Use Object Database db4o

Contents

Introduction

The following information is from db4o web site: db4o is an open source object database that enables Java and .NET developers to store and retrieve any application object with only one line of code, eliminating the need to predefine or maintain a separate, rigid data model.

For more information on db4o, visit http://www.db4o.com/

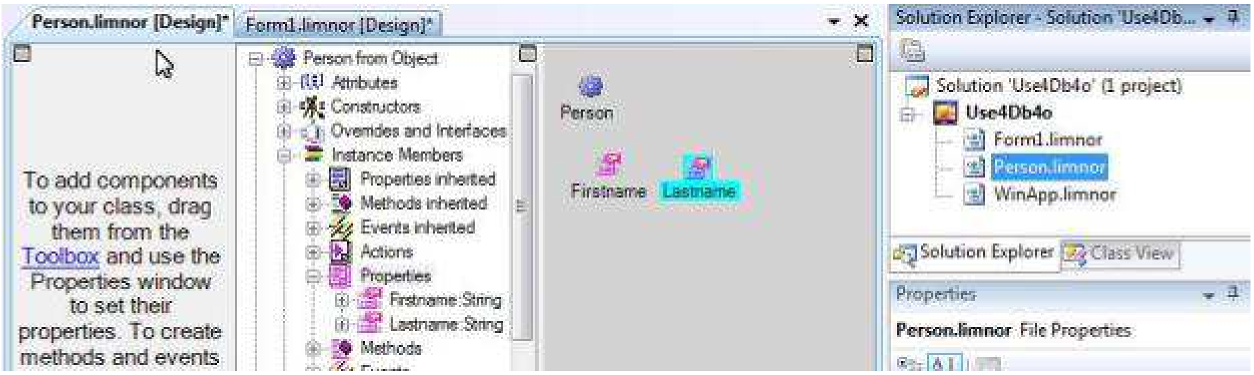
We will create a sample application to use db4o. The sample application demonstrates the using of 3rd party .Net libraries in a project.

We followed db4o document on http://refcardz.dzone.com/refcardz/getting-started-db4o and used SODA Queries approach described in the document to use the db4o.

Sample Project

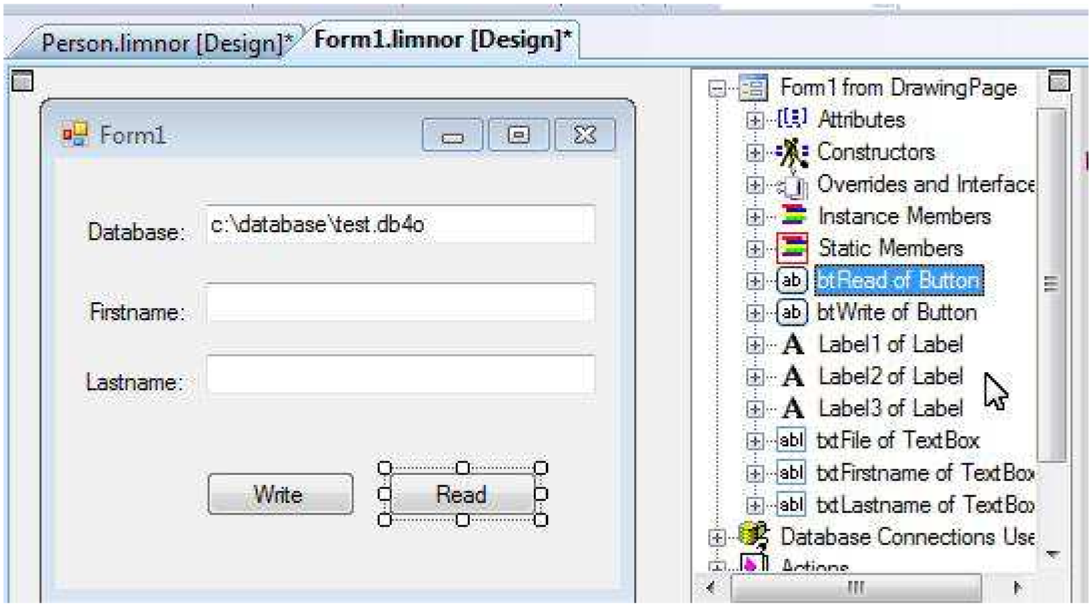
Windows Form project

A sample Windows Form project is created. A class named Person is created. Two properties, Firstname and Lastname, are added to the Person class. We will save a Person object to an Object database and then read it back from the database, using db4o.



To add components to your class, drag them from the Toolbox and use the Properties window to set their properties. To create methods and events

In the form, we use a text box to allow specifying a full file name and path for the database. Two buttons are used; one button for testing writing to the database; one button for testing reading from the database. Two text boxes for the Firstname and Lastname properties, respectively, of a Person object.

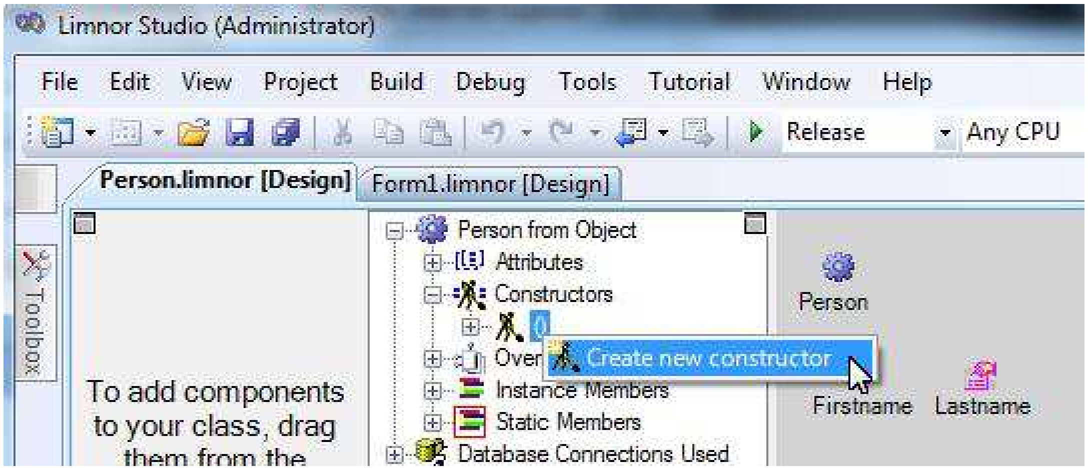


Create a constructor

We added two properties, Firstname and Lastname, to the Person class. For the creation of properties, see http://www.limnor.com/support/Limnor%20Studio%20-%20User%20Guide%20-%20Part%20IV.pdf.

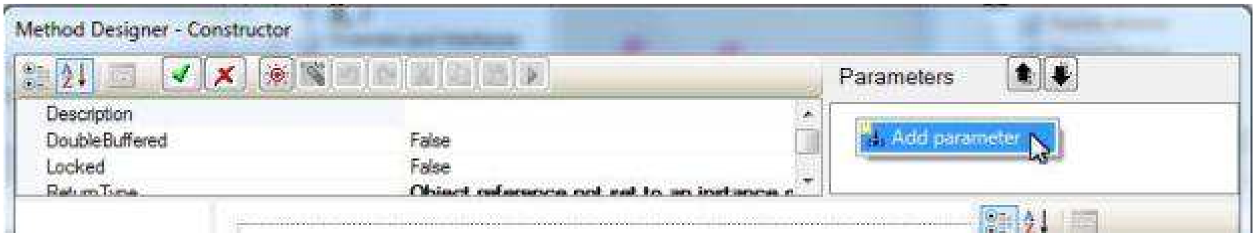
We now add a new constructor to the Person class for easier creation of Person instances. The new constructor will take parameters for setting properties Firstname and Lastname. Later we will see the use of this new constructor.

A new constructor must be derived from an existing constructor. Right-click the default constructor "()"; choose "Create new constructor":



The new constructor will be derived from the default constructor "()". The Method Editor is launched to edit the new constructor.

Right-click the Parameter Pane and choose "Add parameter":



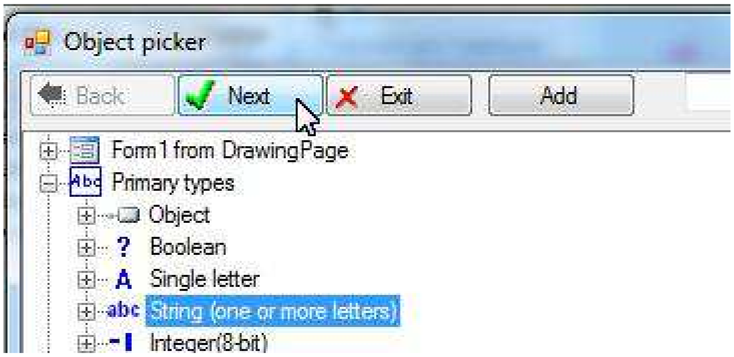
Description

DoubleBuffered False

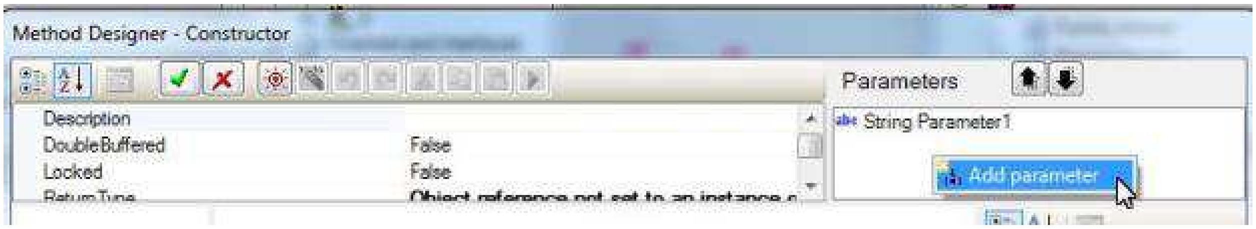
Locked False

ReturnType Object reference not set to an instance o

Select String as the parameter type:



Add another String parameter:



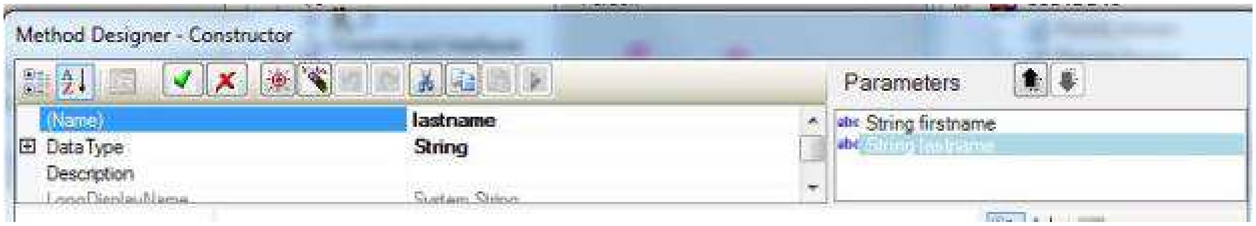
Description

DoubleBuffered False

Locked False

ReturnType Object reference not set to an instance o

Rename the parameters:



(Name) lastname

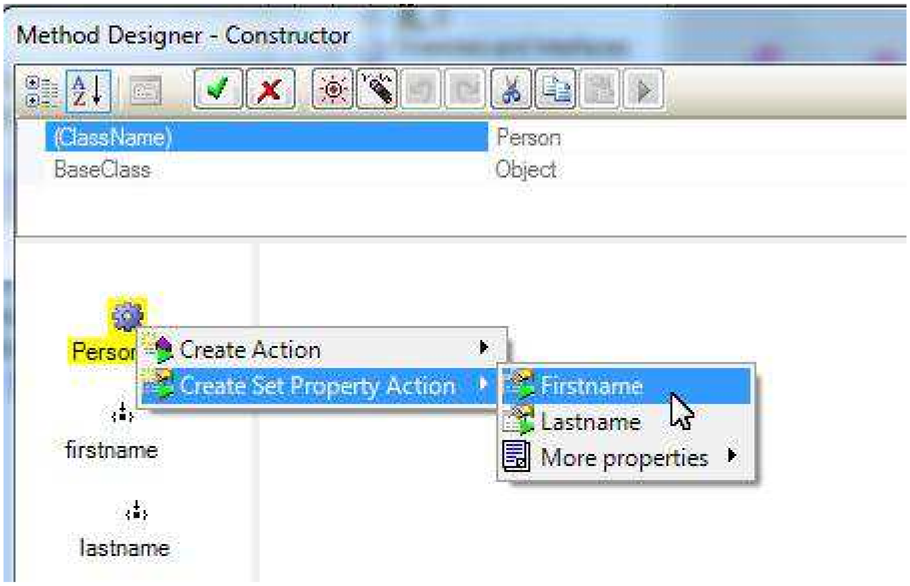
DataType String

Description

LongDisplayName System.String

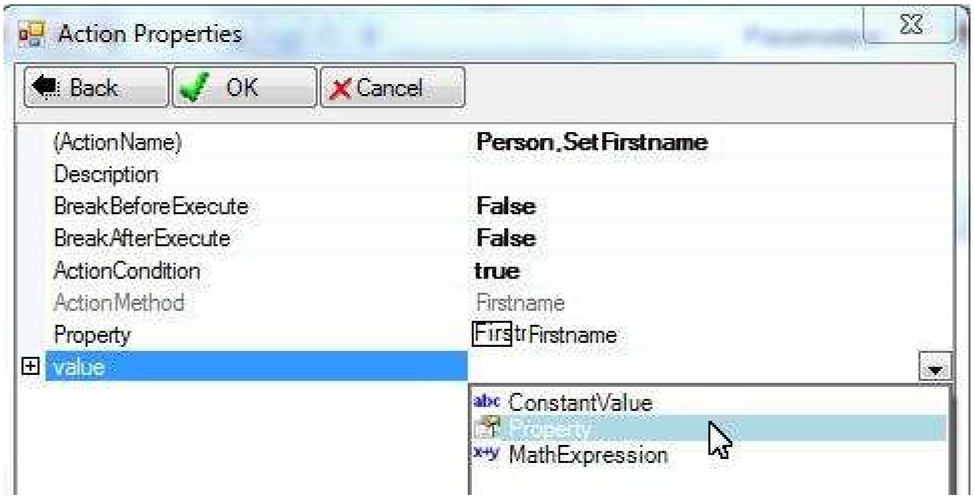
For this constructor, we simply pass the parameters to the properties.

Right-click the Person object; choose “Create Set Property Action”; choose Firstname:


Method Designer - Constructor

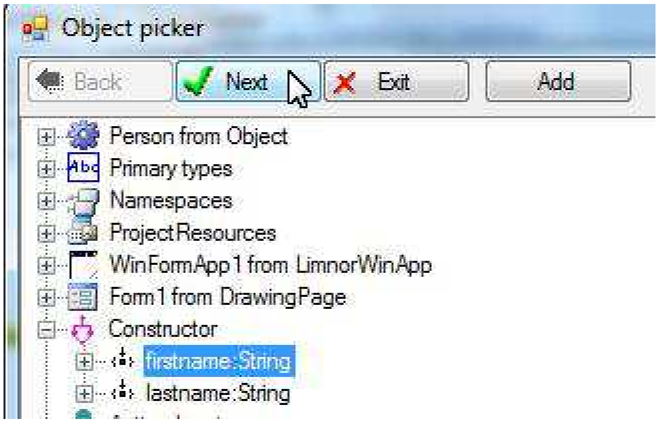
(ClassName)	Person
BaseClass	Object

For “value”, select “Property”:

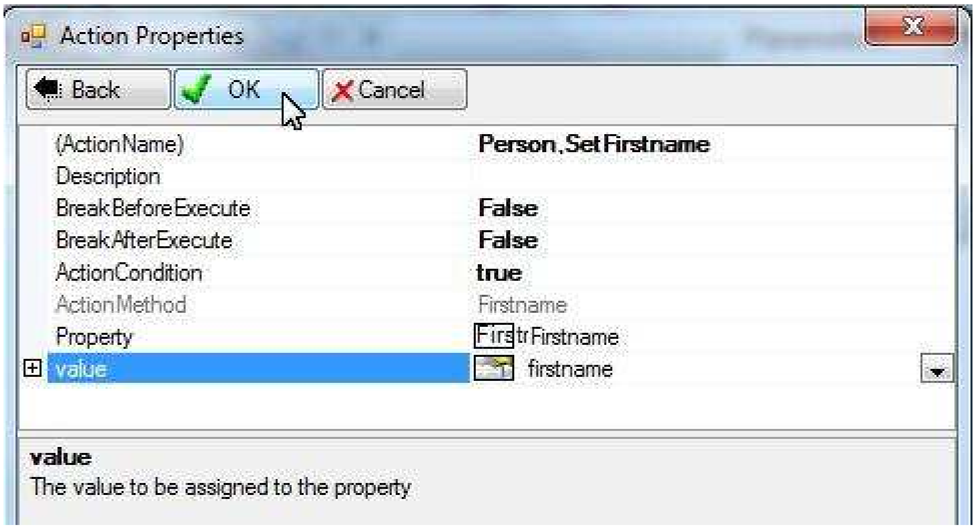

Action Properties

(ActionName)	Person.SetFirstname
Description	
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	Firstname
Property	FirstrFirstname
value	

Choose the parameter firstname:


Object picker

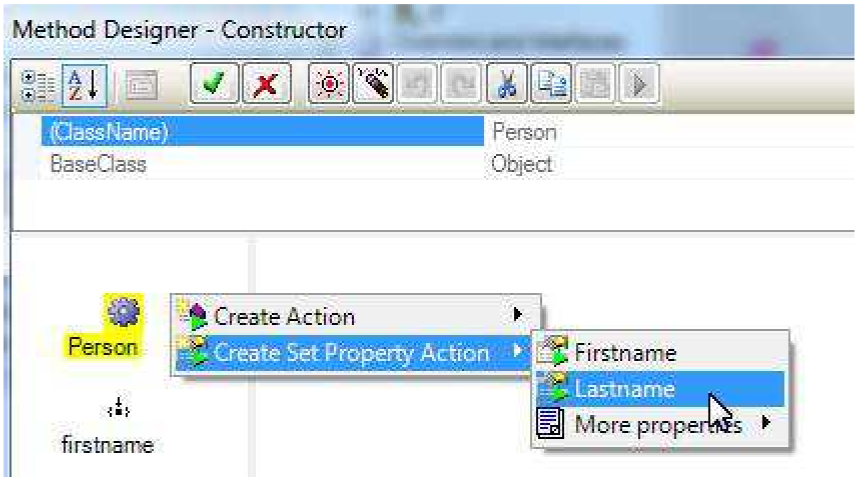
Click OK to finish creating this action:



(ActionName)	**Person.SetFirstname**
Description	
BreakBeforeExecute	**False**
BreakAfterExecute	**False**
ActionCondition	**true**
ActionMethod	Firstname
Property	Firstr Firstname
⊞ value	firstname

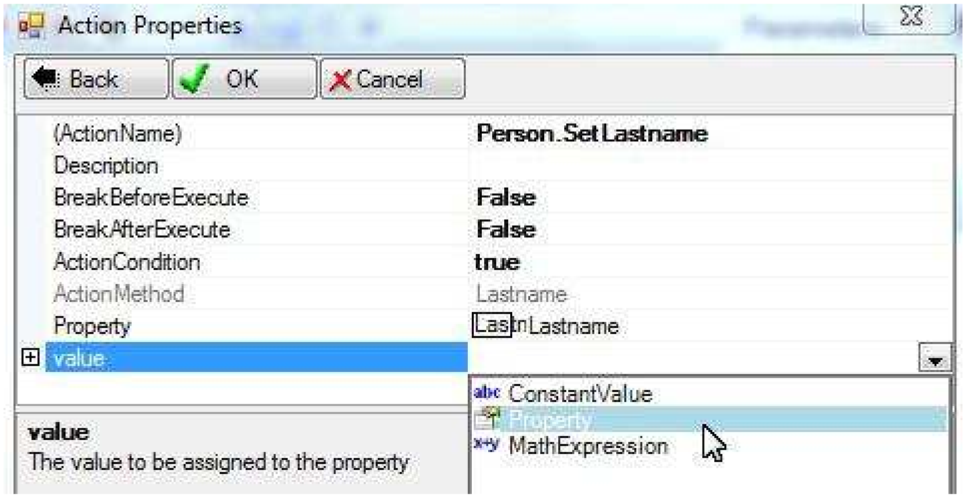
value
The value to be assigned to the property

Create another action. Right-click Person; choose "Create Set Property Action"; choose Lastname:



(ClassName)	Person
BaseClass	Object

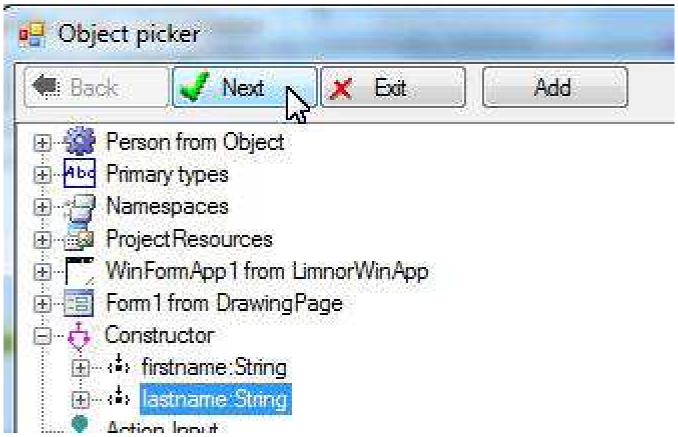
For "value", choose "Property":



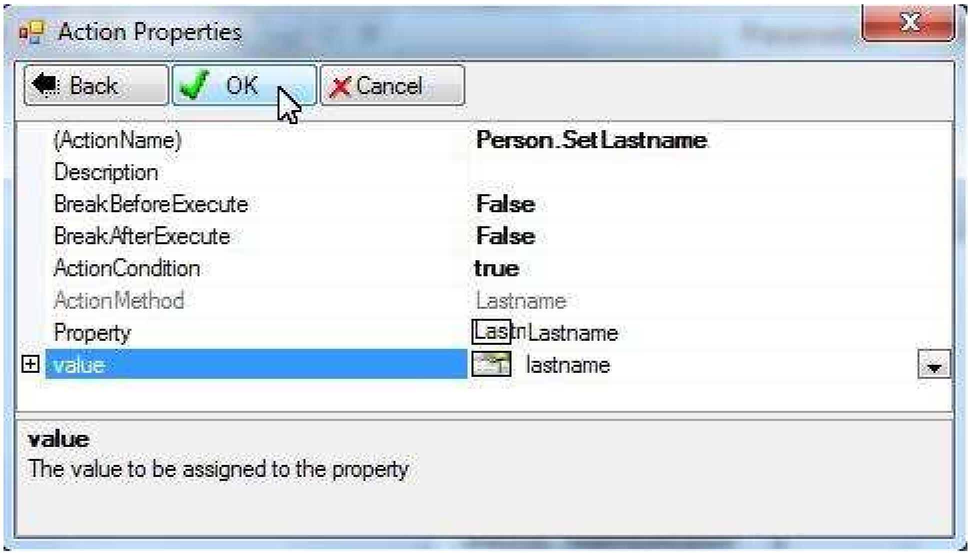
(ActionName)	Person.SetLastname
Description	
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	Lastname
Property	Lastn Lastname
⊞ value	

value
The value to be assigned to the property

Select parameter lastname:



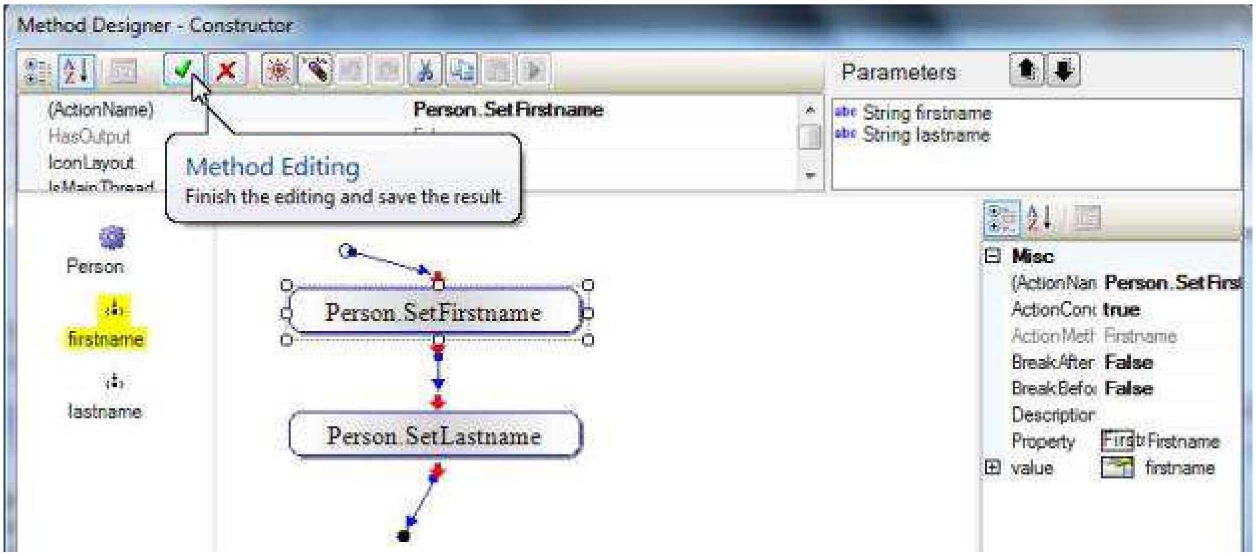
Click OK to finish creating this action:



(ActionName)	Person.SetLastname
Description	
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	Lastname
Property	Lastn Lastname
⊞ value	lastname

value
The value to be assigned to the property

Link the two actions together. This is our new constructor:

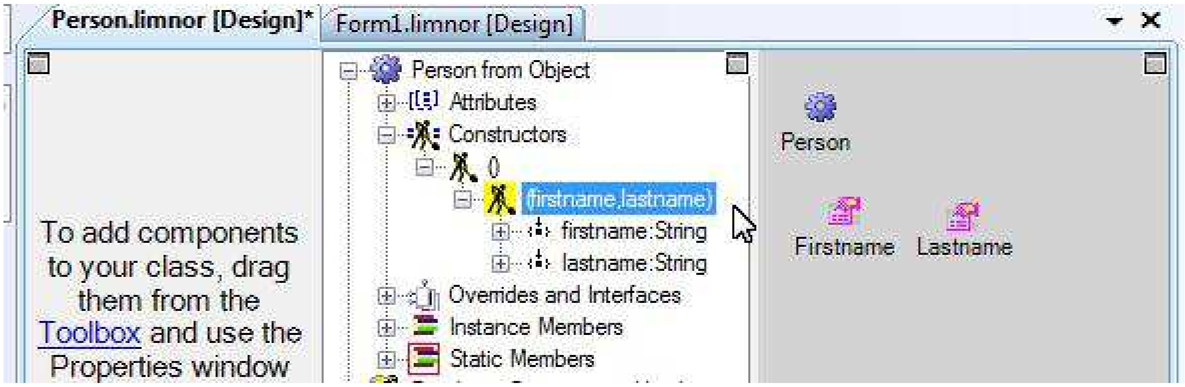


(ActionName) Person.SetFirstname
HasOutput
IconLayout
IsMainThread

String firstname
String lastname

(ActionNan Person.SetFirst
ActionConc true
ActionMeth Firstname
BreakAfter False
BreakBefo False
Description
Property Firstname
value firstname

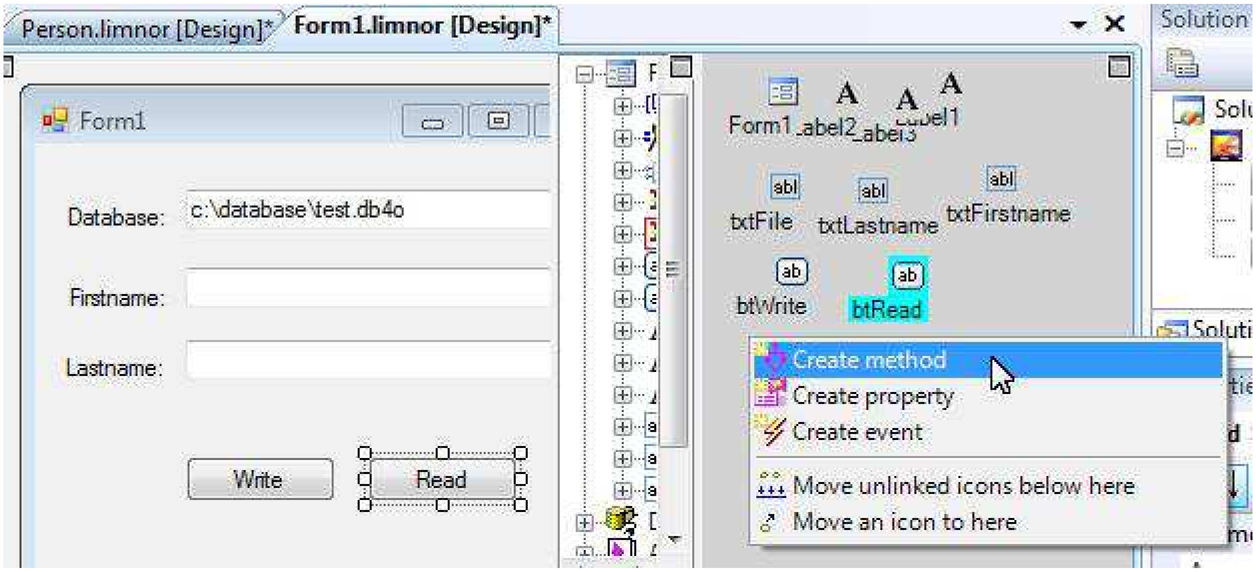
The new constructor appears under its base constructor:



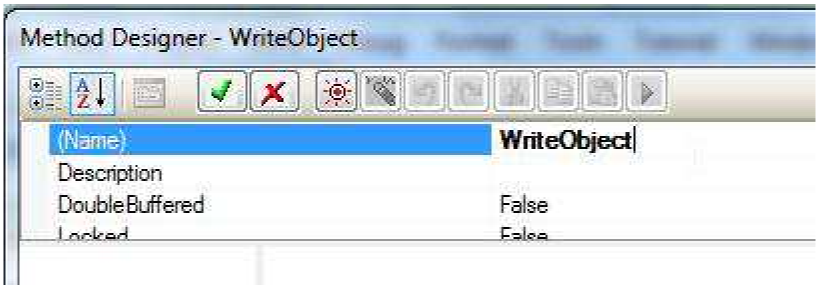
To add components to your class, drag them from the Toolbox and use the Properties window

Write Object to Database

We create a method to write a Person object to the database.



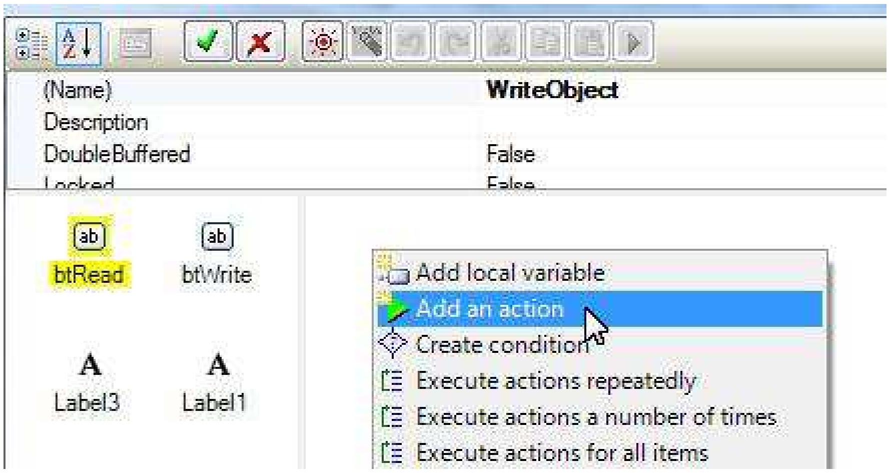
Rename the method to WriteObject:



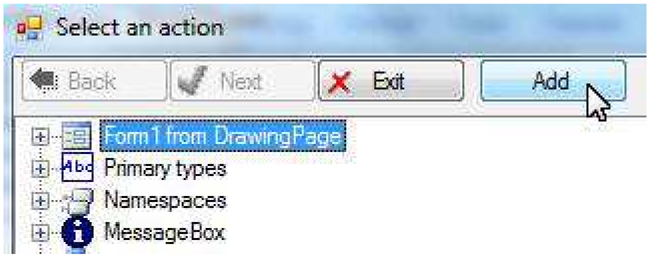
Open database

According to the db4o documentation, we need to use db4oFactory class to open a database, using OpenFile method.

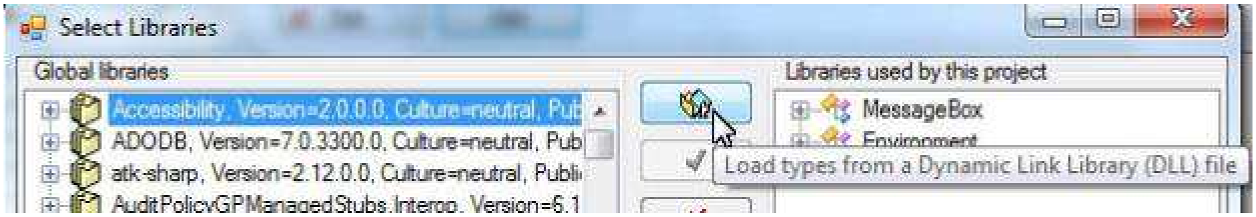
Right-click the Action Pane; choose “Add an action” to create an OpenFile action:



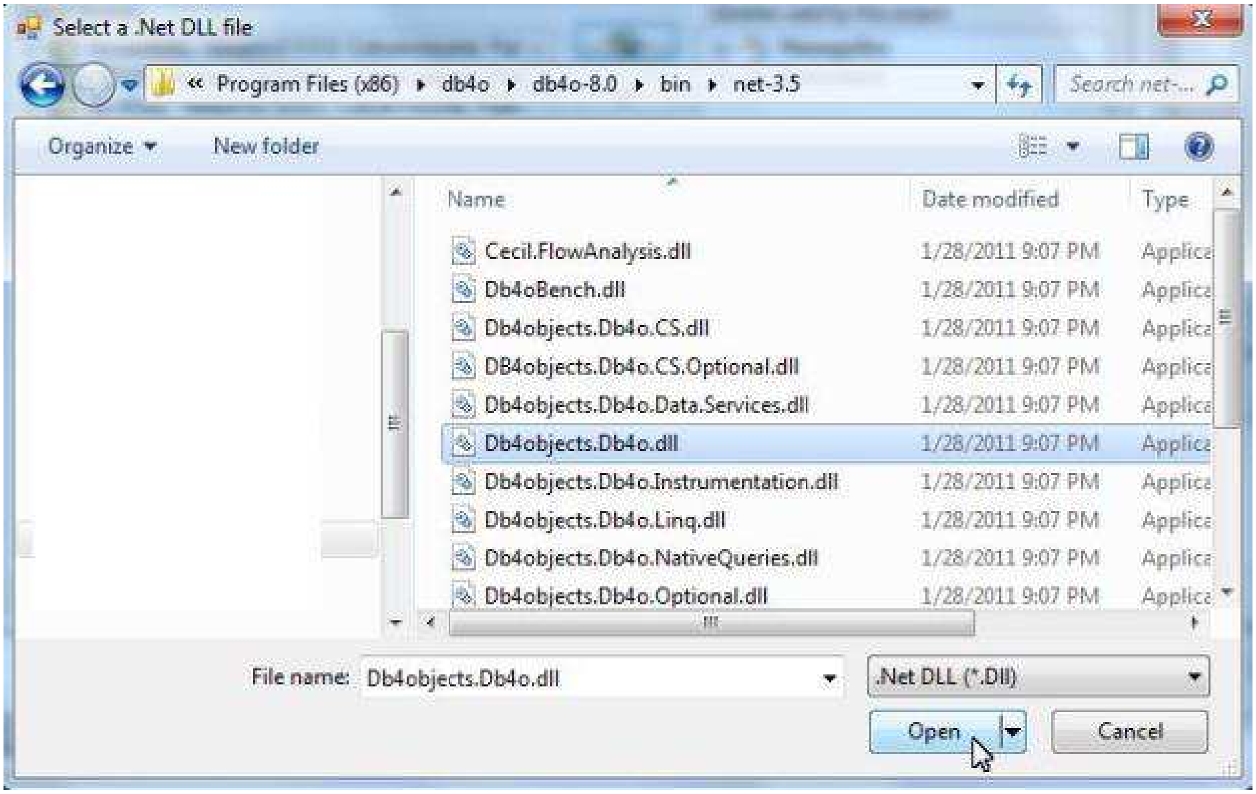
Because db4o is a 3rd party library, we need to add it to our project. Click Add button:



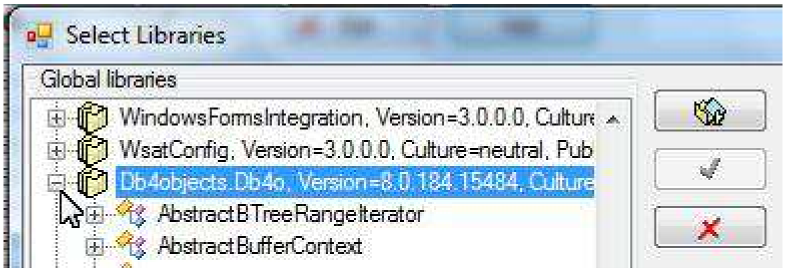
Click the load button to load the library:


Select Libraries
Global libraries
Accessibility, Version=2.0.0.0, Culture=neutral, Pub
ADODB, Version=7.0.3300.0, Culture=neutral, Pub
atk-sharp, Version=2.12.0.0, Culture=neutral, Publi
AuditPolicyGPManagedStubs.Interop, Version=6.1
Libraries used by this project
MessageBox
Environment
Load types from a Dynamic Link Library (DLL) file

Select Db4Objects.db4o.Dll, click Open:


Select a .Net DLL file
« Program Files (x86) ▸ db4o ▸ db4o-8.0 ▸ bin ▸ net-3.5
Search net-...
Organize ▾
New folder
Name
Date modified
Type
Cecil.FlowAnalysis.dll 1/28/2011 9:07 PM Applica
Db4oBench.dll 1/28/2011 9:07 PM Applica
Db4objects.Db4o.CS.dll 1/28/2011 9:07 PM Applica
DB4objects.Db4o.CS.Optional.dll 1/28/2011 9:07 PM Applica
Db4objects.Db4o.Data.Services.dll 1/28/2011 9:07 PM Applica
Db4objects.Db4o.dll 1/28/2011 9:07 PM Applica
Db4objects.Db4o.Instrumentation.dll 1/28/2011 9:07 PM Applica
Db4objects.Db4o.Linq.dll 1/28/2011 9:07 PM Applica
Db4objects.Db4o.NativeQueries.dll 1/28/2011 9:07 PM Applica
Db4objects.Db4o.Optional.dll 1/28/2011 9:07 PM Applica
File name: Db4objects.Db4o.dll
.Net DLL (*.Dll)
Open
Cancel

Expand Db4objects.Db4o:


Select Libraries

Global libraries

WindowsFormsIntegration, Version=3.0.0.0, Culture

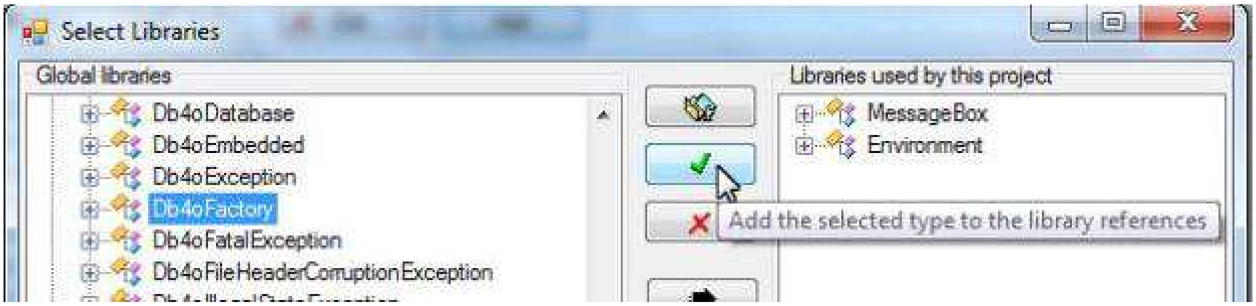
WsatConfig, Version=3.0.0.0, Culture=neutral, Pub

Db4objects.Db4o, Version=8.0.184.15484, Culture

AbstractBTreeRangeIterator

AbstractBufferContext

Select the Db4oFactory and click the button:


Select Libraries

Global libraries

Db4oDatabase

Db4oEmbedded

Db4oException

Db4oFactory

Db4oFatalException

Db4oFileHeaderCorruptionException

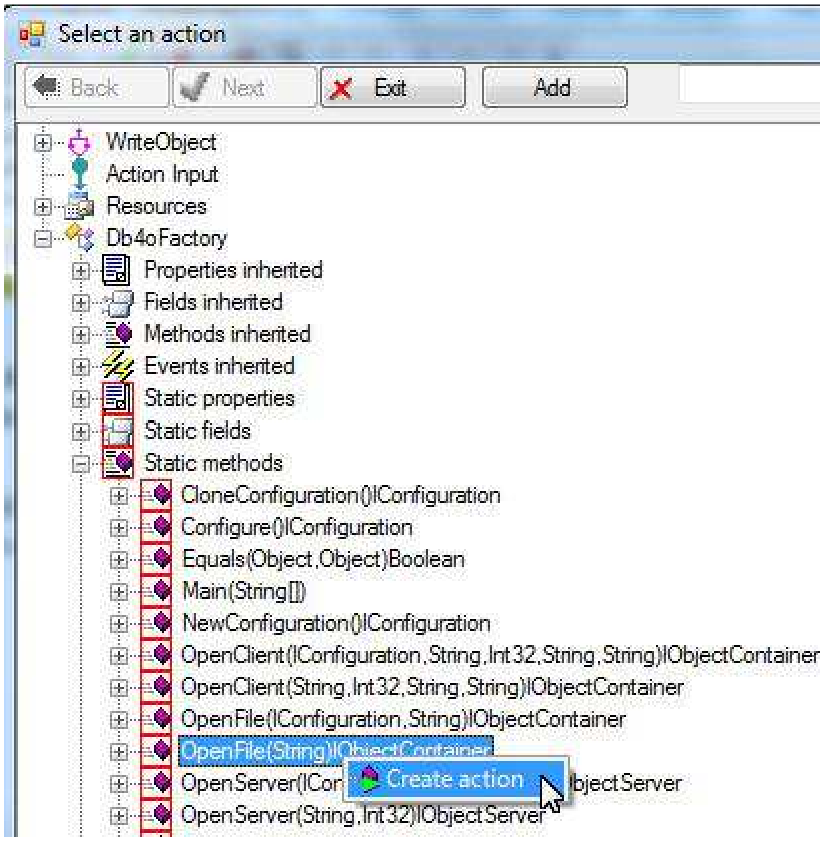
Libraries used by this project

MessageBox

Environment

Add the selected type to the library references

Right-click the OpenFile method; choose “Create action”:


Select an action

WriteObject

Action Input

Resources

Db4oFactory

Properties inherited

Fields inherited

Methods inherited

Events inherited

Static properties

Static fields

Static methods

CloneConfiguration()IConfiguration

Configure()IConfiguration

Equals(Object,Object)Boolean

Main(String[])

NewConfiguration()IConfiguration

OpenClient(IConfiguration,String,Int32,String,String)IObjectContainer

OpenClient(String,Int32,String,String)IObjectContainer

OpenFile(IConfiguration,String)IObjectContainer

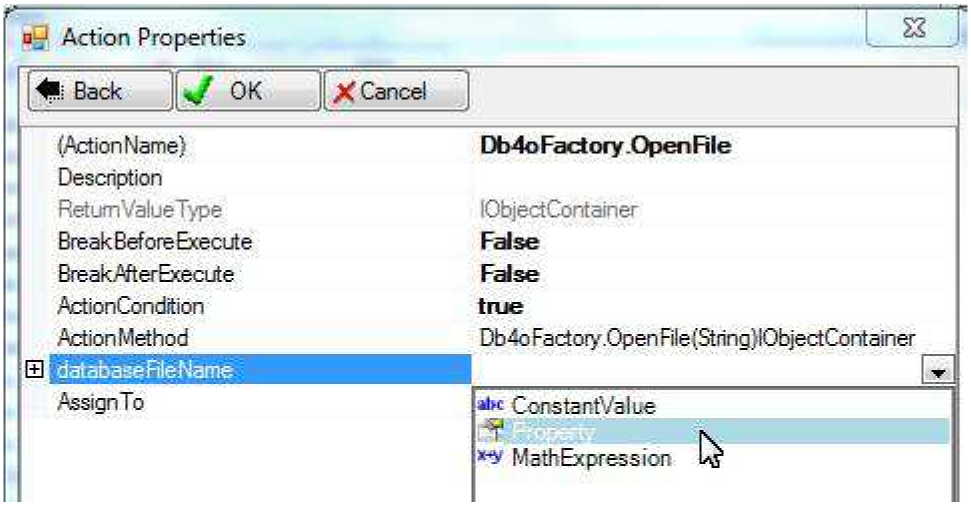
OpenFile(String)IObjectContainer

OpenServer(ICon… bjectServer

OpenServer(String,Int32)IObjectServer

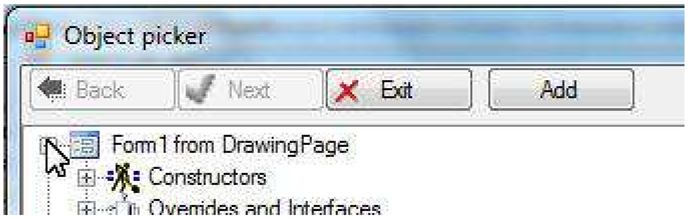
Create action

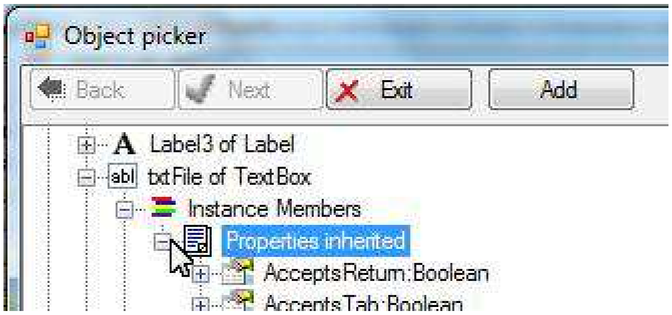
For “databaseFileName”, choose “Property”:

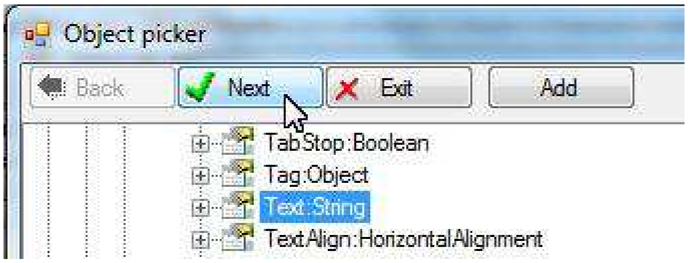


(ActionName)	Db4oFactory.OpenFile
Description	
ReturnValueType	IObjectContainer
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	Db4oFactory.OpenFile(String)IObjectContainer
databaseFileName	
AssignTo	

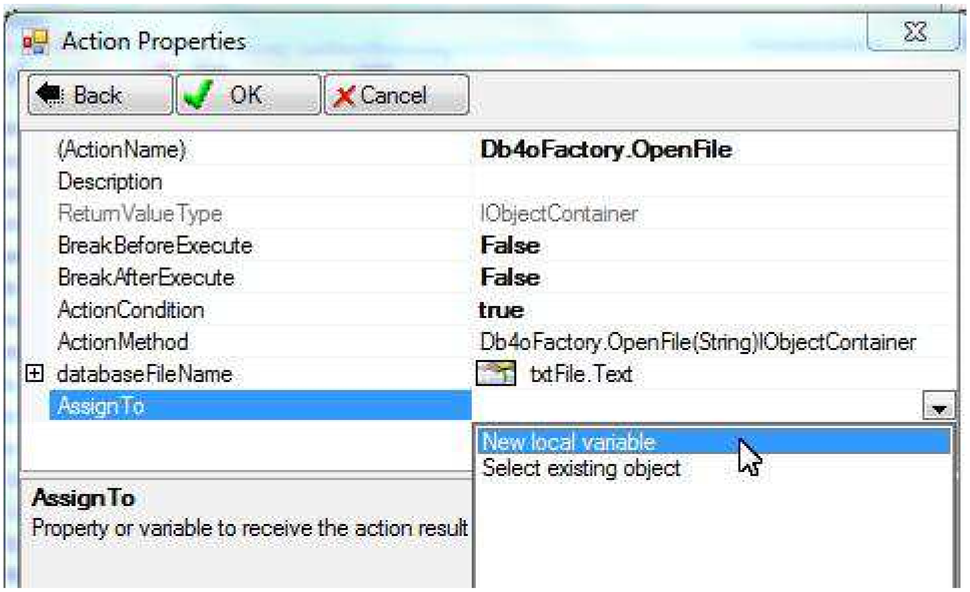
Select the Text property of the database text box:







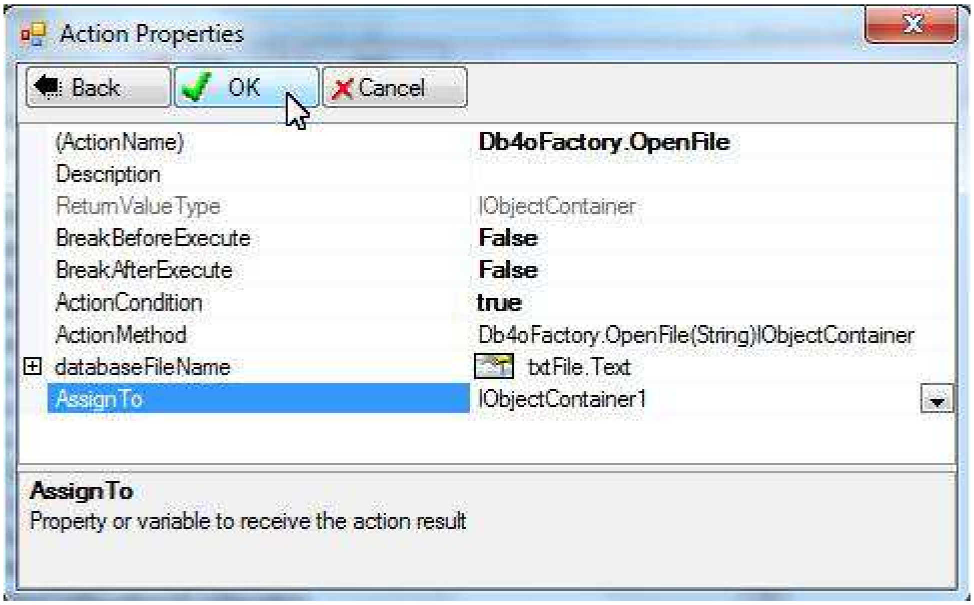
For “AssignTo”, choose “New local variable”:



(ActionName)	Db4oFactory.OpenFile
Description	
ReturnValueType	IObjectContainer
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	Db4oFactory.OpenFile(String)IObjectContainer
⊞ databaseFileName	txtFile.Text
AssignTo	

AssignTo
Property or variable to receive the action result

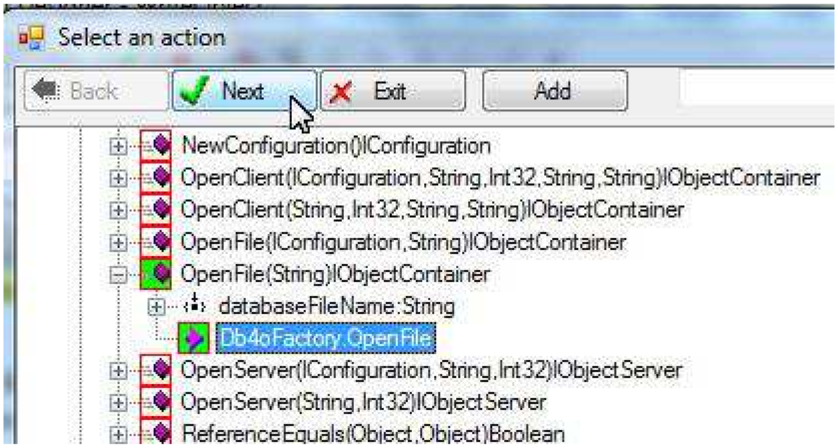
Click OK to finish creating this action:



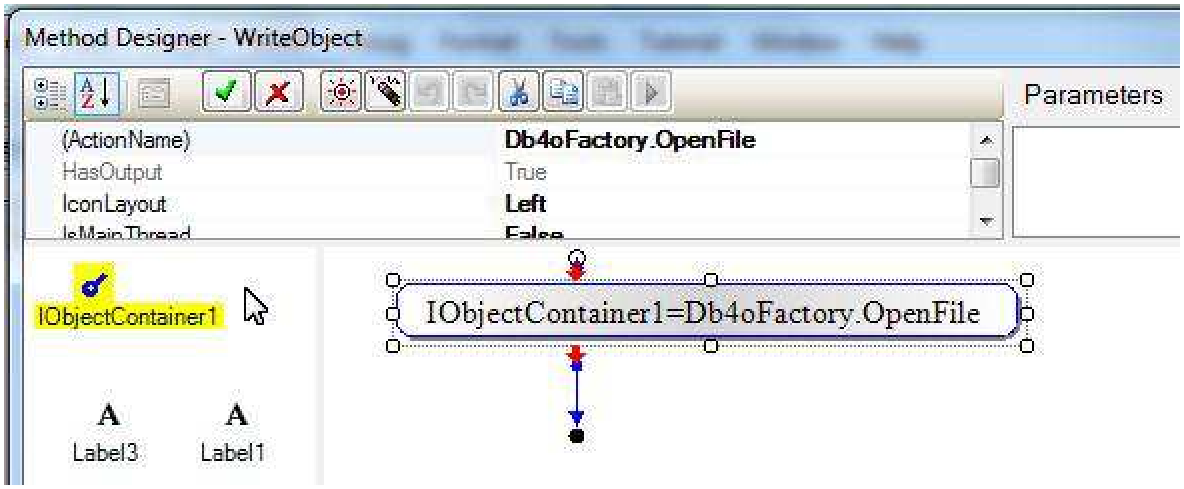
(ActionName)	Db4oFactory.OpenFile
Description	
ReturnValueType	IObjectContainer
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	Db4oFactory.OpenFile(String)IObjectContainer
⊞ databaseFileName	txtFile.Text
AssignTo	IObjectContainer1

AssignTo
Property or variable to receive the action result

With the new action selected, click “Next”:


Select an action
Back
Next
Exit
Add
NewConfiguration()IConfiguration
OpenClient(IConfiguration,String,Int32,String,String)IObjectContainer
OpenClient(String,Int32,String,String)IObjectContainer
OpenFile(IConfiguration,String)IObjectContainer
OpenFile(String)IObjectContainer
databaseFileName:String
Db4oFactory.OpenFile
OpenServer(IConfiguration,String,Int32)IObjectServer
OpenServer(String,Int32)IObjectServer
ReferenceEquals(Object,Object)Boolean

The new action appears in the Action Pane. A new variable, IObjectContainer1, appears in the variable Pane:

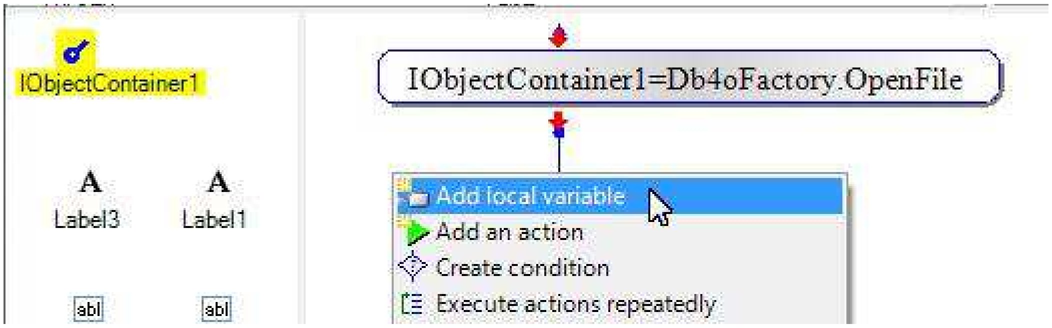

Method Designer - WriteObject
Parameters
IObjectContainer1
Label3
Label1
IObjectContainer1=Db4oFactory.OpenFile

(ActionName)	Db4oFactory.OpenFile
HasOutput	True
IconLayout	Left
IsMainThread	False

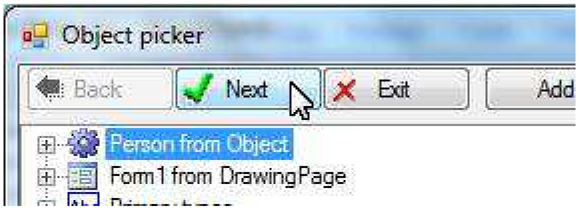
Create a sample object

We create a Person object to demonstrate saving object to database.

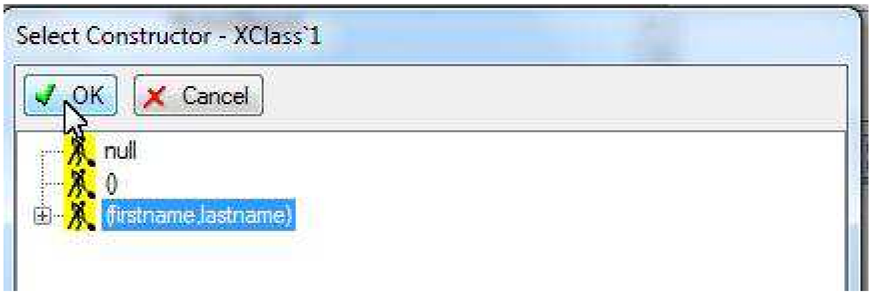
Right-click the Action-Pane; choose “Add local variable”



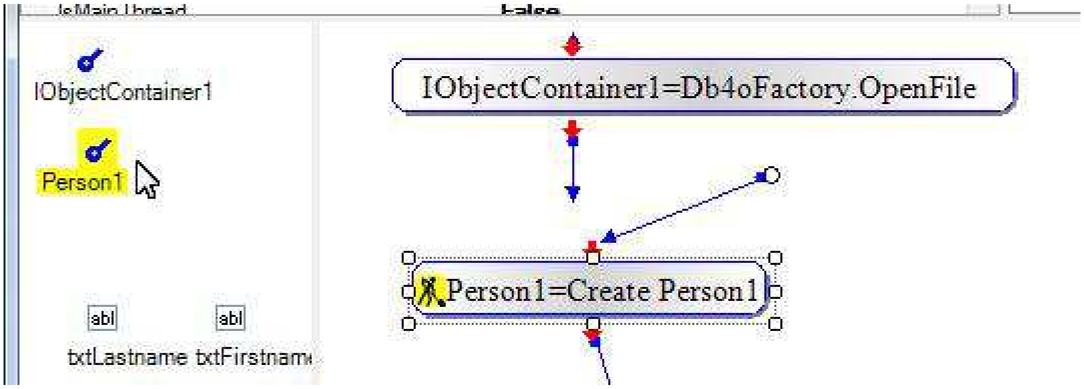
Select Person; click “Next”:



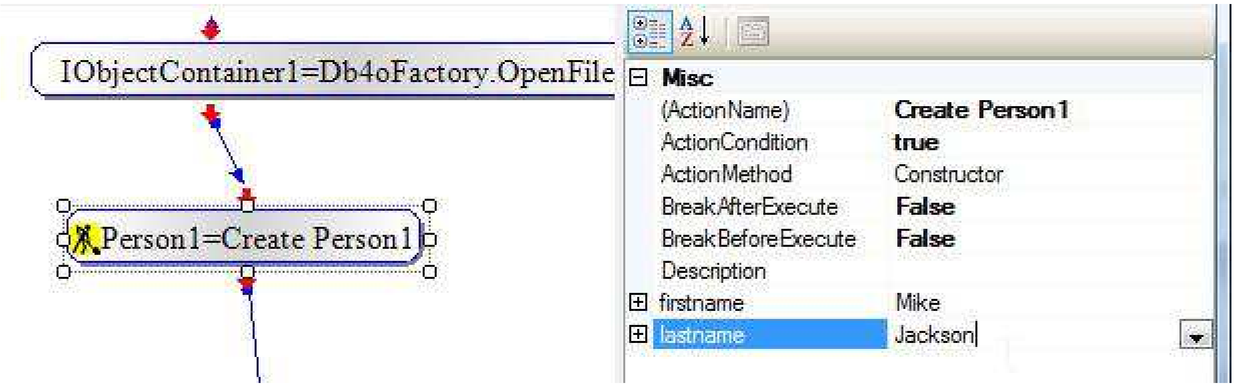
Select the constructor with parameters:



A new Person object, Person1, appears in the Variable Pane. The action creating the Person object appears in the Action-Pane.



Link the action to the previous action. We may set the parameters for the action:

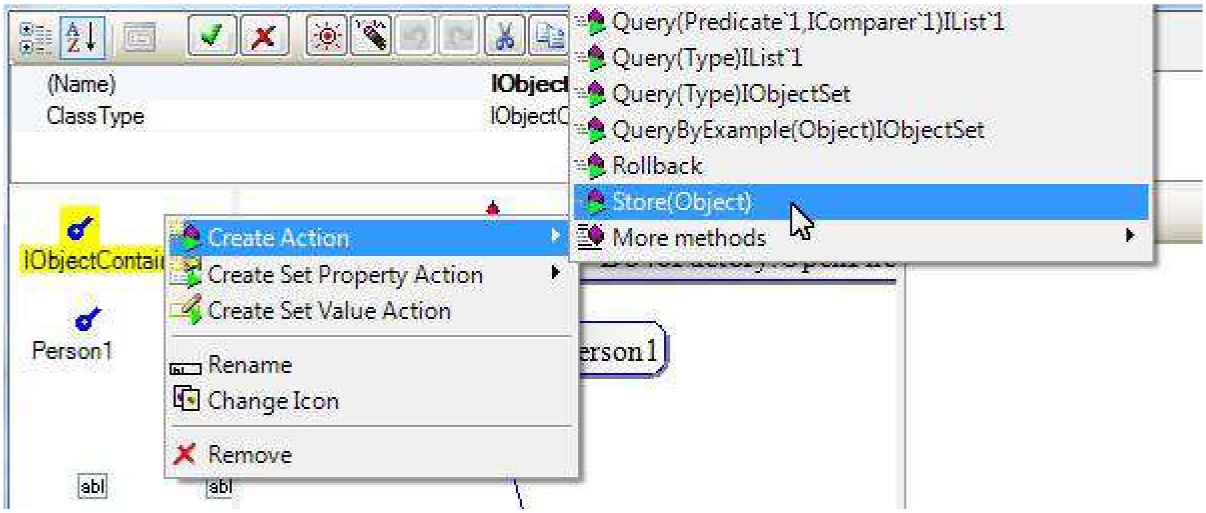


Misc	
(ActionName)	Create Person1
ActionCondition	true
ActionMethod	Constructor
BreakAfterExecute	False
BreakBeforeExecute	False
Description	
⊞ firstname	Mike
⊞ lastname	Jackson

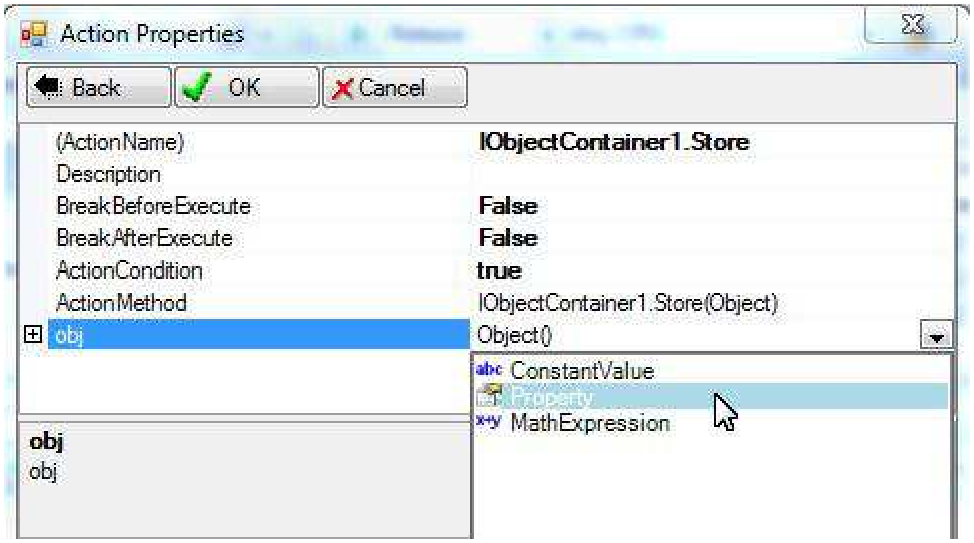
Save object to database

To save this Person object to the database, we may use a Store action.

Right-click the variable IObjectContainer1; choose “Create action”; choose “Store” method:



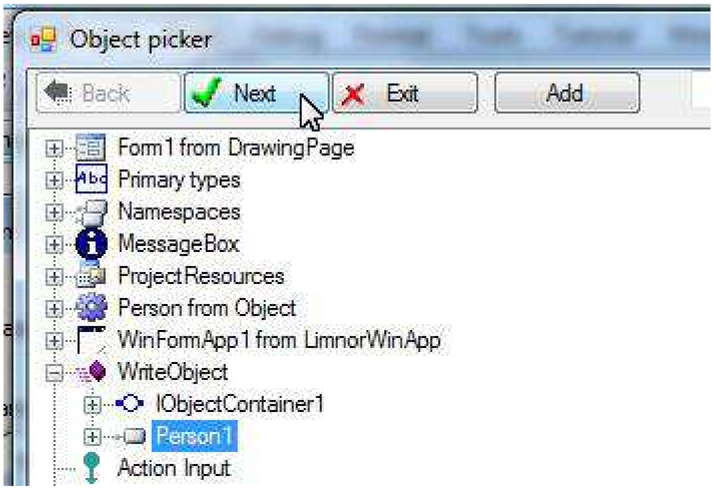
For “obj”, choose “Property”:



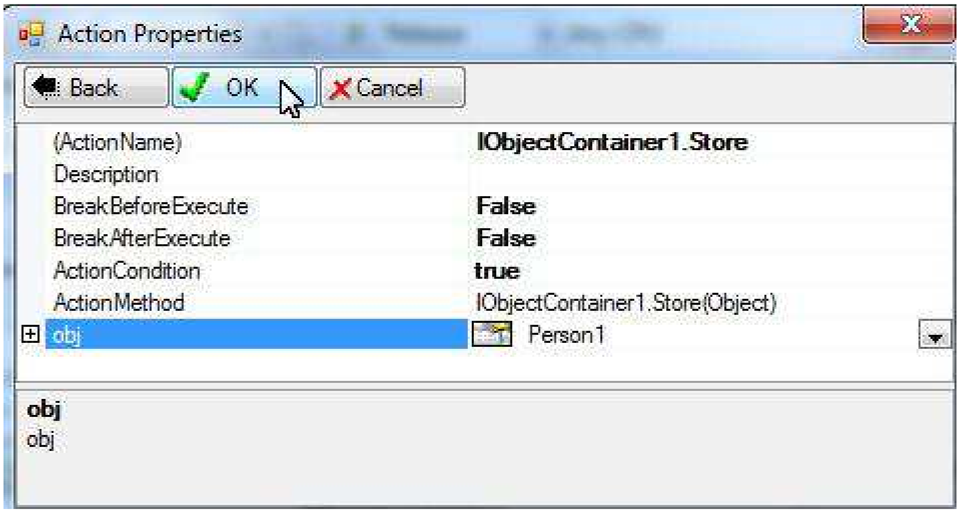
(ActionName)	IObjectContainer1.Store
Description	
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	IObjectContainer1.Store(Object)
obj	Object()

obj
obj

Choose “Person1”; click “Next”:



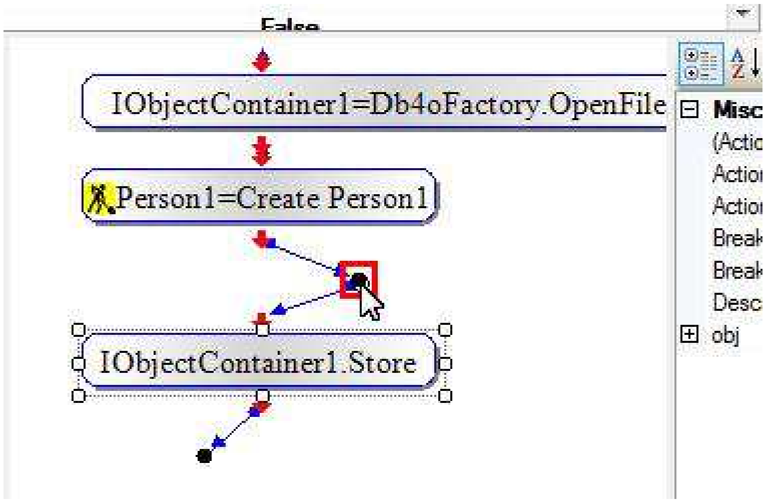
Click OK to finish creating this action:



(ActionName)	IObjectContainer1.Store
Description	
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	IObjectContainer1.Store(Object)
obj	Person1

obj
obj

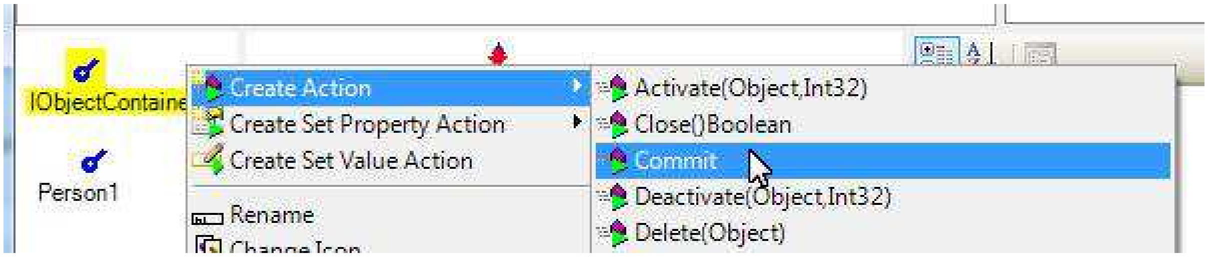
The action appears in the Action Pane. Link it to the last action:



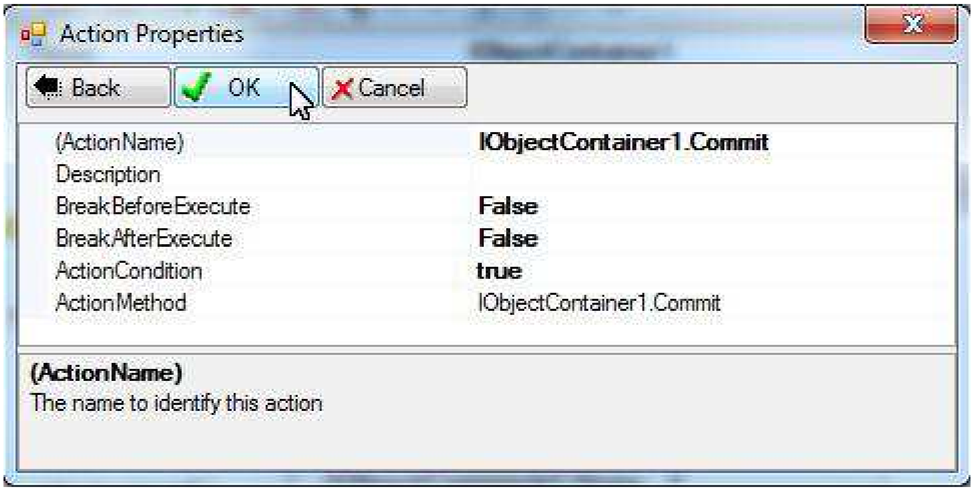
Commit and close database

According to the db4o documentation, we need to add a Commit action and a Close action to finish saving the object to the database.

Right-click variable IObjectContainer; choose “Create Action”; choose “Commit”:



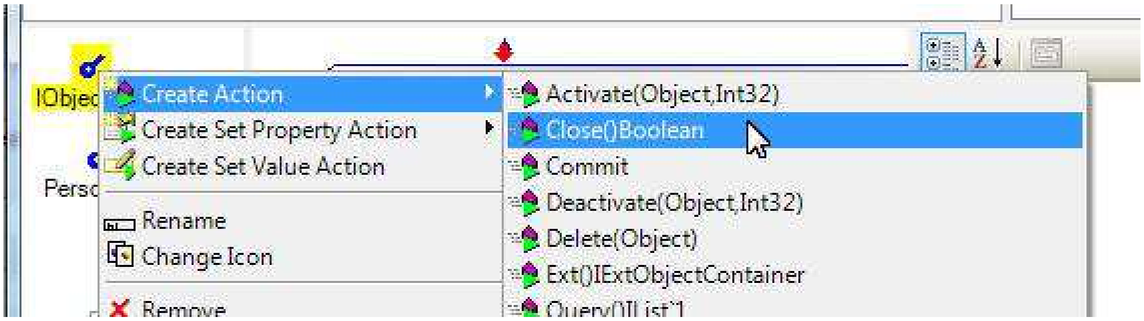
Click OK to finish creating this action:



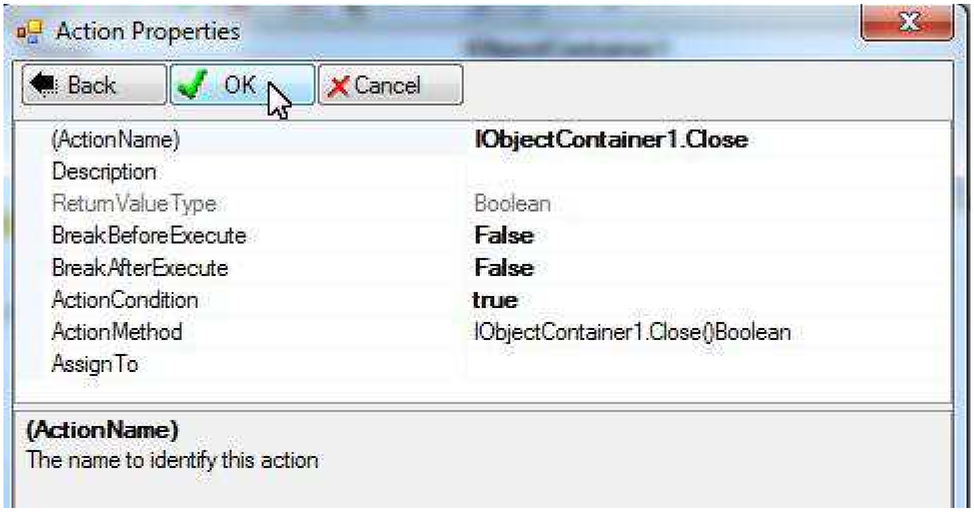
(ActionName)	**IObjectContainer1.Commit**
Description	
BreakBeforeExecute	**False**
BreakAfterExecute	**False**
ActionCondition	**true**
ActionMethod	IObjectContainer1.Commit

(ActionName)
The name to identify this action

Right-click variable IObjectContainer; choose “Create Action”; choose “Close”:



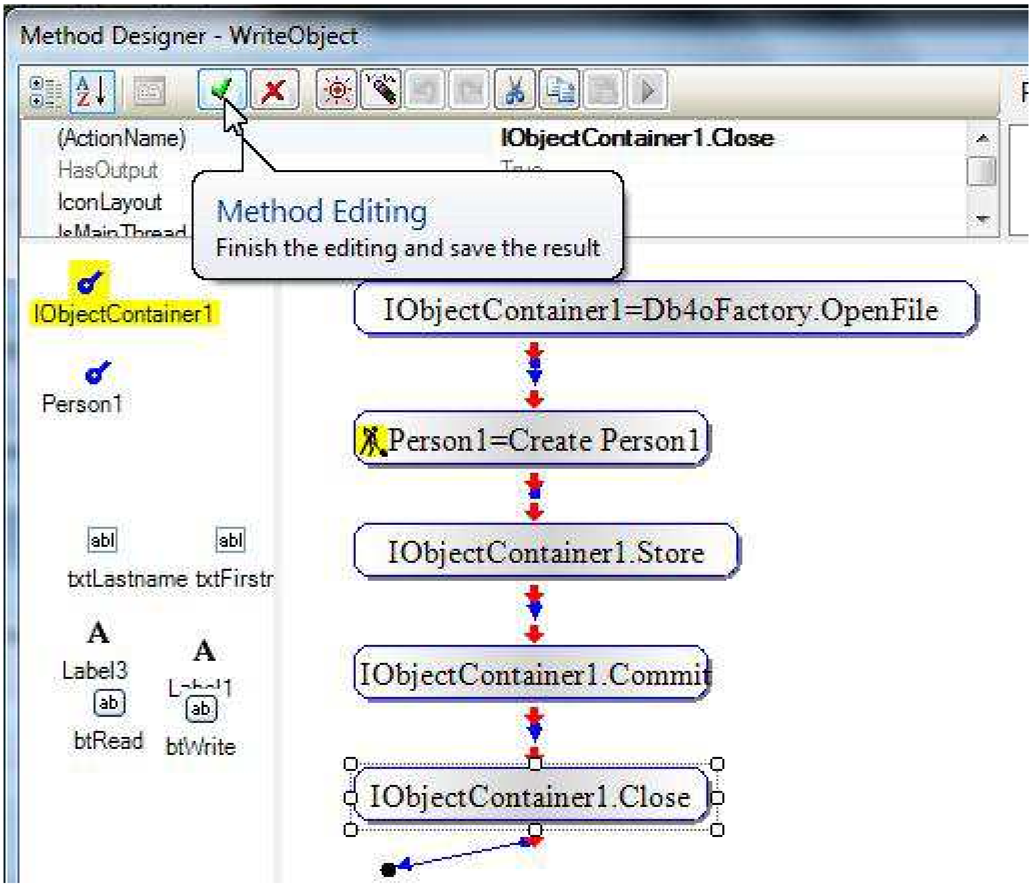
Click OK to finish creating this action:



(ActionName)	**IObjectContainer1.Close**
Description	
ReturnValueType	Boolean
BreakBeforeExecute	**False**
BreakAfterExecute	**False**
ActionCondition	**true**
ActionMethod	IObjectContainer1.Close()Boolean
AssignTo	

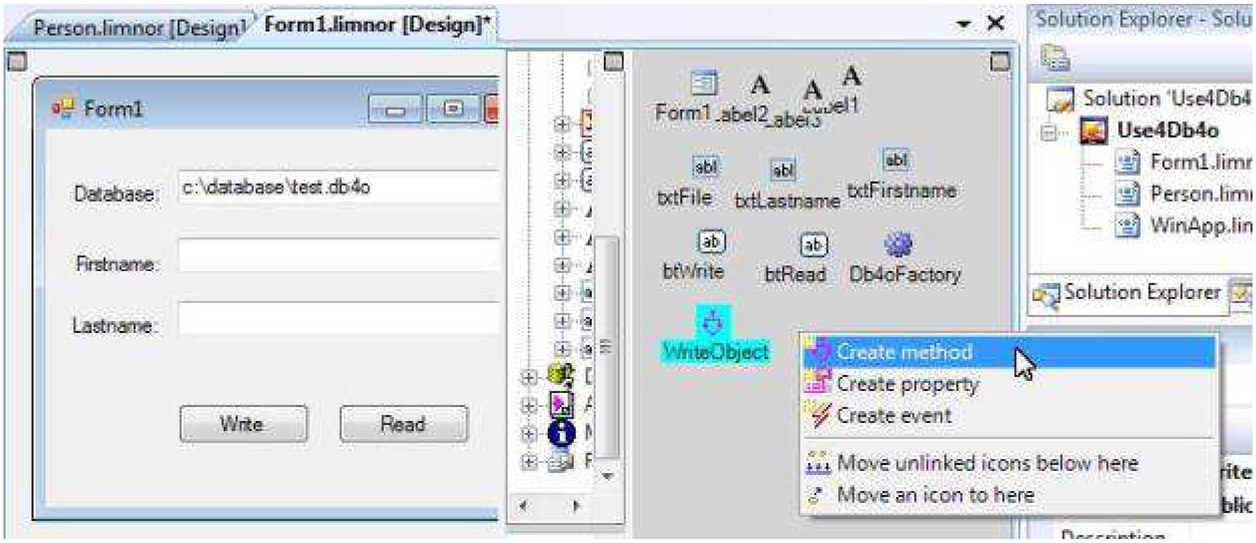
(ActionName)
The name to identify this action

Link the actions together. We are done creating this method:

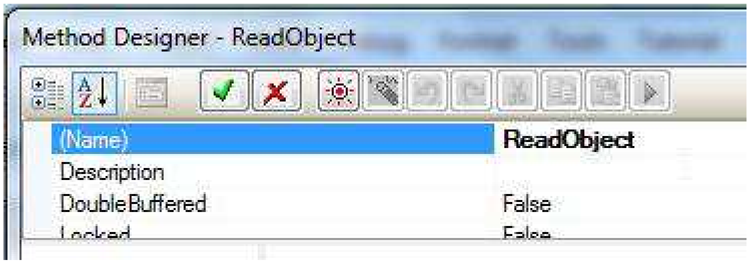


Read Object from Database

Create another method to read object from the database.



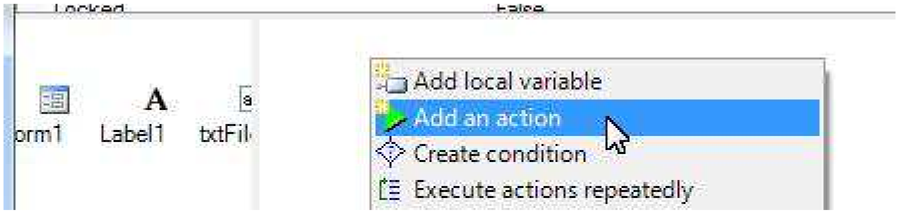
Rename it to ReadObject:



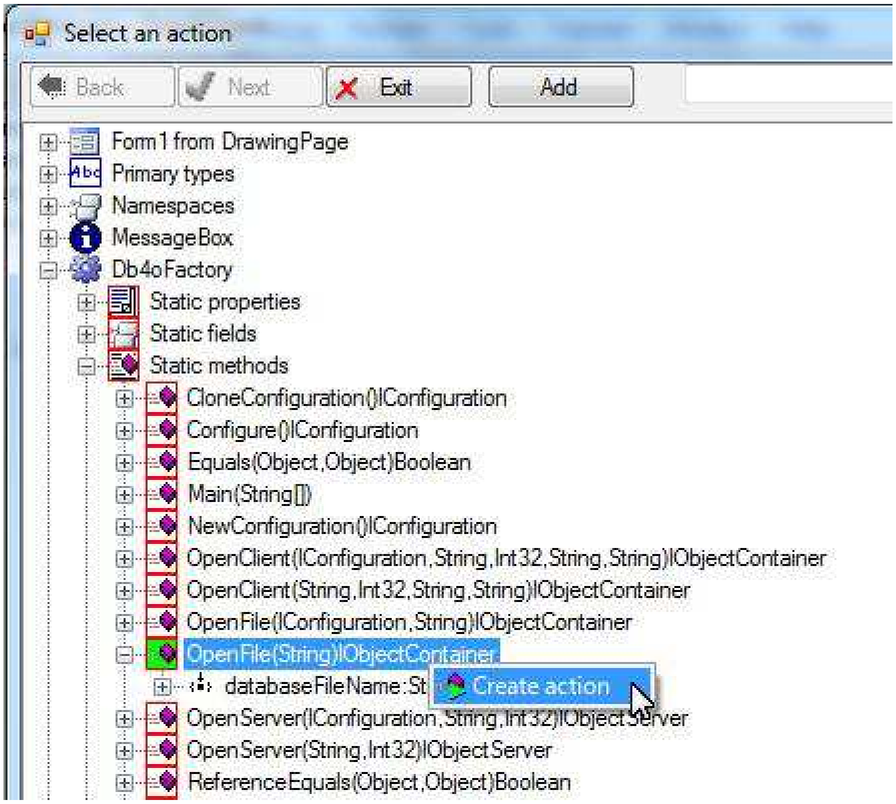
(Name)	ReadObject
Description	
DoubleBuffered	False
Locked	False

Open Database

Right-click the Action Pane; choose “Add an action”:

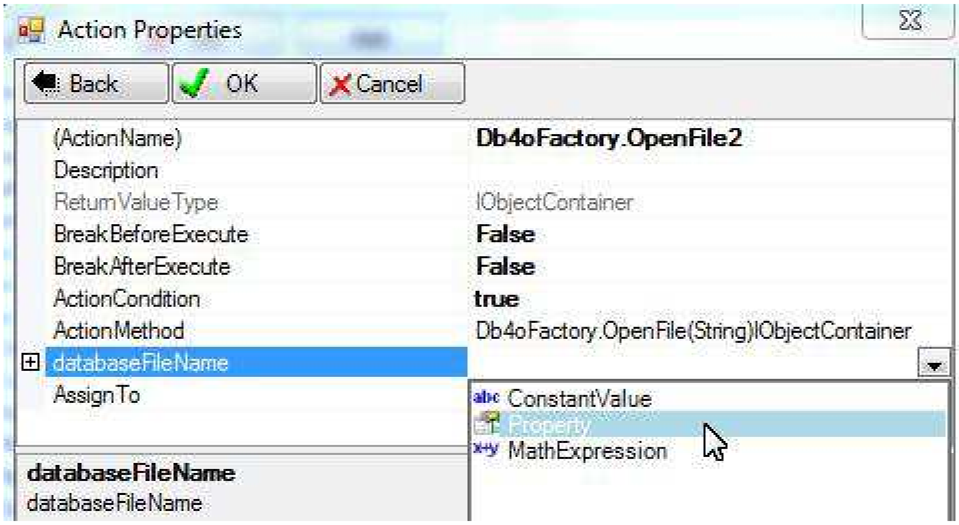


Right-click “OpenFile” method; choose “Create action”:



- Form1 from DrawingPage
- Primary types
- Namespaces
- MessageBox
- Db4oFactory
  - Static properties
  - Static fields
  - Static methods
    - CloneConfiguration()IConfiguration
    - Configure()IConfiguration
    - Equals(Object,Object)Boolean
    - Main(String[])
    - NewConfiguration()IConfiguration
    - OpenClient(IConfiguration,String,Int32,String,String)IObjectContainer
    - OpenClient(String,Int32,String,String)IObjectContainer
    - OpenFile(IConfiguration,String)IObjectContainer
    - OpenFile(String)IObjectContainer
      - databaseFileName:St
    - OpenServer(IConfiguration,String,Int32)IObjectServer
    - OpenServer(String,Int32)IObjectServer
    - ReferenceEquals(Object,Object)Boolean

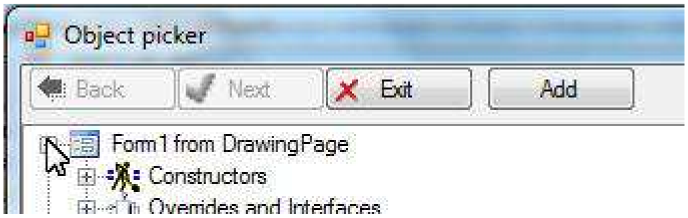
For “databaseFileName”, choose “Property”:

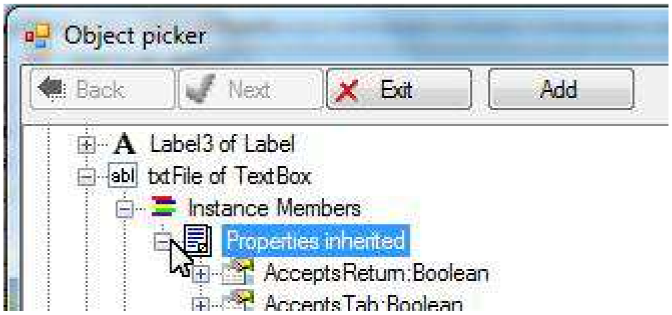


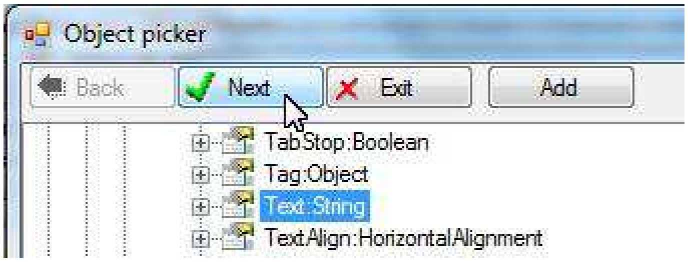
(ActionName)	Db4oFactory.OpenFile2
Description	
ReturnValueType	IObjectContainer
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	Db4oFactory.OpenFile(String)IObjectContainer
databaseFileName	
AssignTo	

databaseFileName
databaseFileName

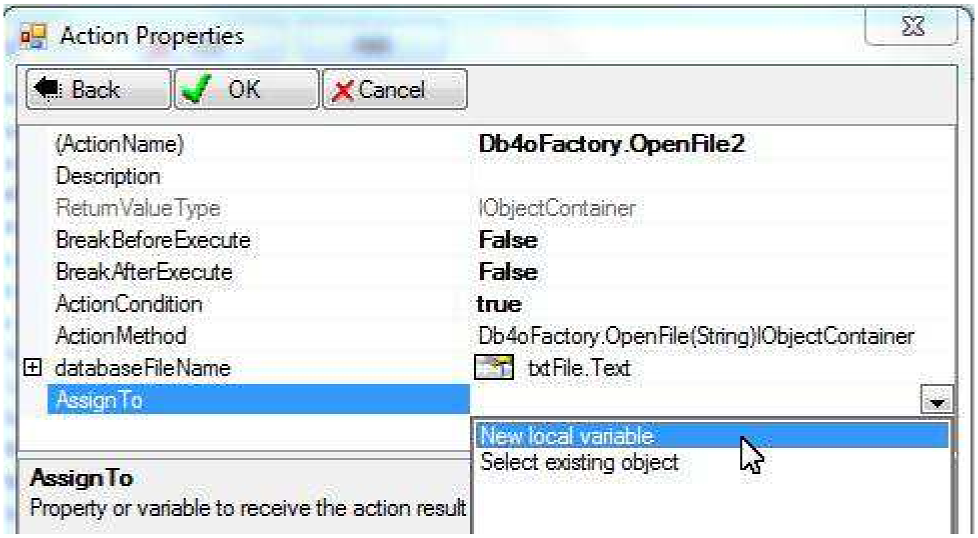
Select the Text property of the database text box:







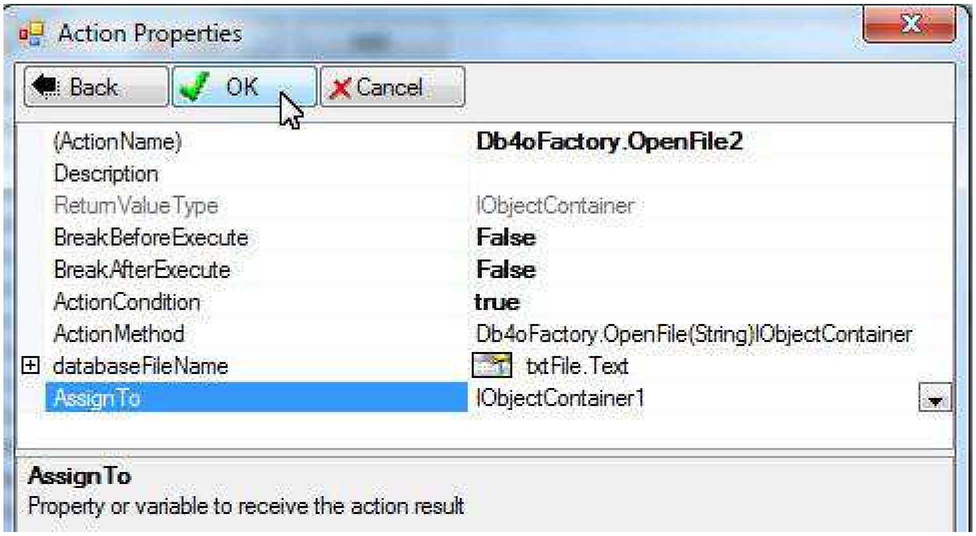
For “AssignTo”, choose “New local variable”:



(ActionName)	**Db4oFactory.OpenFile2**
Description	
ReturnValueType	IObjectContainer
BreakBeforeExecute	**False**
BreakAfterExecute	**False**
ActionCondition	**true**
ActionMethod	Db4oFactory.OpenFile(String)IObjectContainer
⊞ databaseFileName	txtFile.Text
AssignTo	

AssignTo
Property or variable to receive the action result

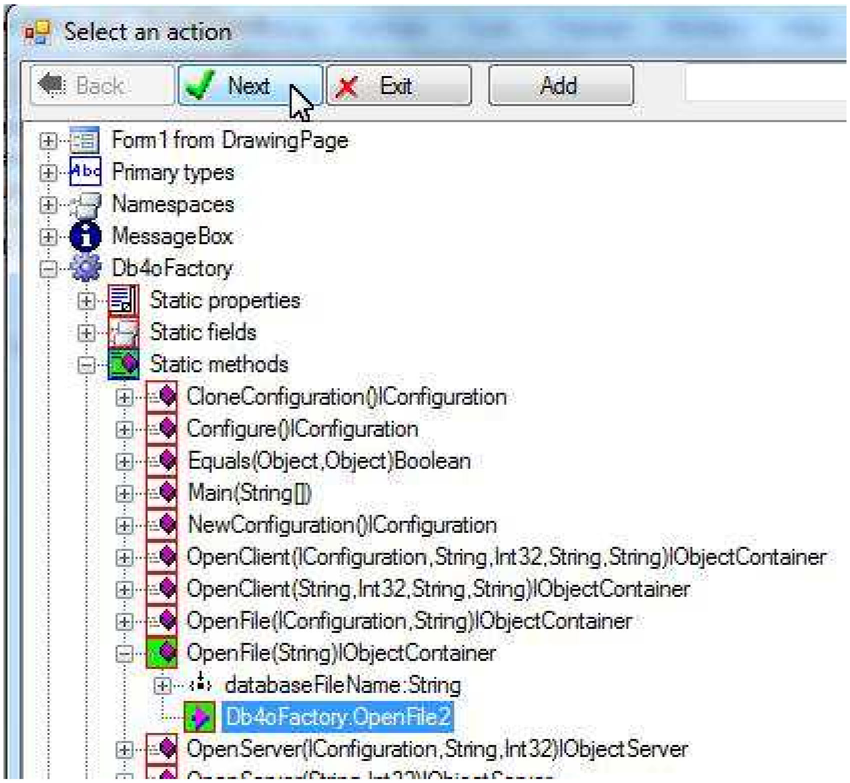
Click OK to finish creating this action:



(ActionName)	**Db4oFactory.OpenFile2**
Description	
ReturnValueType	IObjectContainer
BreakBeforeExecute	**False**
BreakAfterExecute	**False**
ActionCondition	**true**
ActionMethod	Db4oFactory.OpenFile(String)IObjectContainer
⊞ databaseFileName	txtFile.Text
AssignTo	IObjectContainer1

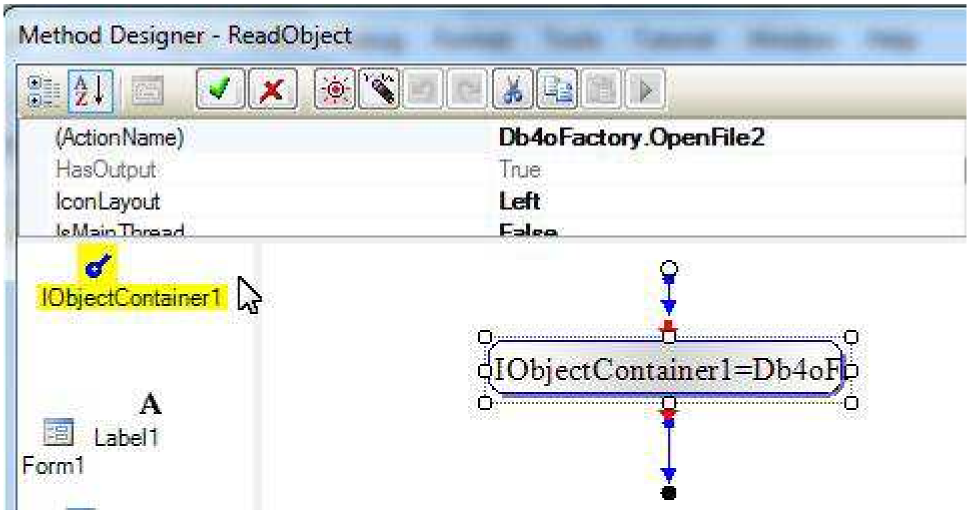
AssignTo
Property or variable to receive the action result

With the new action selected, click “Next”:



- Form1 from DrawingPage
- Primary types
- Namespaces
- MessageBox
- Db4oFactory
  - Static properties
  - Static fields
  - Static methods
    - CloneConfiguration()IConfiguration
    - Configure()IConfiguration
    - Equals(Object,Object)Boolean
    - Main(String[])
    - NewConfiguration()IConfiguration
    - OpenClient(IConfiguration,String,Int32,String,String)IObjectContainer
    - OpenClient(String,Int32,String,String)IObjectContainer
    - OpenFile(IConfiguration,String)IObjectContainer
    - OpenFile(String)IObjectContainer
      - databaseFileName:String
      - Db4oFactory.OpenFile2
    - OpenServer(IConfiguration,String,Int32)IObjectServer

The new action appears in the Action Pane. A new variable, IObjectContainer1, appears in the variable Pane:



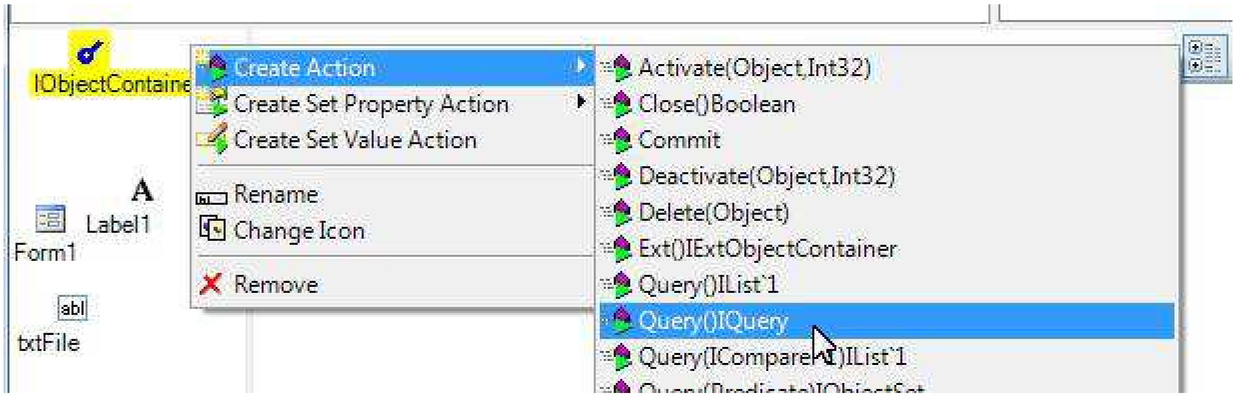
Method Designer - ReadObject

(ActionName)	Db4oFactory.OpenFile2
HasOutput	True
IconLayout	Left
IsMainThread	False

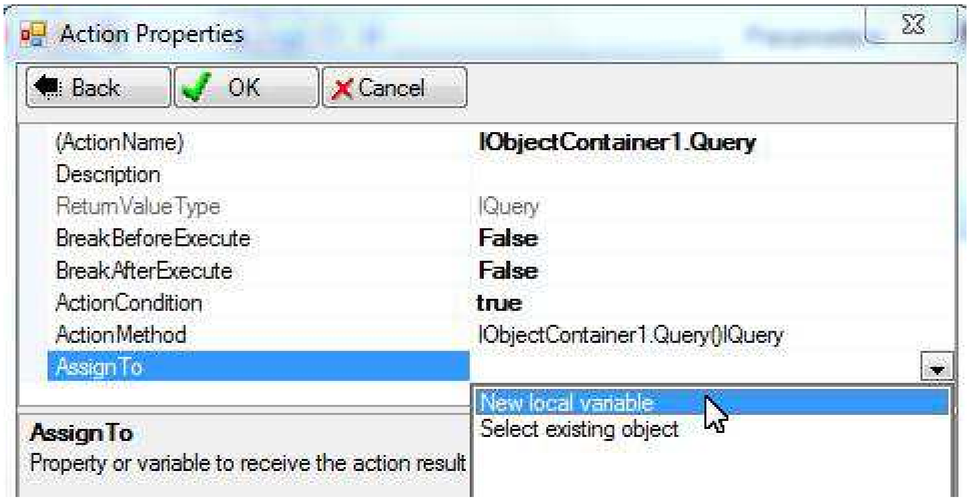
Create Query

According to the db4o document, we need to create IQuery to read from the database.

Right-click variable IObjectContainer; choose “Create Action”; choose “Query”:



For “AssignTo”, choose “New local variable”:



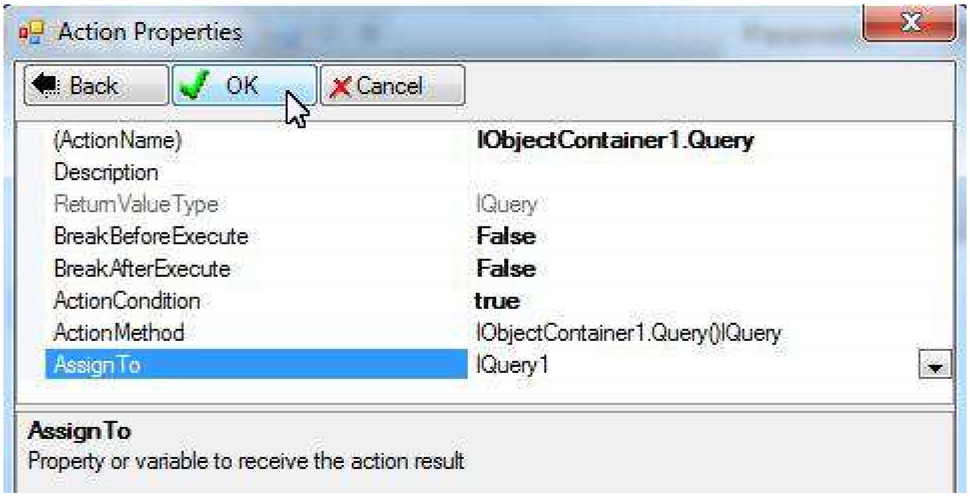
Action Properties

◀ Back | ✓ OK | ✗ Cancel

(ActionName)	IObjectContainer1.Query
Description	
ReturnValueType	IQuery
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	IObjectContainer1.Query()IQuery
AssignTo	

AssignTo
Property or variable to receive the action result

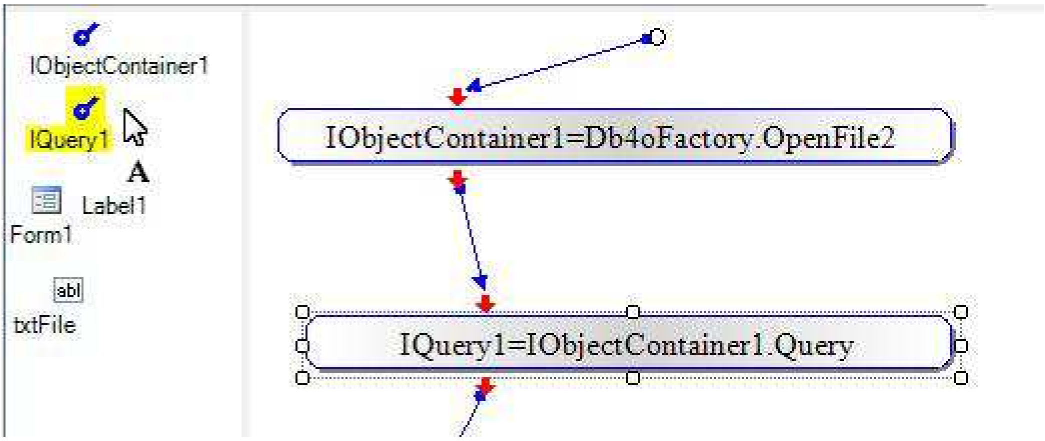
Click OK to finish creating this action:



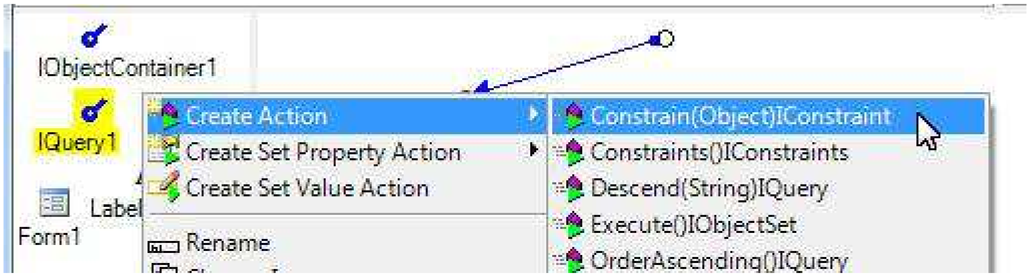
(ActionName)	**IObjectContainer1.Query**
Description	
ReturnValueType	IQuery
BreakBeforeExecute	**False**
BreakAfterExecute	**False**
ActionCondition	**true**
ActionMethod	IObjectContainer1.Query()IQuery
AssignTo	IQuery1

AssignTo
Property or variable to receive the action result

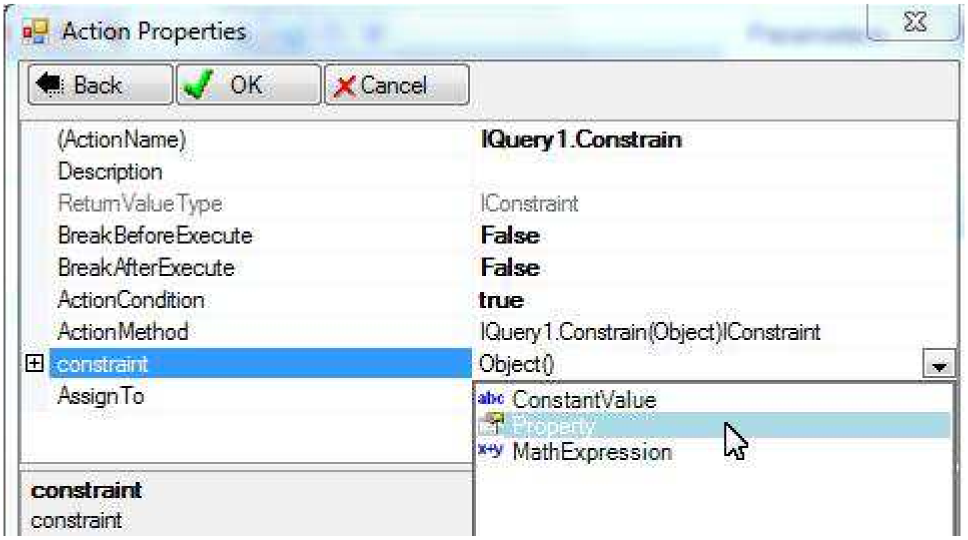
The new action appears in the Action Pane; a new variable, IQuery1, appears in the Variable Pane:


IObjectContainer1
IQuery1
A
Label1
Form1
abl
txtFile
IObjectContainer1=Db4oFactory.OpenFile2
IQuery1=IObjectContainer1.Query

Right-click IQuery1; choose “Create Action”; choose “Constrain”:



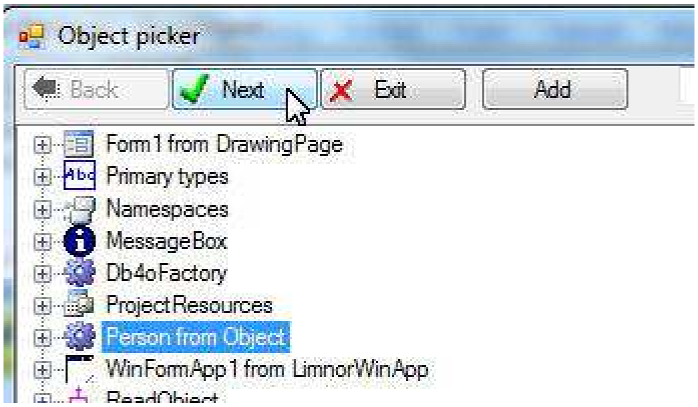
For “constraint”, choose “Property”:



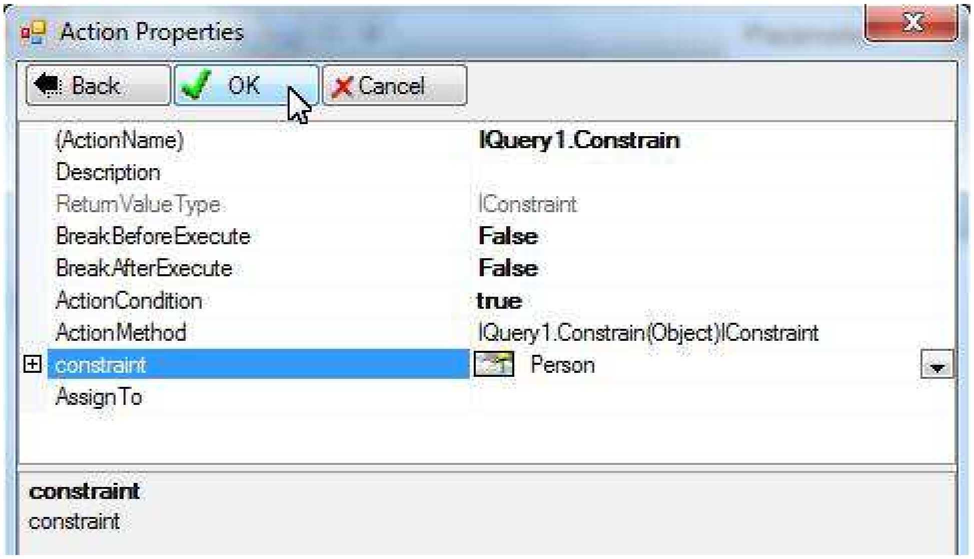
(ActionName)	**IQuery1.Constrain**
Description	
ReturnValueType	IConstraint
BreakBeforeExecute	**False**
BreakAfterExecute	**False**
ActionCondition	**true**
ActionMethod	IQuery1.Constrain(Object)IConstraint
⊞ constraint	Object()
AssignTo	

constraint
constraint

Select Person, click “Next”:



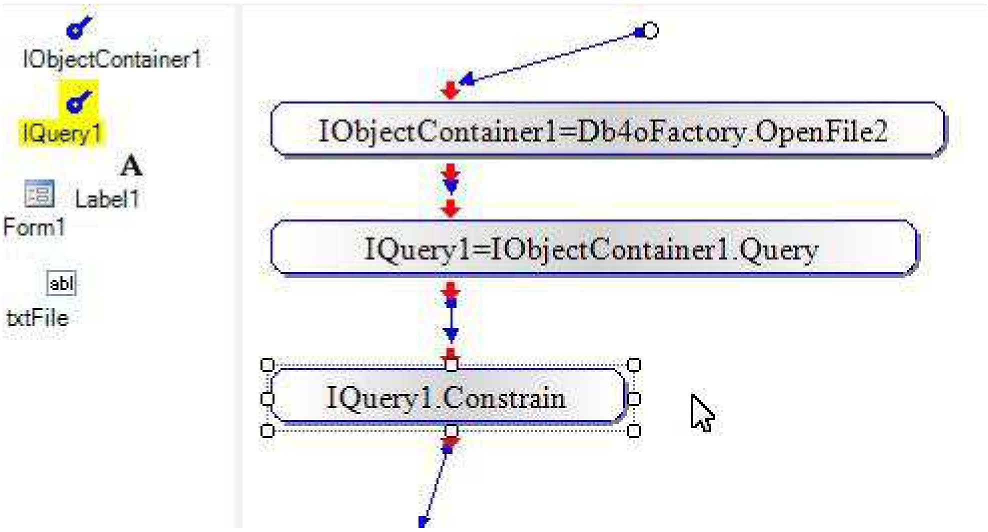
Click OK to finish creating this action:



(ActionName)	**IQuery1.Constrain**
Description	
ReturnValueType	IConstraint
BreakBeforeExecute	**False**
BreakAfterExecute	**False**
ActionCondition	**true**
ActionMethod	IQuery1.Constrain(Object)IConstraint
⊞ constraint	Person
AssignTo	

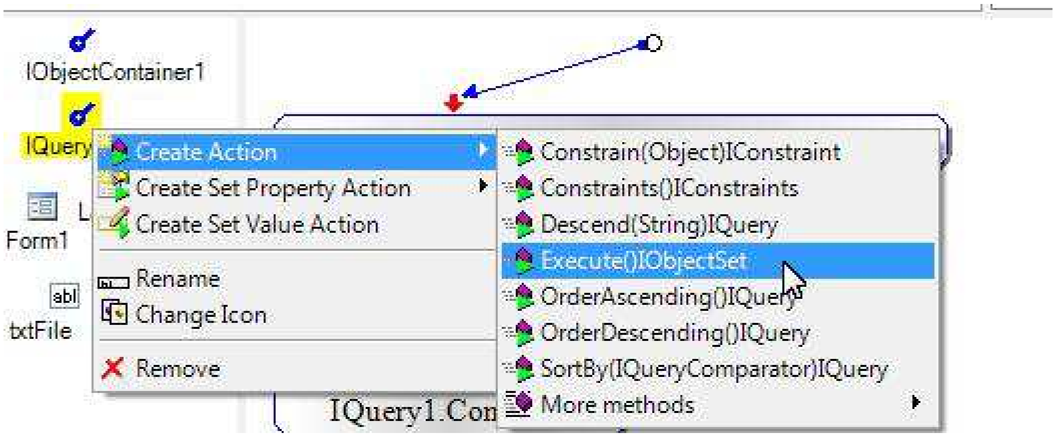
constraint
constraint

Link the actions together:

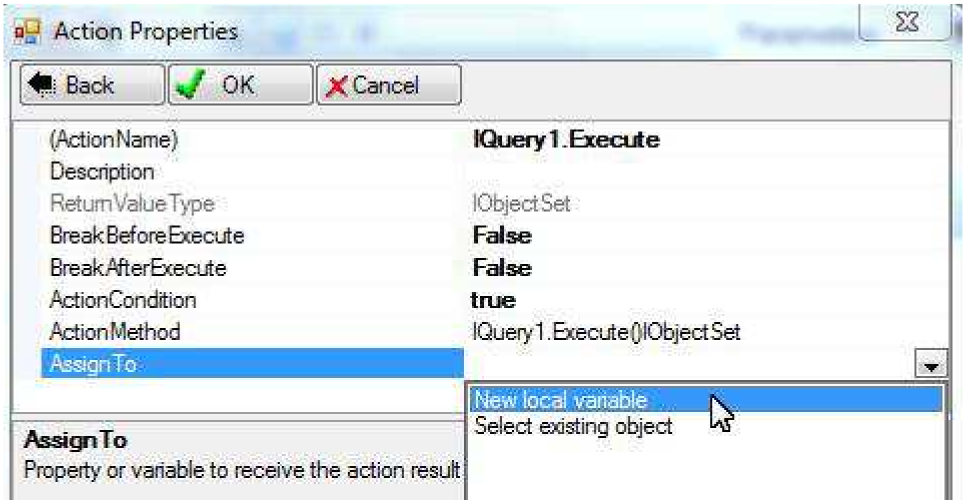

IObjectContainer1
IQuery1
Label1
Form1
txtFile
IObjectContainer1=Db4oFactory.OpenFile2
IQuery1=IObjectContainer1.Query
IQuery1.Constrain

Execute query

Right-click IQuery1; choose “Create Action”; choose “Execute”:



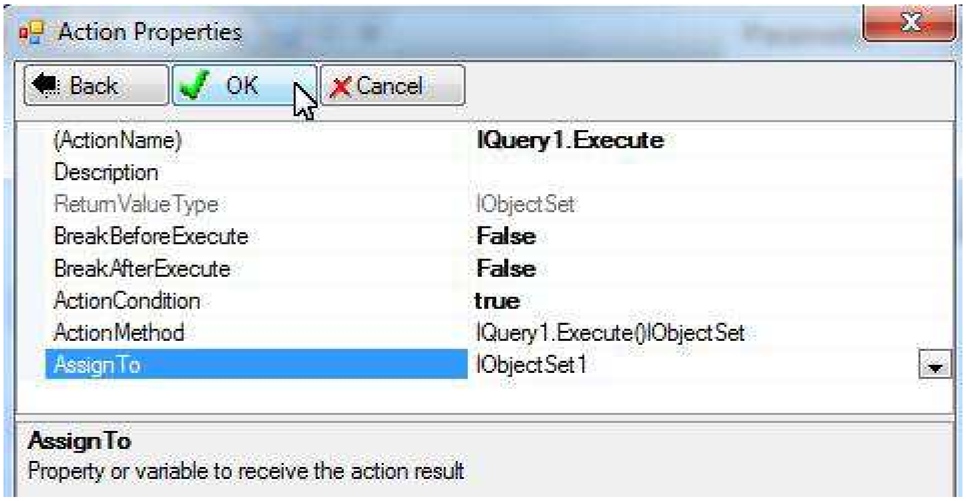
For “AssignTo”, choose “New local variable”:



(ActionName)	IQuery1.Execute
Description	
ReturnValueType	IObjectSet
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	IQuery1.Execute()IObjectSet
AssignTo	

AssignTo
Property or variable to receive the action result

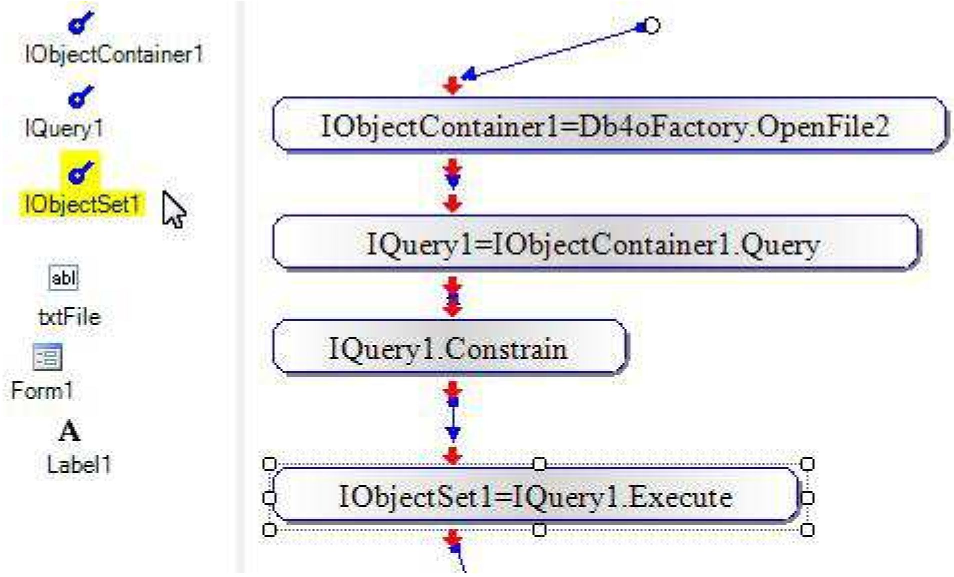
Click OK to finish creating this action:



(ActionName)	IQuery1.Execute
Description	
ReturnValueType	IObjectSet
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	IQuery1.Execute()IObjectSet
AssignTo	IObjectSet1

AssignTo
Property or variable to receive the action result

The new action appears in the Action Pane. A new variable, IObjectSet1, appears in the Variable Pane:

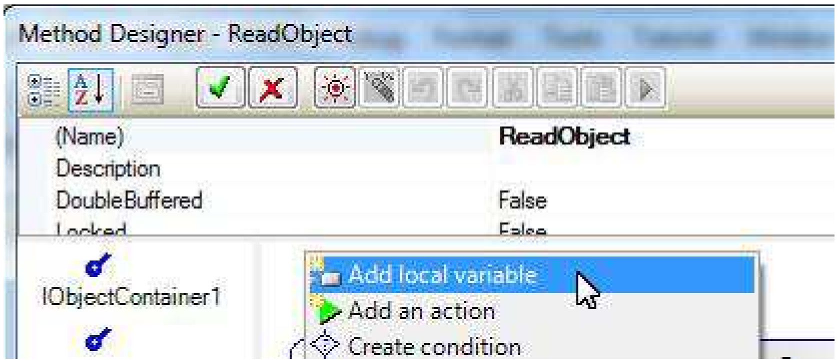

IObjectContainer1
IQuery1
IObjectSet1
txtFile
Form1
Label1
IObjectContainer1=Db4oFactory.OpenFile2
IQuery1=IObjectContainer1.Query
IQuery1.Constrain
IObjectSet1=IQuery1.Execute

Read Object

IObjectSet has a Next method for reading an object. We may repeatedly call Next to read all objects.

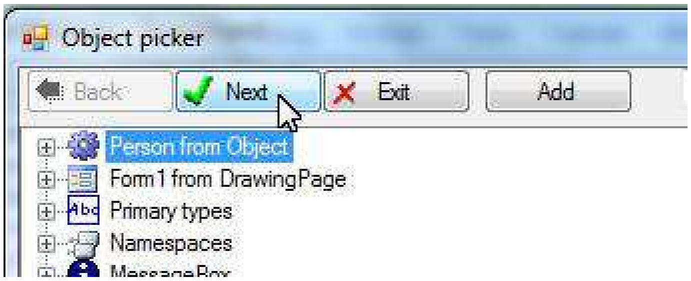
For this simple sample, we just call Next once to get one object.

To read a Person object, we first create a Person variable. Right-click the Action Pane; choose “Add local variable”:

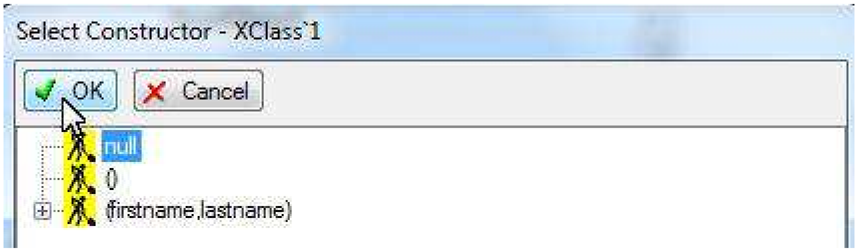


(Name)	ReadObject
Description	
DoubleBuffered	False
Locked	False

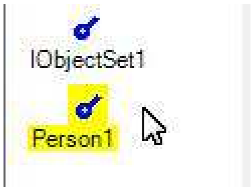
Select Person; click “Next”:



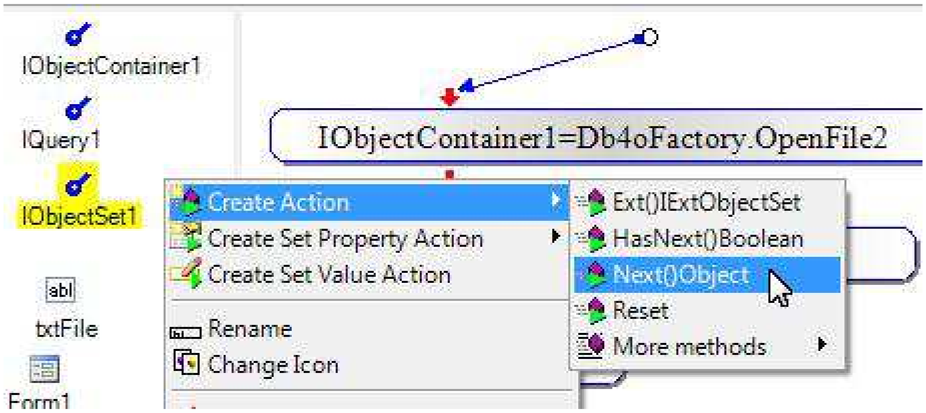
Select “null” because the object will be read from the database; click “OK”:



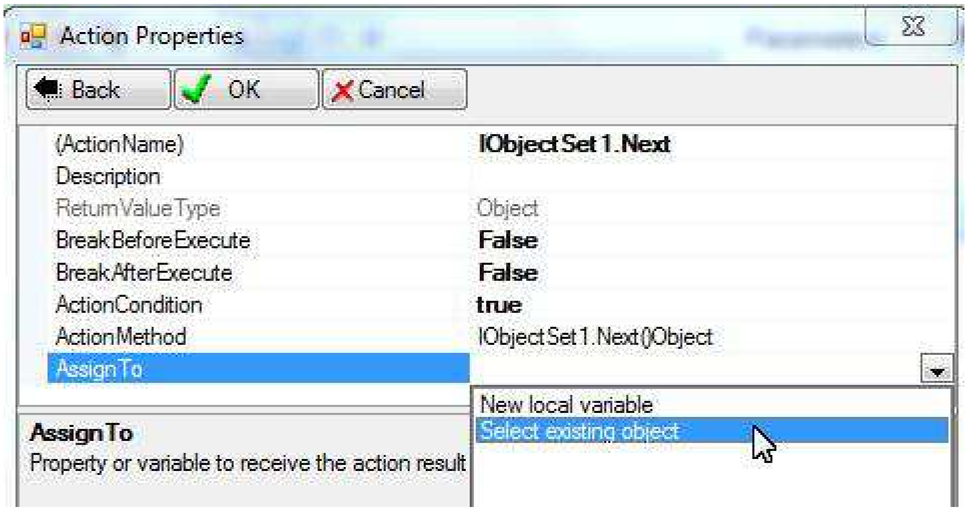
The new variable, Person1, appears in the Variable Pane:



Right-click IObjectSet1; choose “Create Action”; choose “Next”:



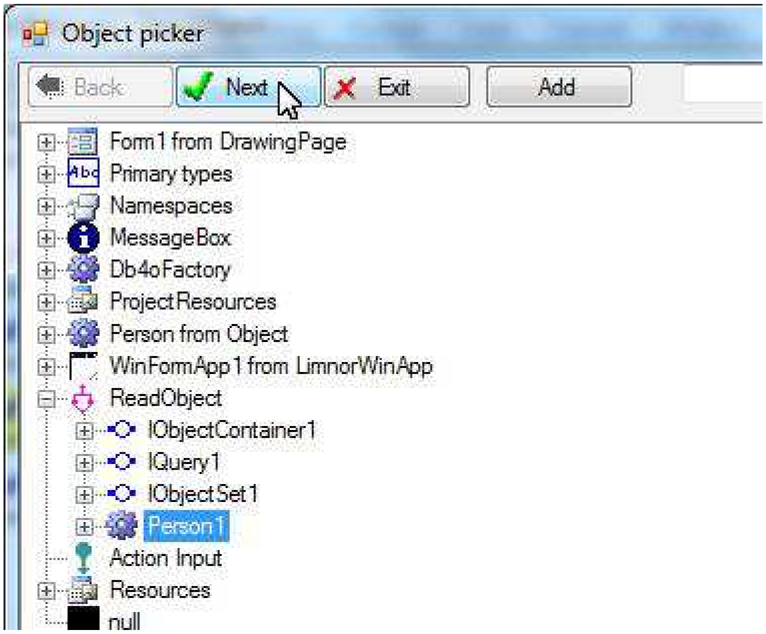
For “AssignTo”, choose “Select existing object”:



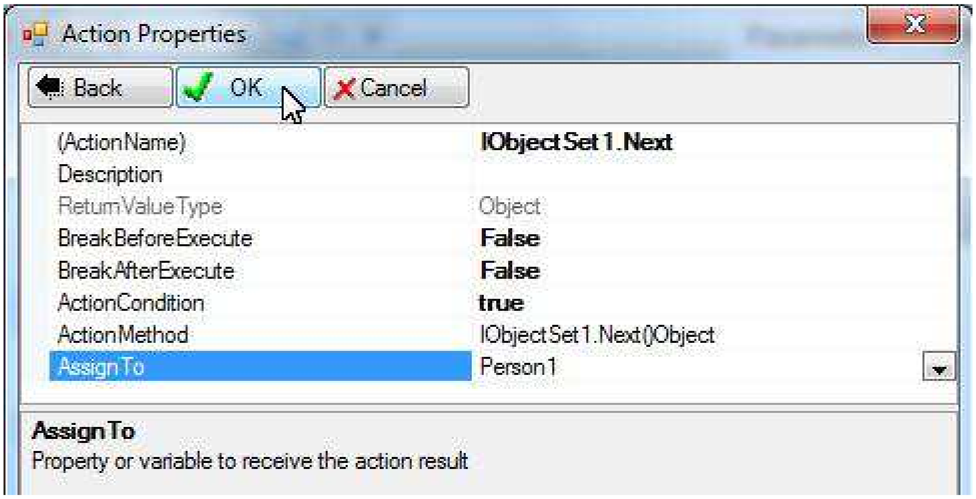
(ActionName)	IObjectSet1.Next
Description	
ReturnValueType	Object
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	IObjectSet1.Next()Object
AssignTo	

AssignTo
Property or variable to receive the action result

Select variable Person1; click “Next”:



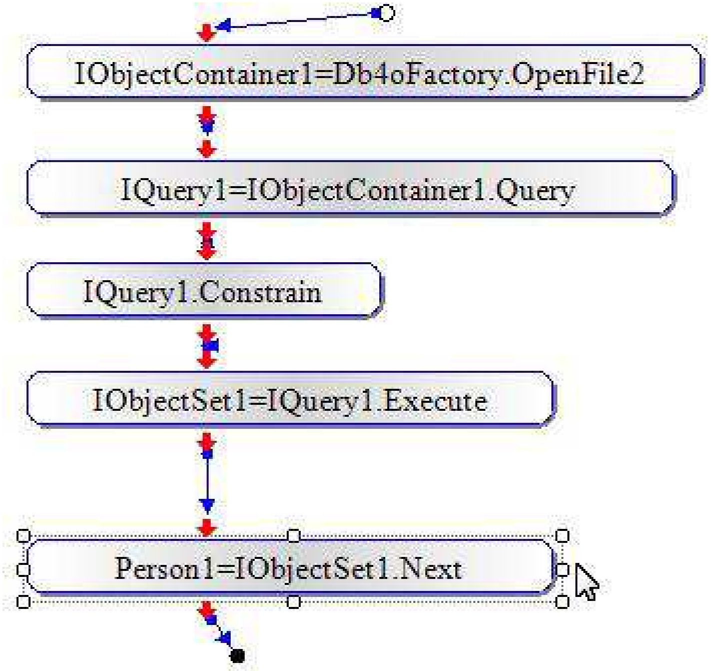
Click OK to finish creating this action:



(ActionName)	**IObjectSet1.Next**
Description	
ReturnValueType	Object
BreakBeforeExecute	**False**
BreakAfterExecute	**False**
ActionCondition	**true**
ActionMethod	IObjectSet1.Next()Object
AssignTo	Person1

AssignTo
Property or variable to receive the action result

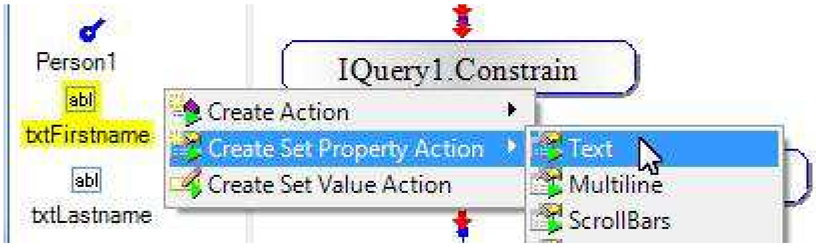
The new action appears in the Action Pane. Link it to the last action:


IObjectContainer1=Db4oFactory.OpenFile2
IQuery1=IObjectContainer1.Query
IQuery1.Constrain
IObjectSet1=IQuery1.Execute
Person1=IObjectSet1.Next

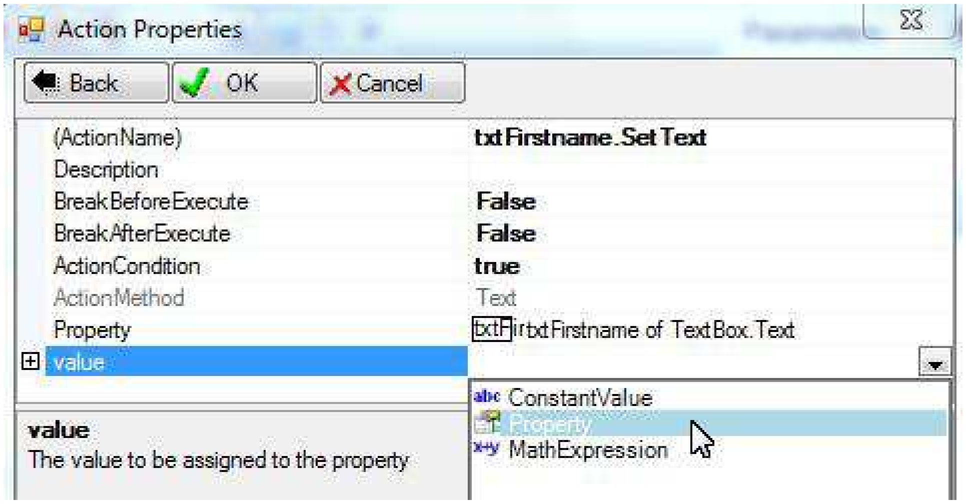
Show the object

After reading the object, we may display it on the form.

Right-click the text box for the first name; choose “Create Set Property Action”; choose “Text”:



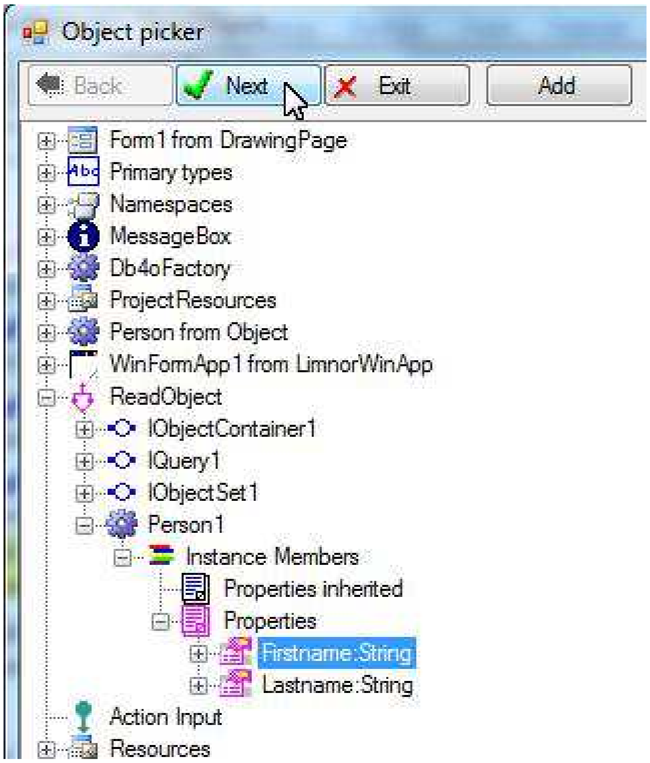
For the “value”, choose “Property”:



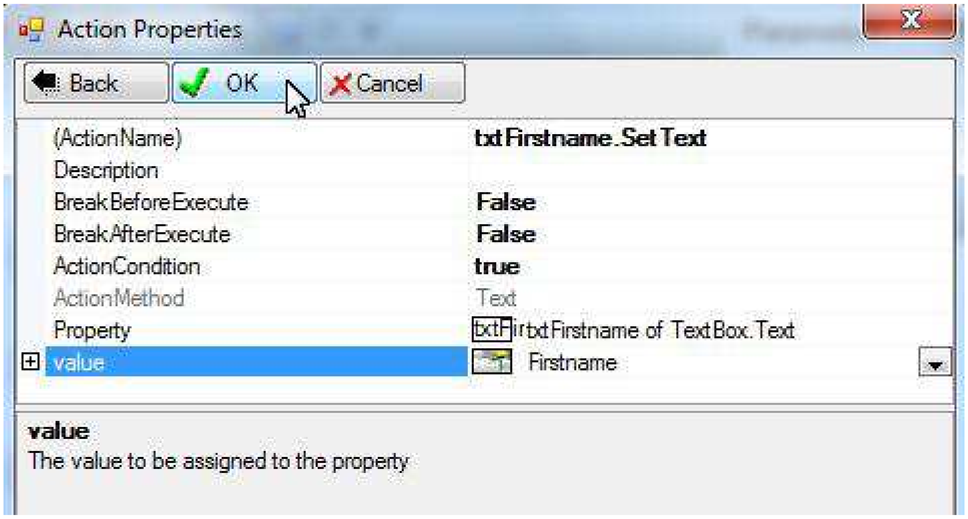
(ActionName)	txtFirstname.SetText
Description	
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	Text
Property	txtFirtxtFirstname of TextBox.Text
value	

value
The value to be assigned to the property

Select the Firstname of Person1; click “Next”:



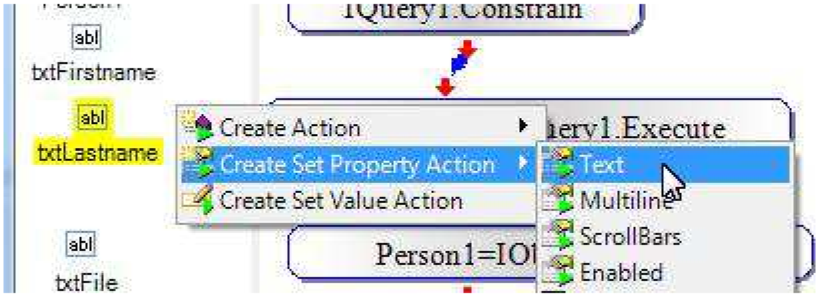
Click OK to finish creating this action:



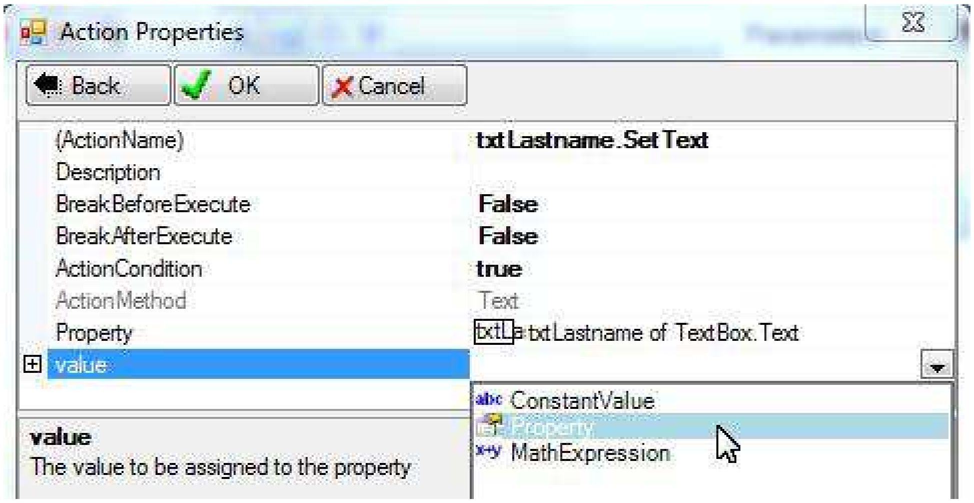
(ActionName)	txtFirstname.SetText
Description	
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	Text
Property	txtFirtxtFirstname of TextBox.Text
value	Firstname

value
The value to be assigned to the property

Right-click the text box for last name; choose "Create Set Property Action"; choose "Text":



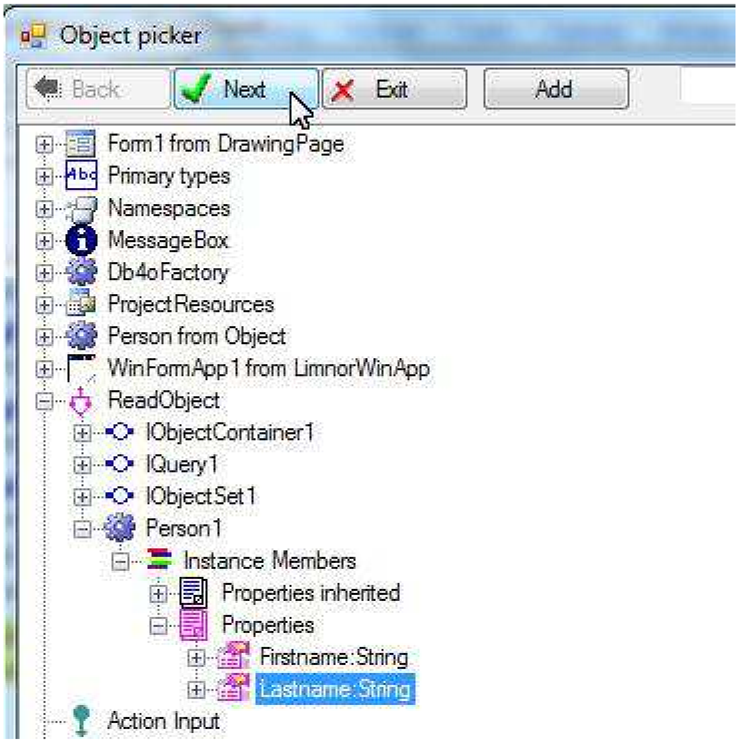
For "value", choose "Property":



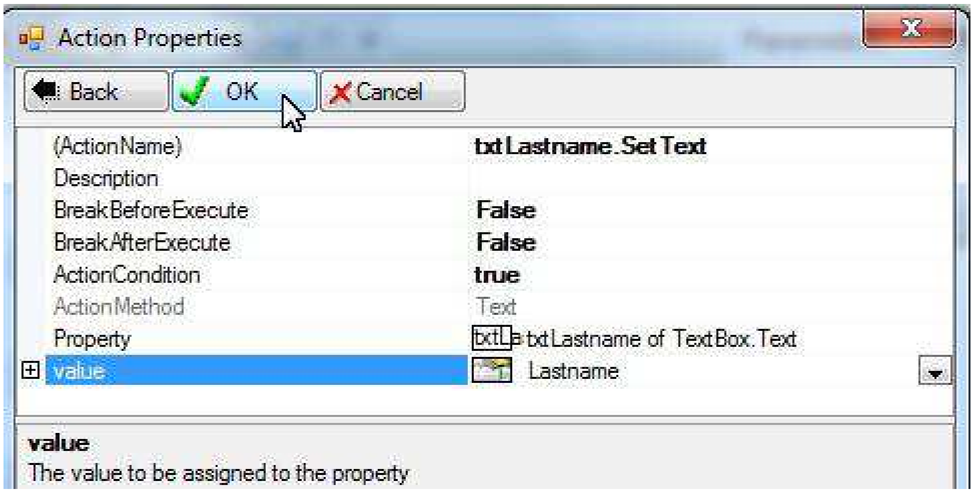
(ActionName)	txtLastname.SetText
Description	
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	Text
Property	txtLatxtLastname of TextBox.Text
value	

value
The value to be assigned to the property

Choose Lastname of Person1; click "Next":



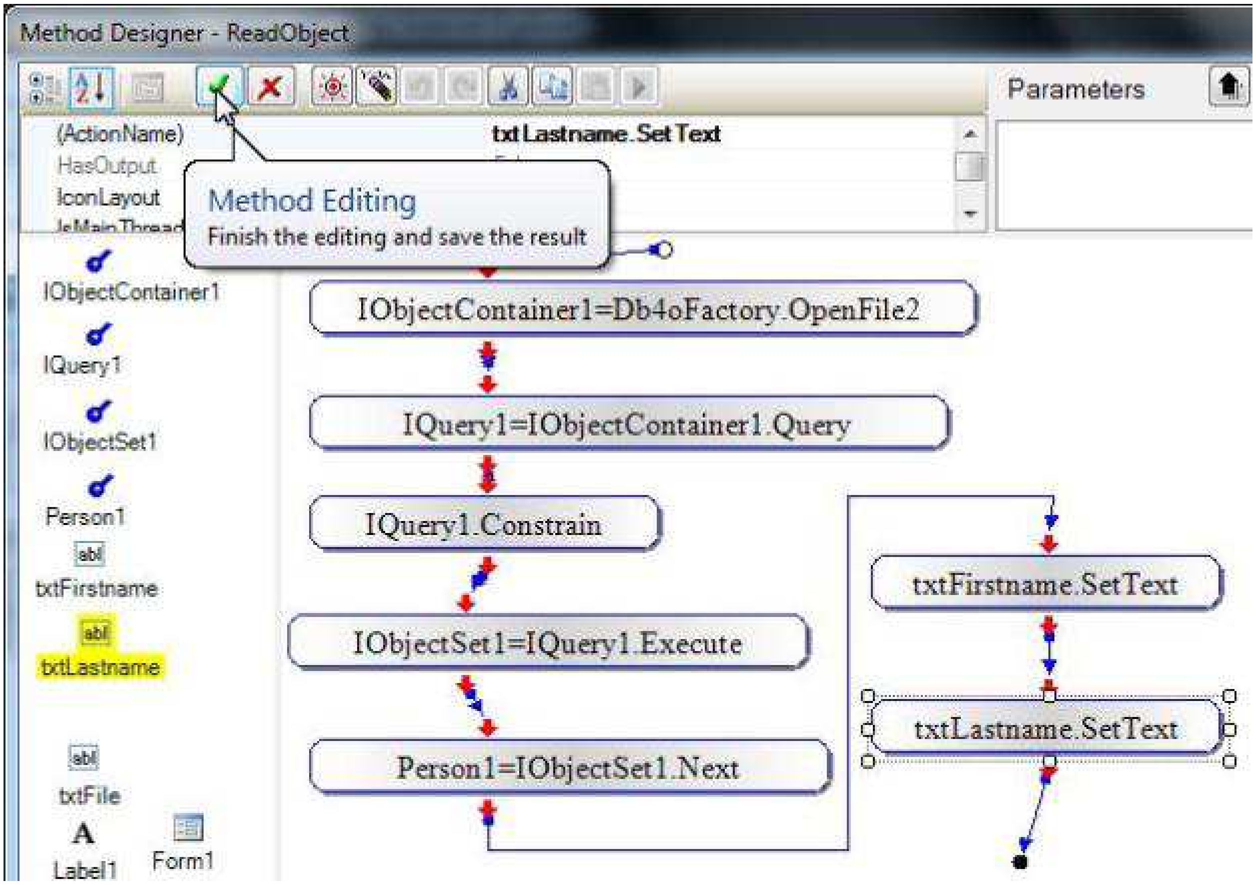
Click OK to finish creating this action:



(ActionName)	txtLastname.SetText
Description	
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	Text
Property	txtLastname of TextBox.Text
value	Lastname

value
The value to be assigned to the property

Link the actions together. We are done creating this method:

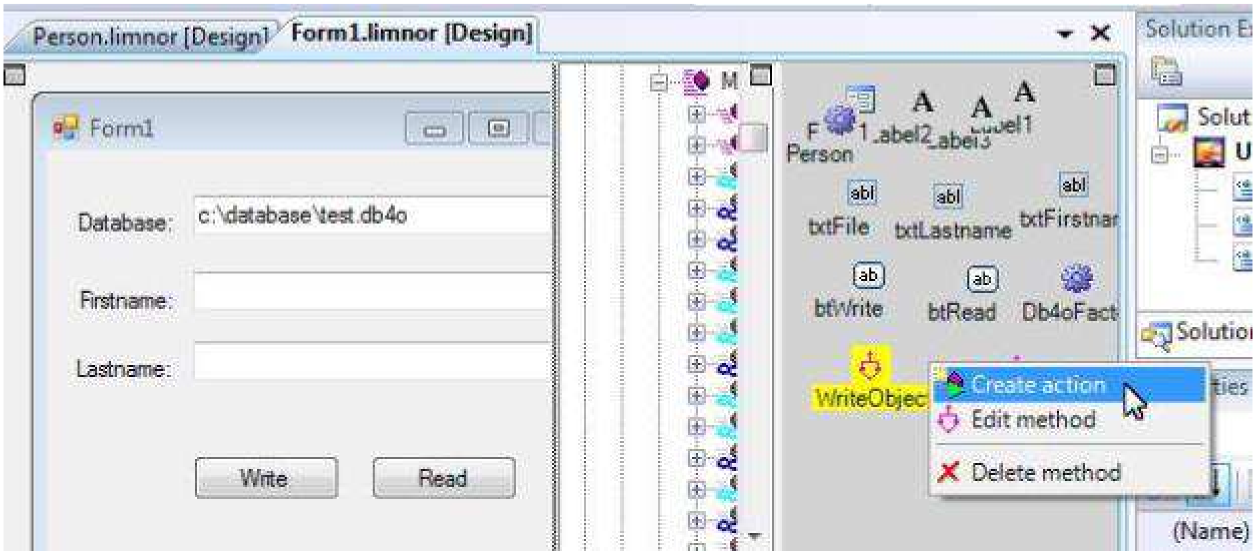


Execute Actions

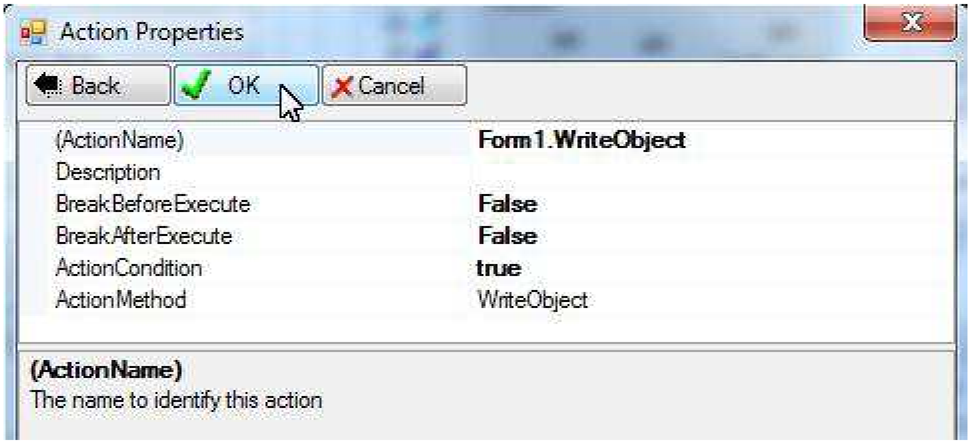
Create actions

We have created two methods, WriteObject and ReadObject. We need to create actions using them.

Right-click WriteObject; choose “Create action”:



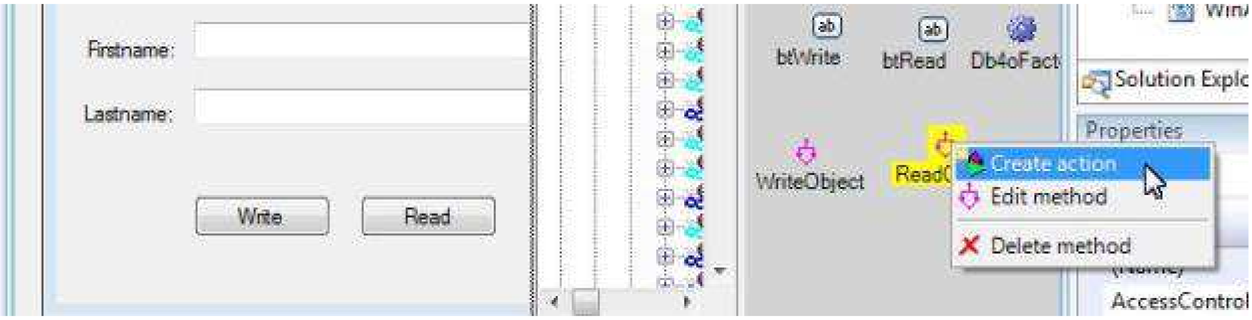
Click OK to finish creating this action:



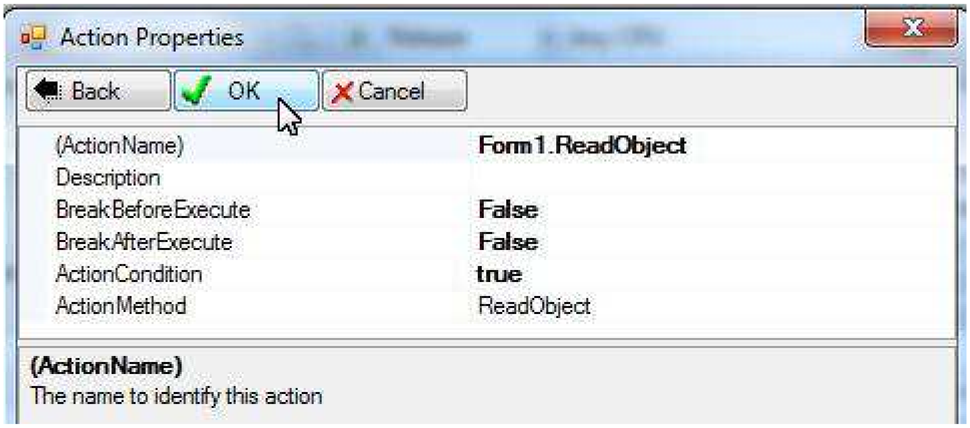
(ActionName)	Form1.WriteObject
Description	
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	WriteObject

(ActionName)
The name to identify this action

Right-click ReadObject; choose “Create action”:



Click OK to finish creating this action:

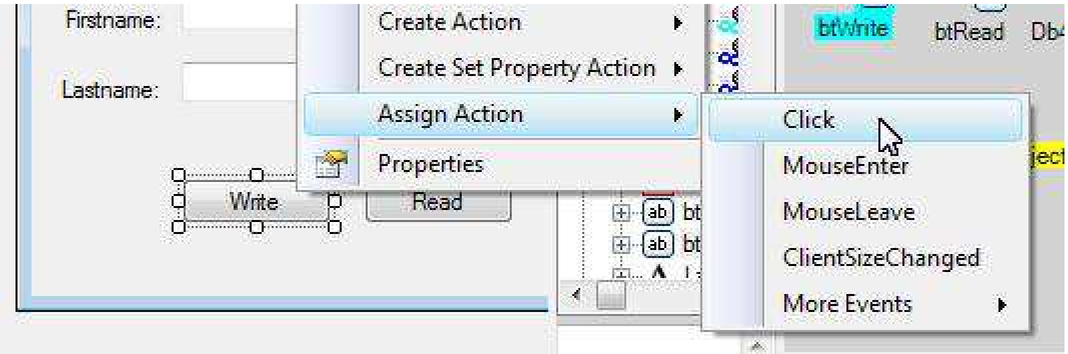


(ActionName)	Form1.ReadObject
Description	
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	ReadObject

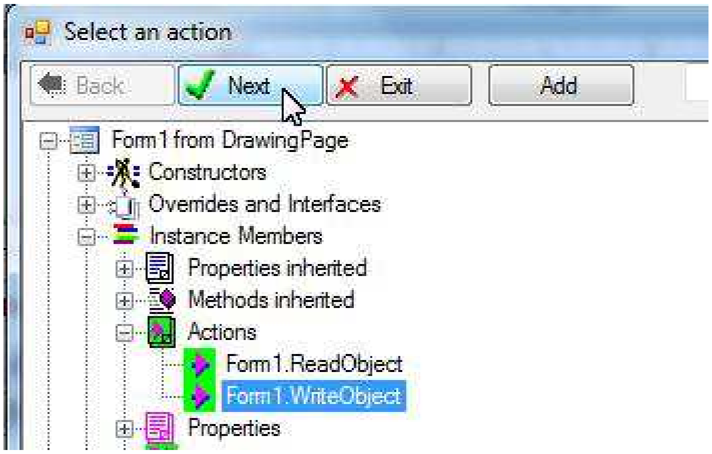
(ActionName)
The name to identify this action

Assign actions to events

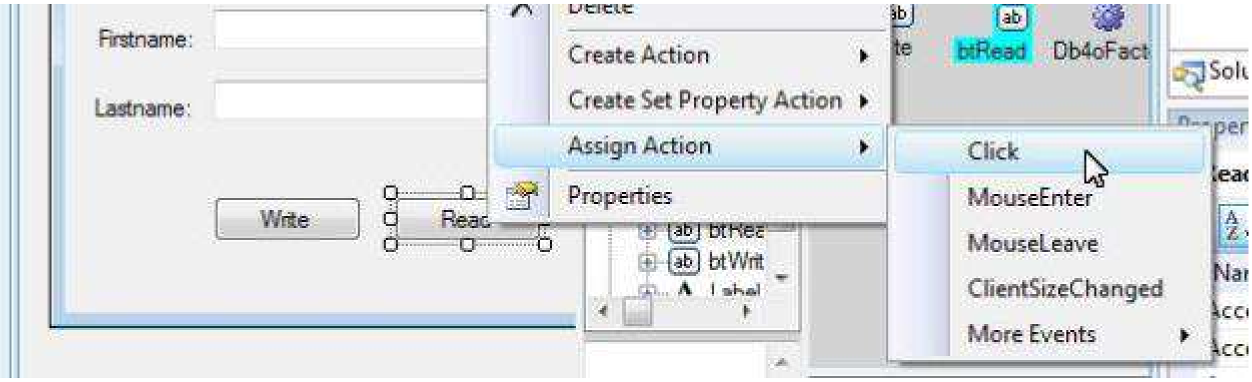
Right-click Write button; choose “Assign Action”; choose “Click”:



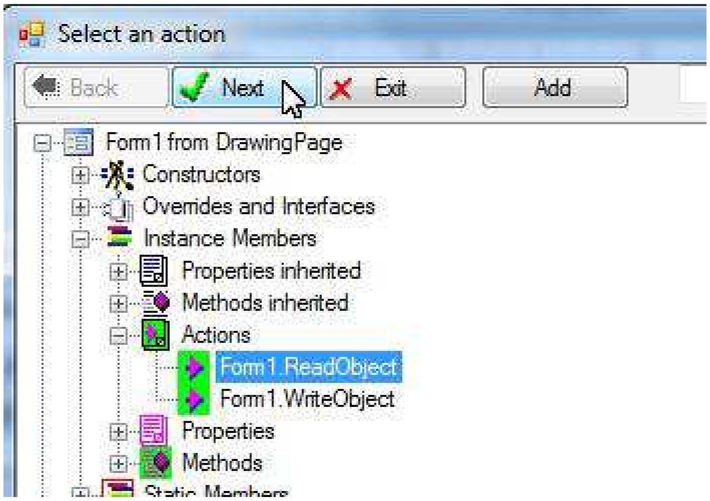
Choose WriteObject action; click “Next”:



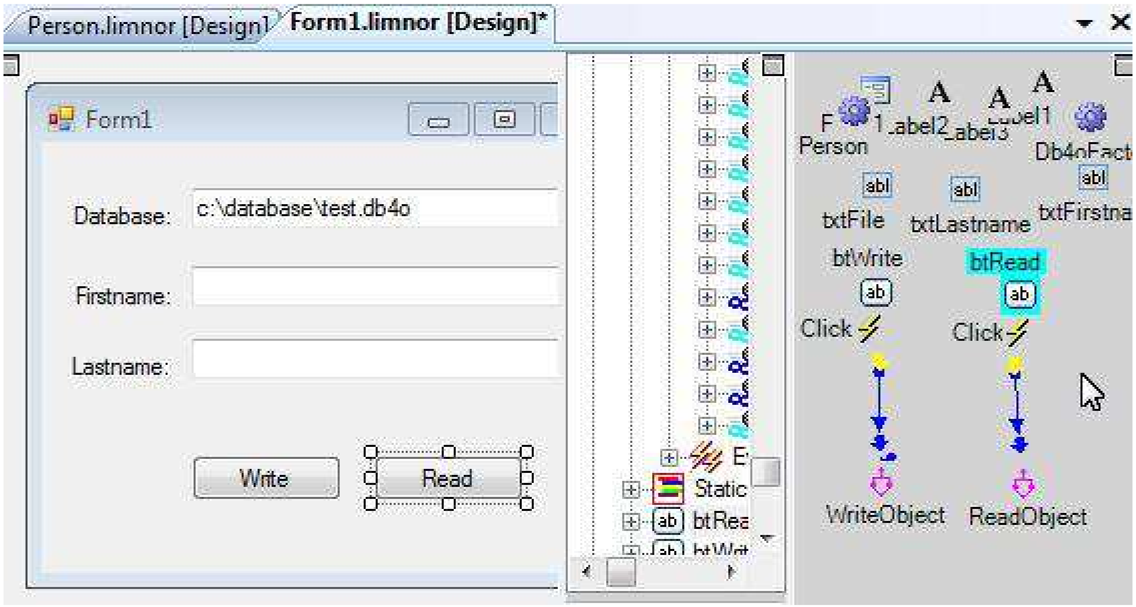
Right-click Read button; choose “Assign Action”; choose “Click”:



Select ReadObject action; click "Next":

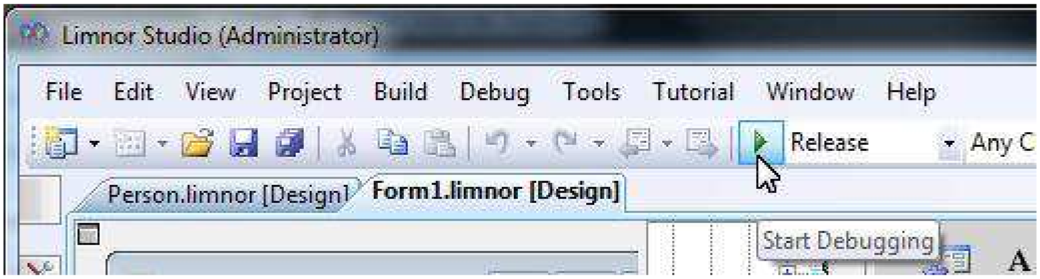


The actions are assigned to events:

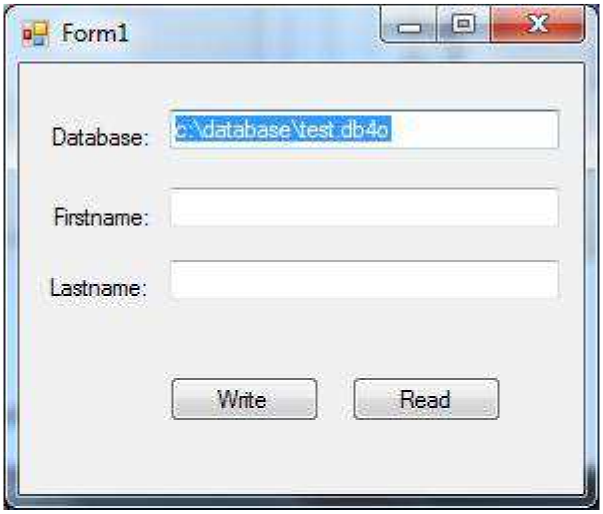


Test

Click the Run button to compile and run the application:



The Form1 appears after compiling:



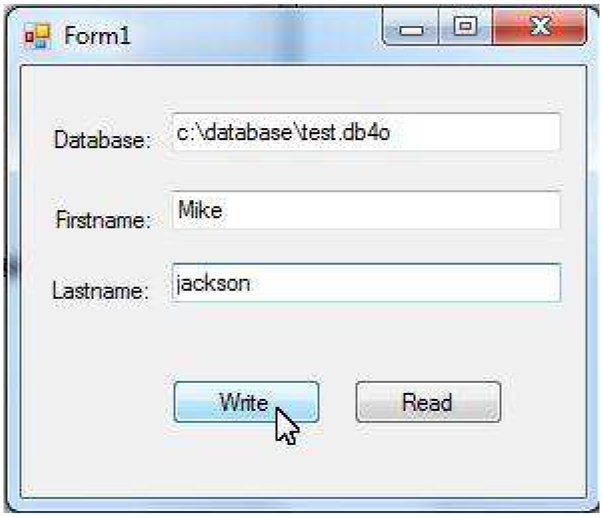
Database: c:\database\test.db4o

Firstname:

Lastname:

Test writing

Enter first name and last name. Click Write button:

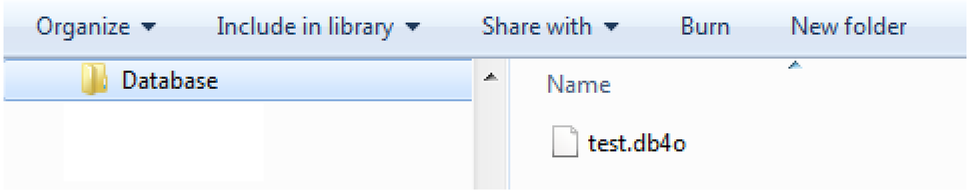


Database: c:\database\test.db4o

Firstname: Mike

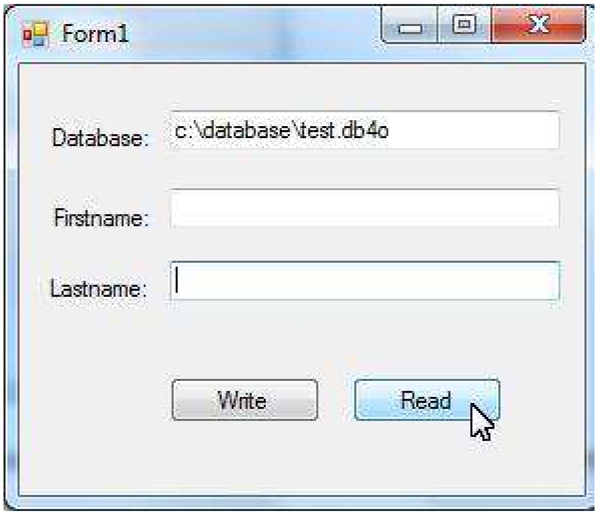
Lastname: jackson

Originally file c:\database\test.db4o does not exist. On clicking Write button, this file is generated:



Test reading

Before test reading, clear the text boxes. Then click Read button:

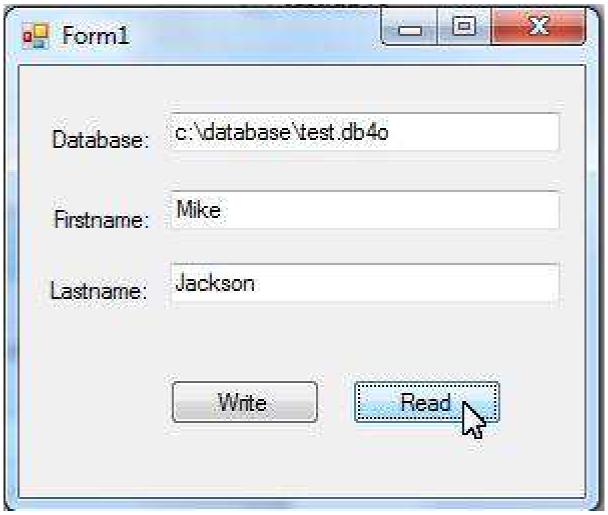


Database: c:\database\test.db4o

Firstname:

Lastname:

We can see that the first name and last name appear in the text boxes:



Database: c:\database\test.db4o

Firstname: Mike

Lastname: Jackson

Feedback

Please send your feedback to support@limnor.com