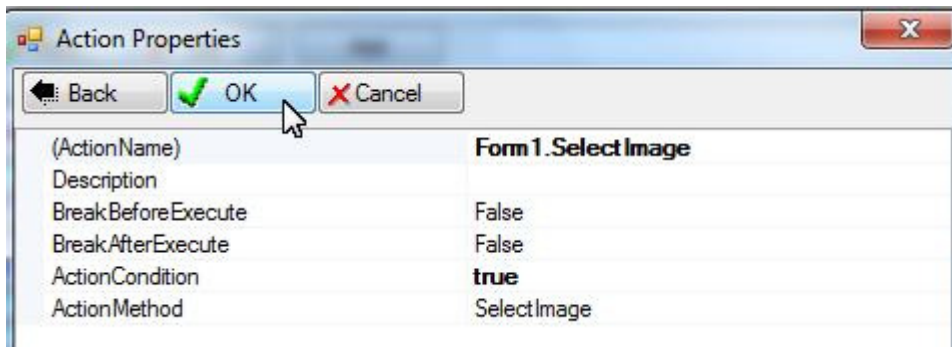
Right-click the Picture Box; choose “Assign Action”; choose “Click”:



Select method SelectImage and click “Next”:

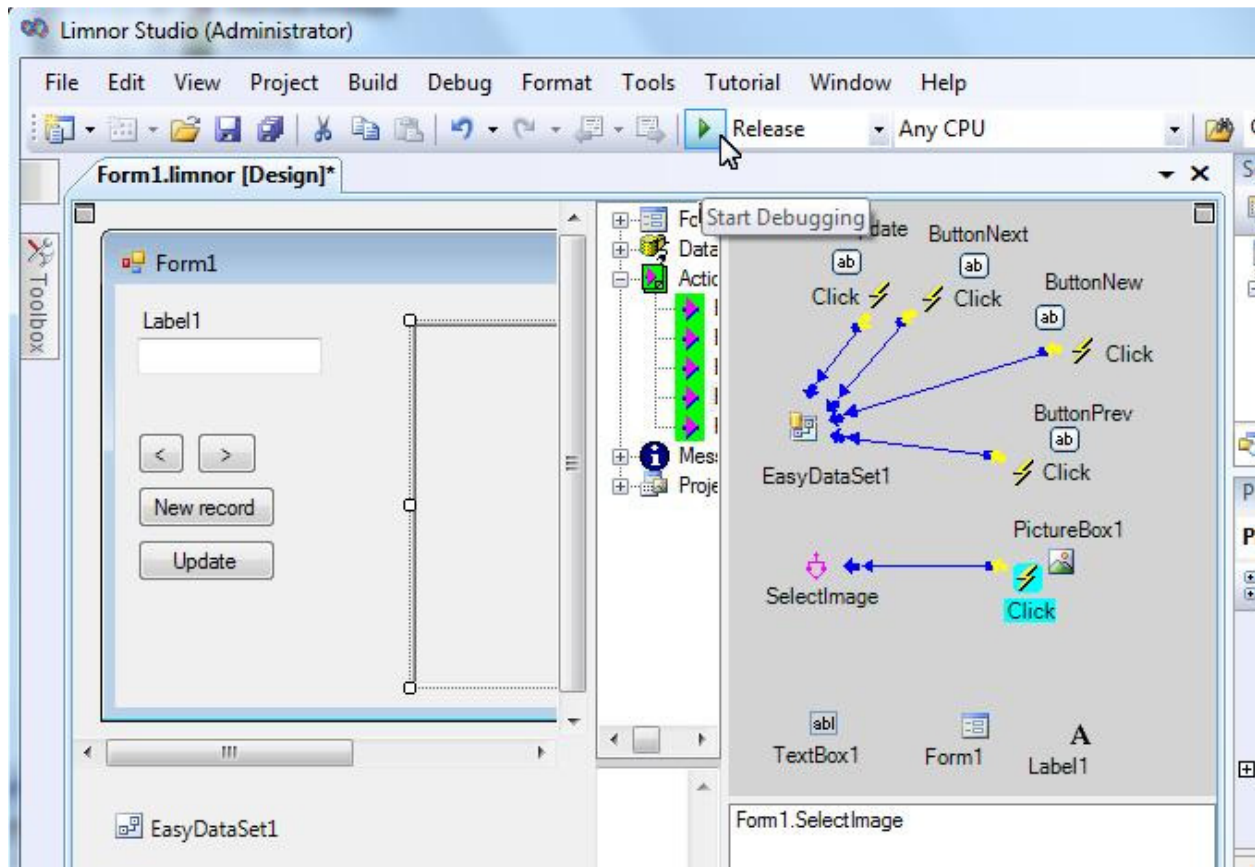


Click OK. A new action is created and assigned to the Click event of the Picture Box:

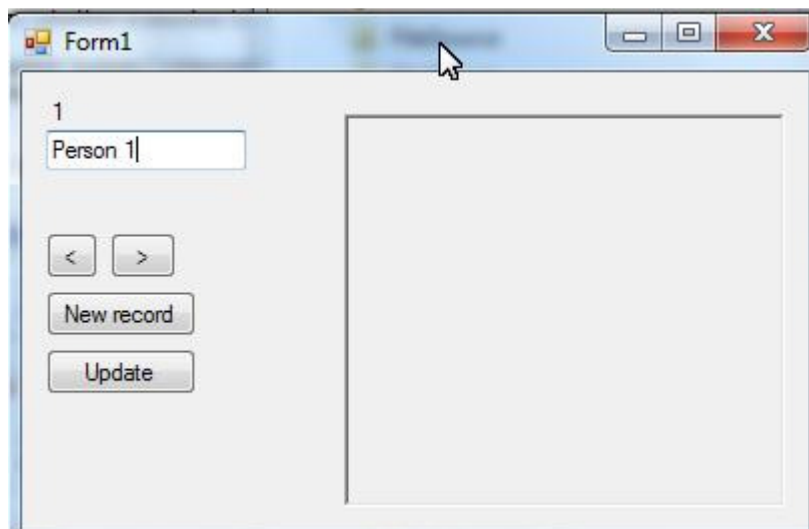


Test

We may test the sample now.



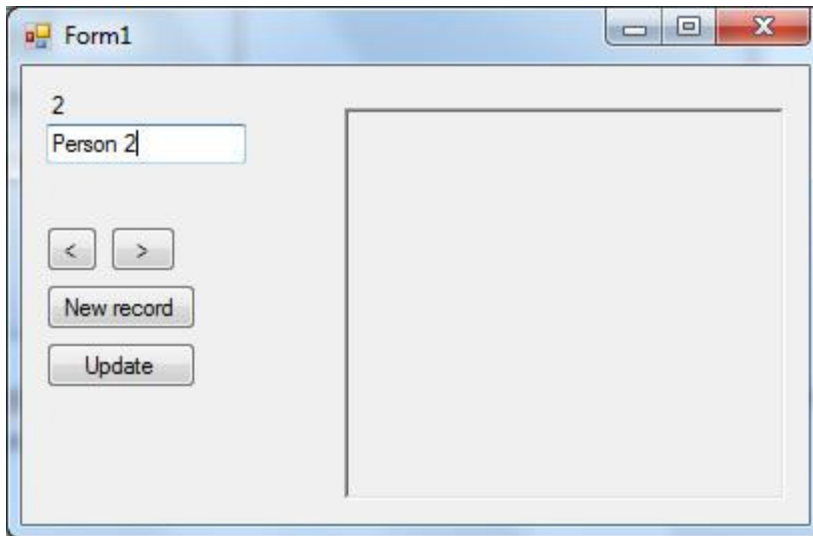
The form appears with the first record from the database displayed.



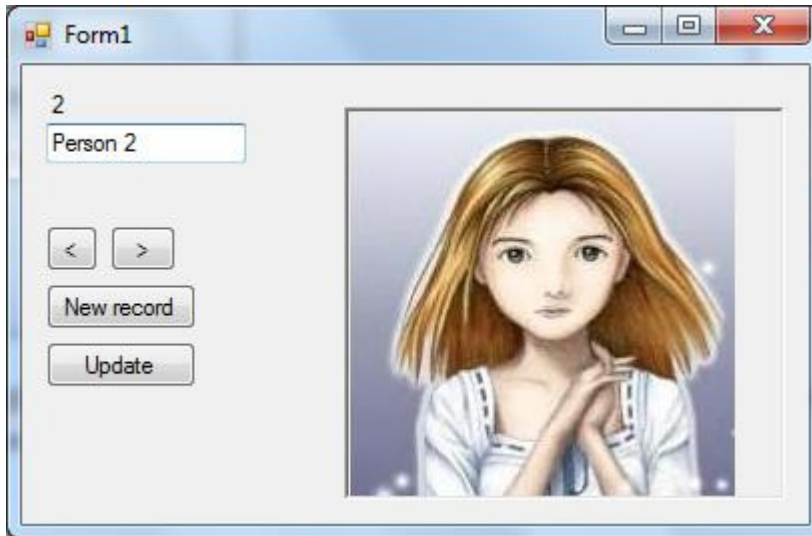
Click the PictureBox. A dialogue box appears for selecting a file. Select an image file. The image appears in the picture box:



Click ">" button to move to the next record:



Click the Picture Box to select another image file into it:



Click "<" and ">" buttons to navigate through the records. We can see that the selected images stay with the records. But note that at this time the images are still just in memory, not saved in the database. To save the data modifications to the database, click Update button to execute an Update action.

Binding Images in Web Applications

For doing image data binding in web applications, see