How to Print a Control

Contents

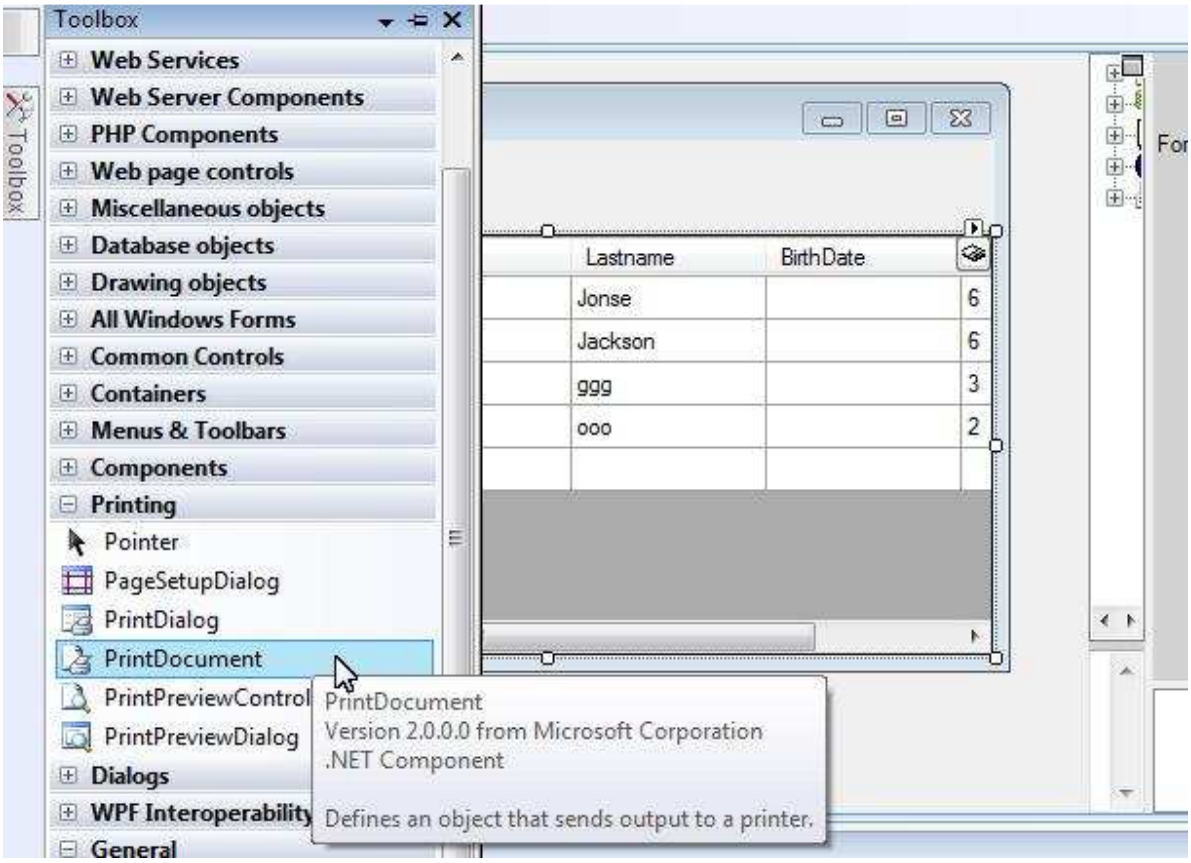
Use Generic Printing process

This sample shows printing a control using an EasyGrid as a sample. Note that the process described in this section can be applied to print any controls. But the result may not be acceptable. EasyGrid has better printing methods, which are described in the next section. If an object provides its own printing methods then you should try to use those methods first. If an object does not provide its own printing methods then you may use the process described below to do printing.

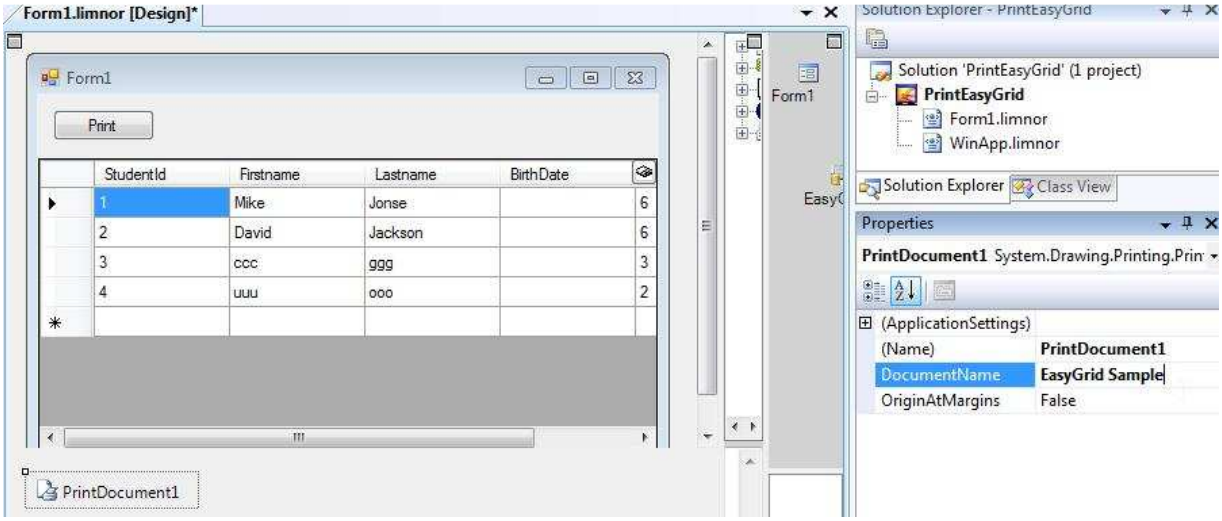
Create an EasyGrid and link it to a database to get data. For using databases, see http://www.limnor.com/support/Limnor%20Studio%20-%20User%20Guide%20-%20Part%20VI.pdf

Use PrintDocument

Drop a PrintDocument to the form to do the printing:

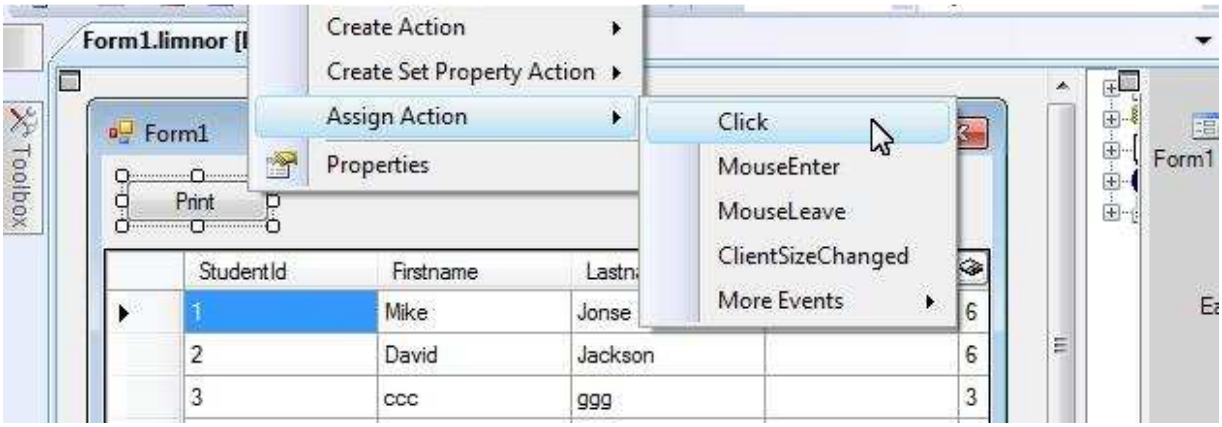


Set DocumentName to identify the print job:

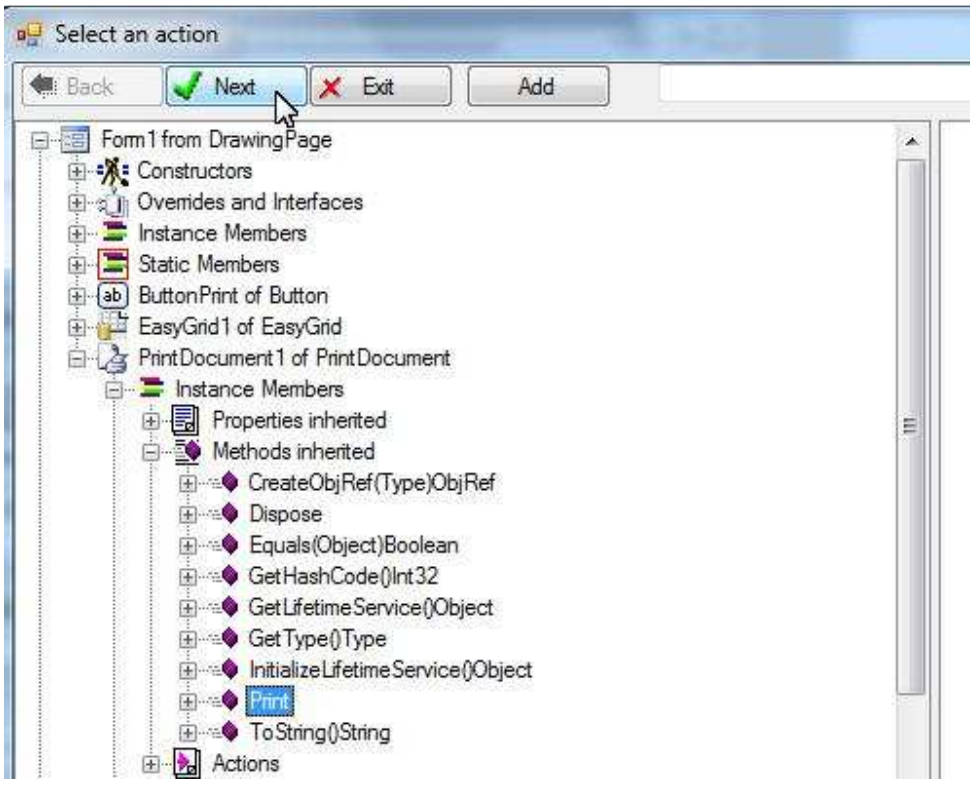


Invoke Printing

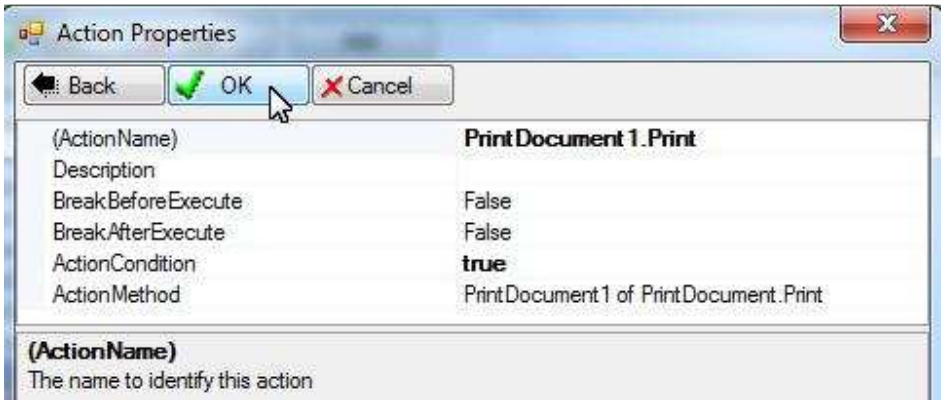
We use a button to initiate the printing. Right-click the button; choose “Assign action”; choose “Click” event:



Select the Print method of the PrintDocument component:

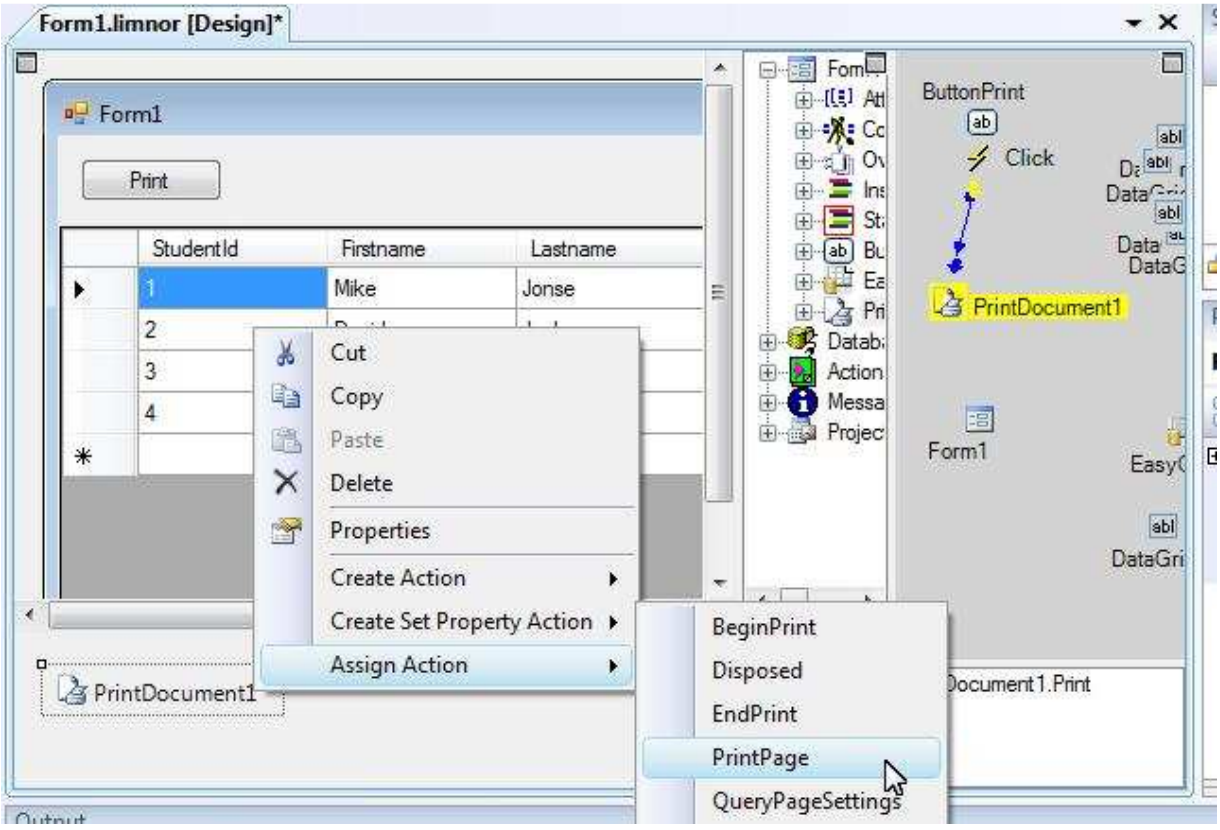


Click OK to create the action and assign the action to the button:

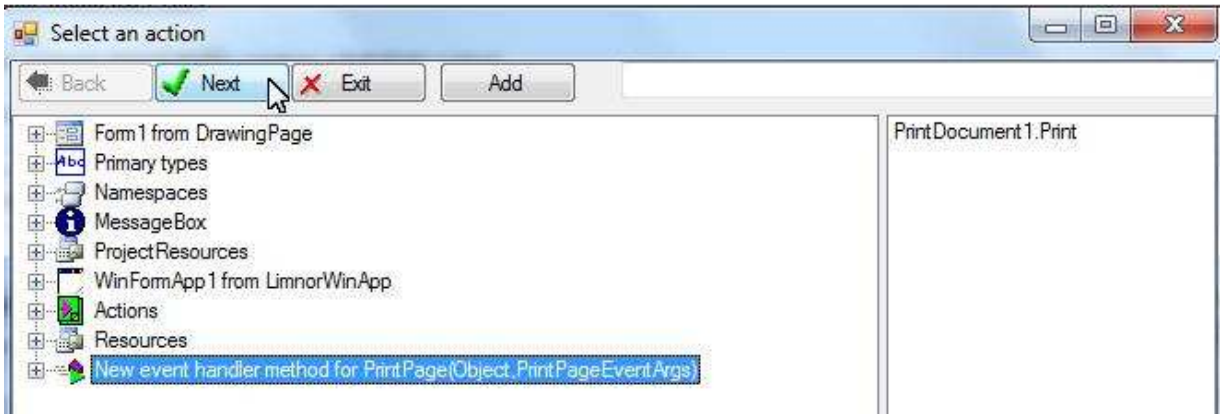


Doing actual printing

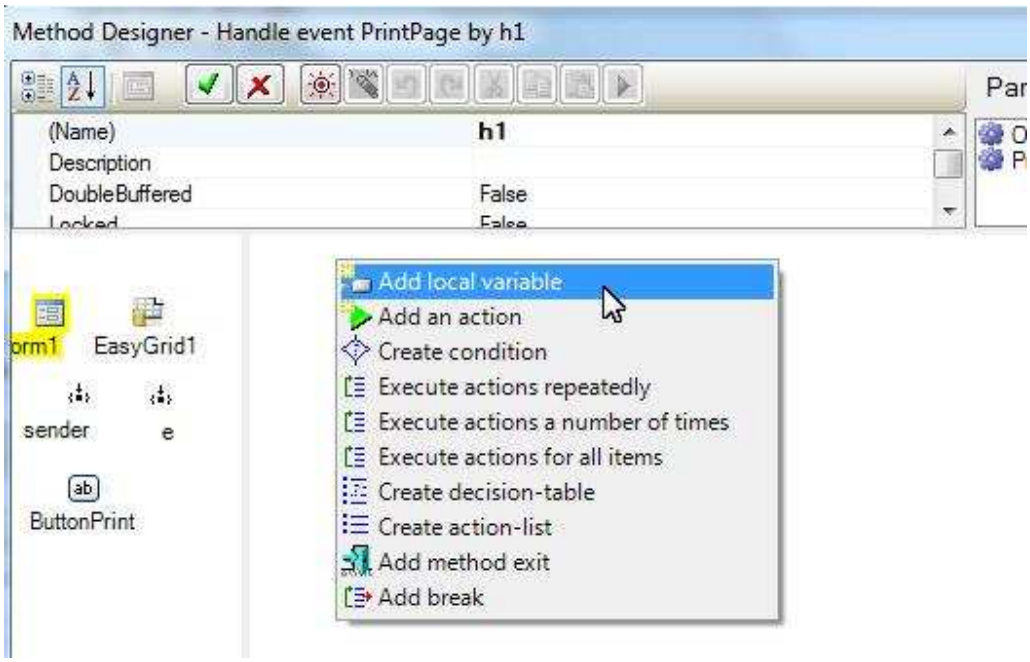
Handle PrintPage event to do the actual printing. Right-click the PrintDocument component; choose "Assign action"; choose "PrintPage" event:

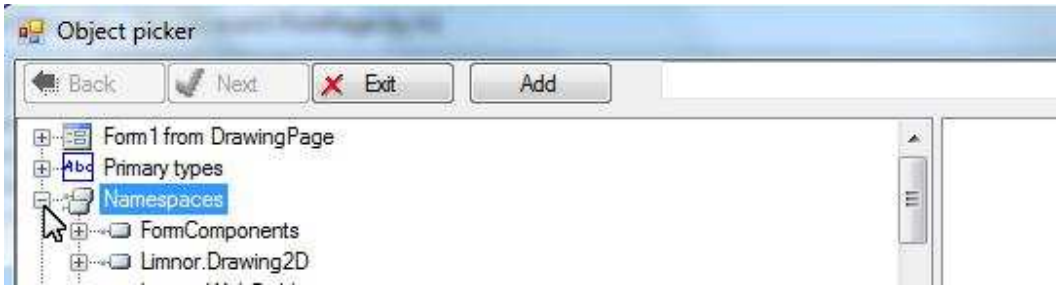


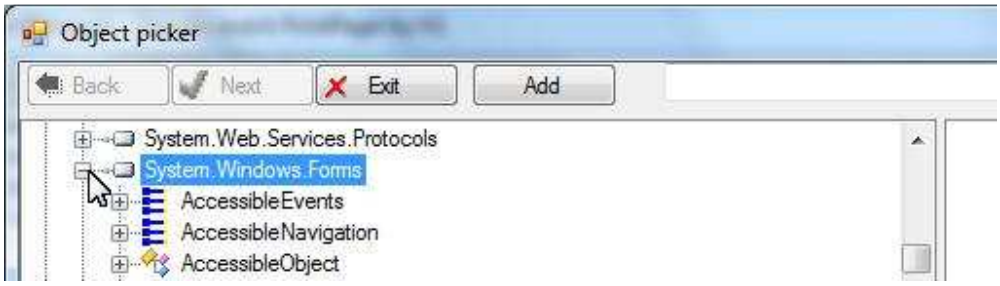
Choose "New event handler …" to create a new event handler method:

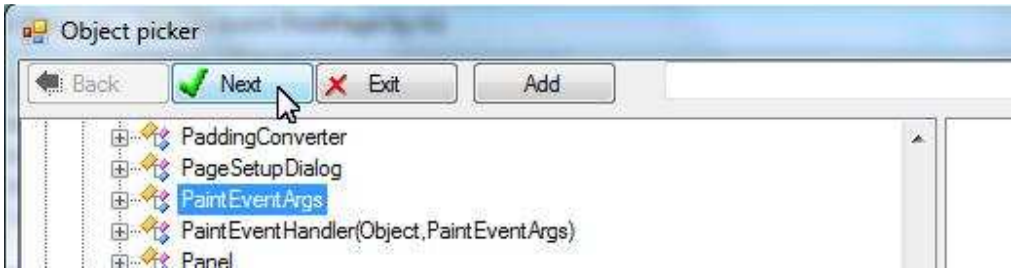


Create a PaintEventArgs variable:



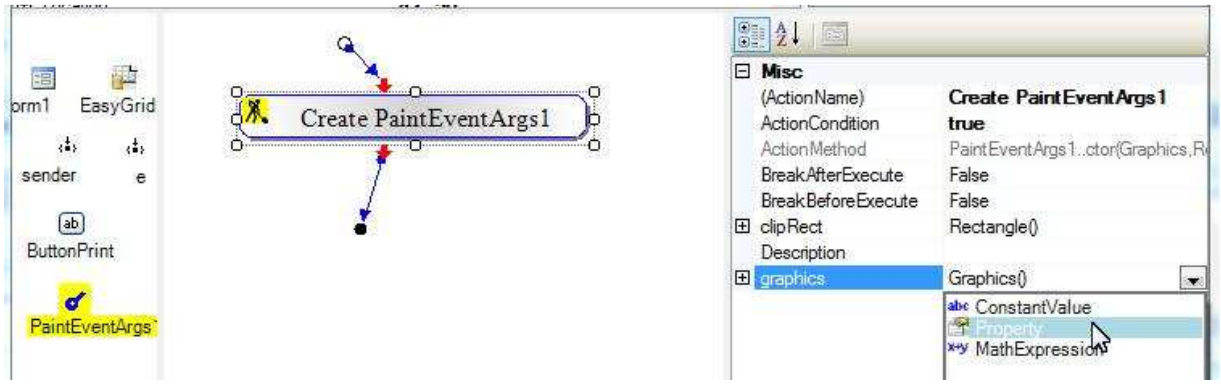




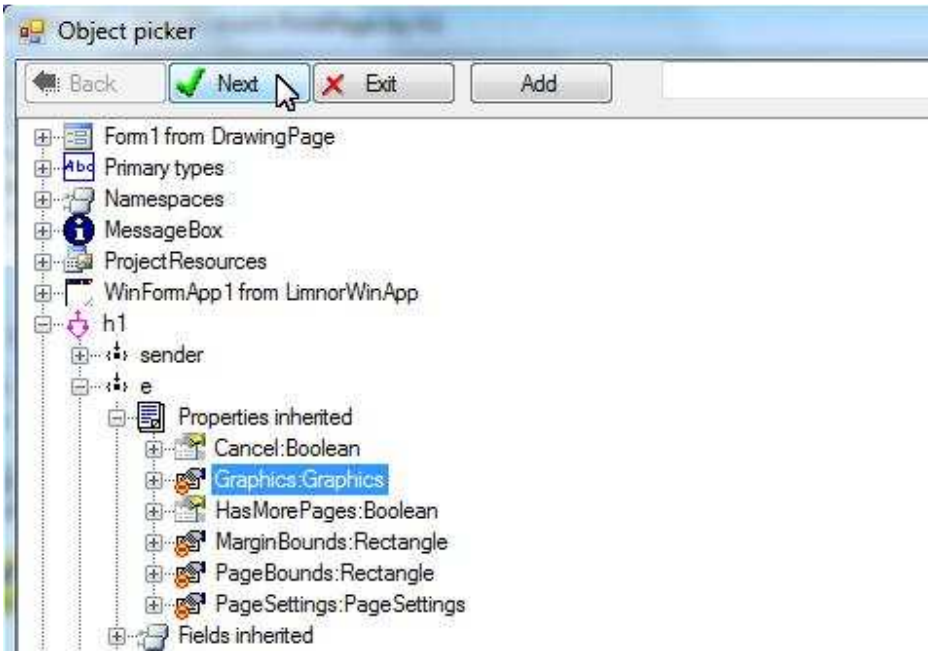


Choose the constructor:

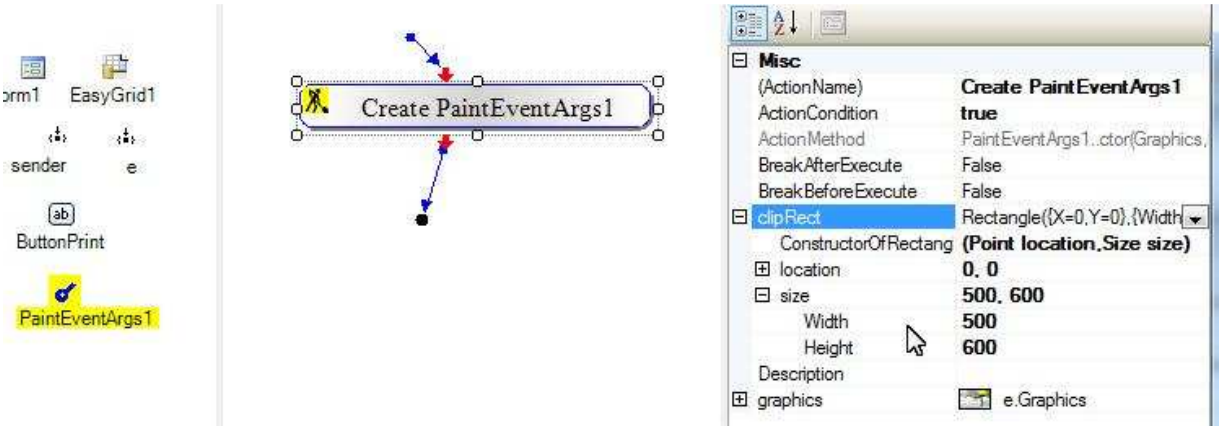
The new variable appears in the Variable Pane. An action appears in the Action Pane. The action uses the constructor to create the variable. We may set the action parameters. For graphics, choose Property because we will get its value from the event parameter:



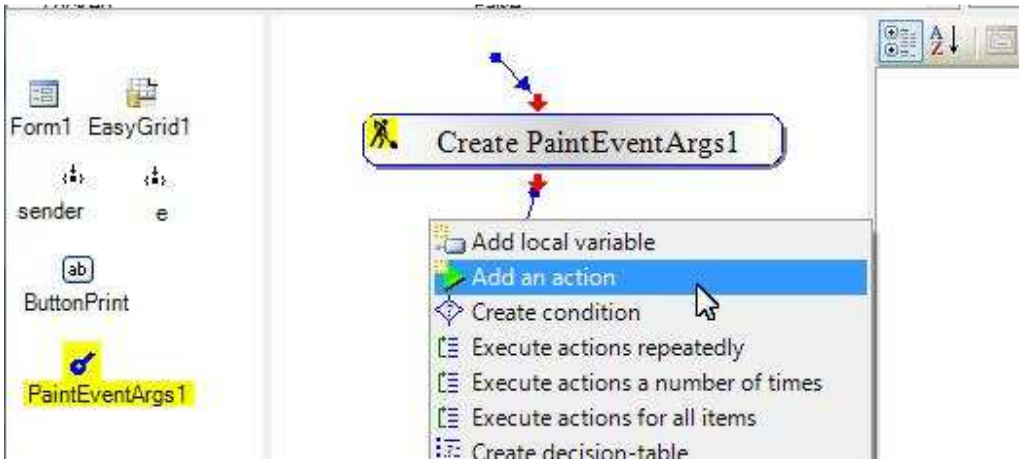
Select the Graphics property of the event parameter:



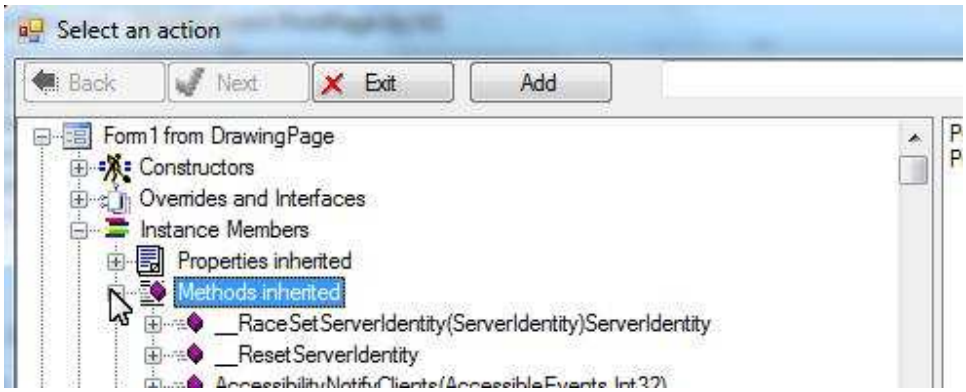
Parameter clipRect is the printing rectangle on the page:



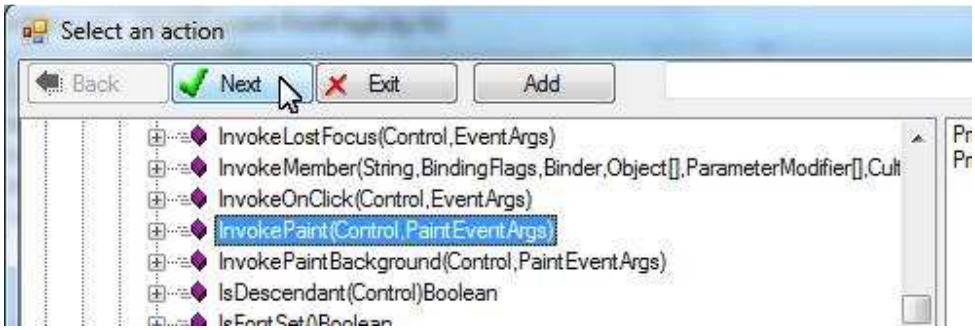
Create an InvokePaint action to paint the control. Right-click the Action Pane; choose “Add an action”:



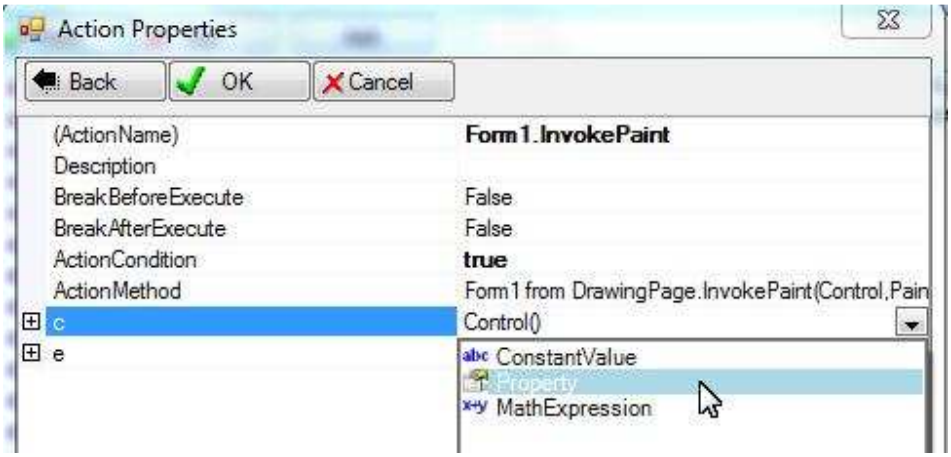
Expand the Methods of the form:



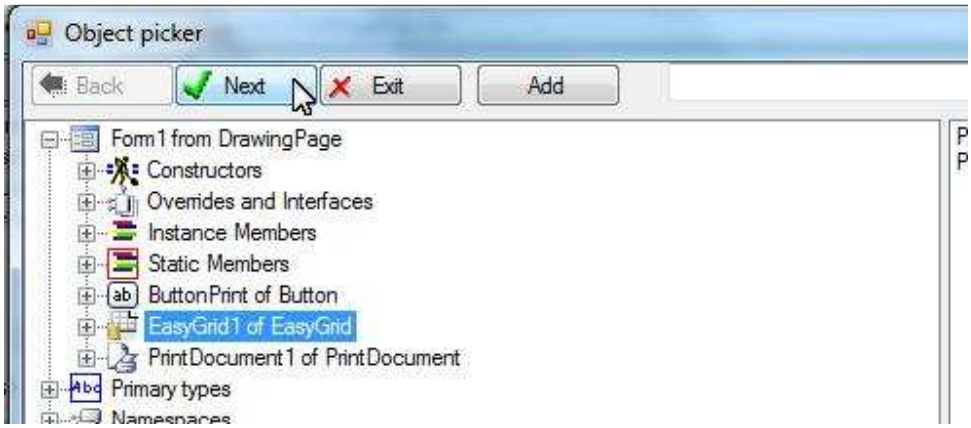
Select the InvokePaint method:



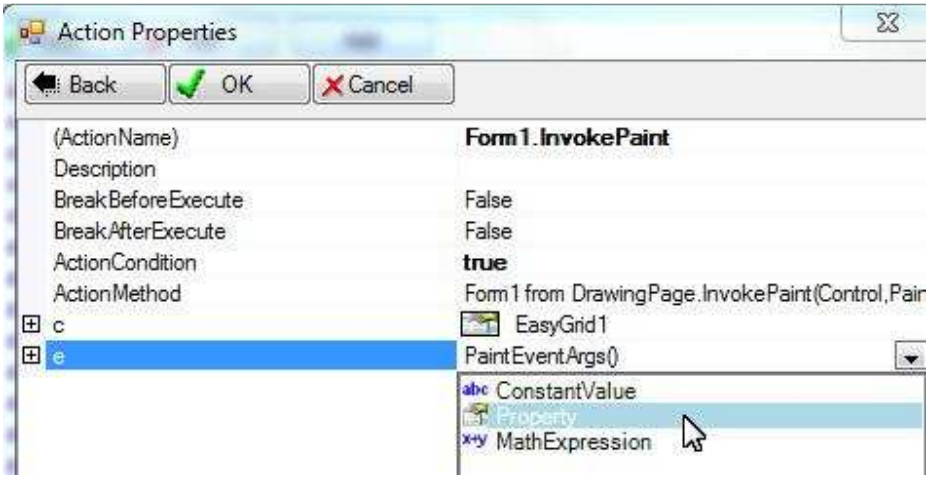
Parameter c is the control to be printed. Select Property so that we may choose the EasyGrid:



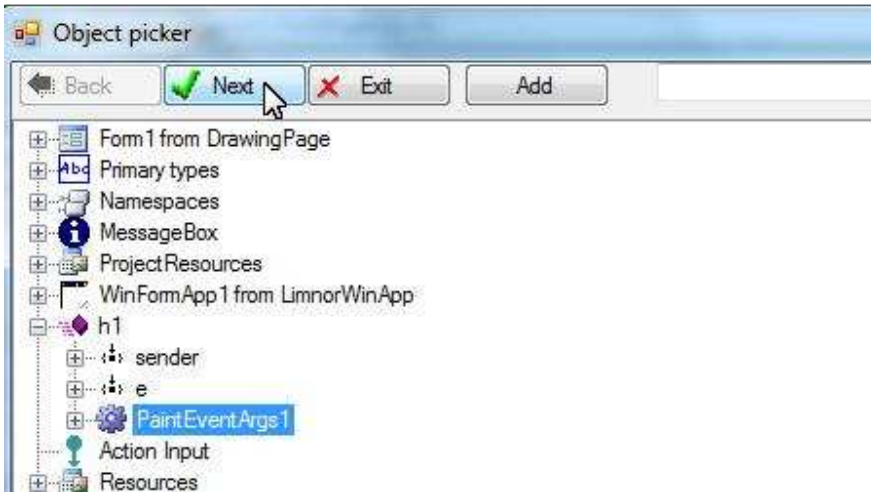
Select the EasyGrid:



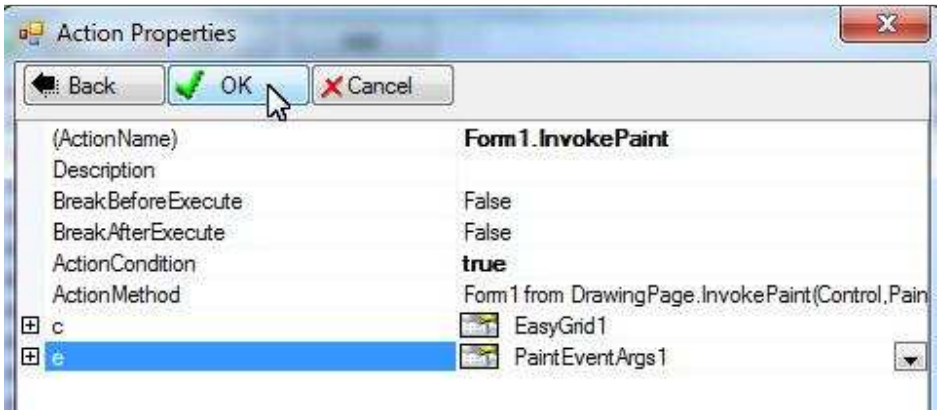
Parameter e is the drawing parameters. Select Property so that we may use the PaintEventArgs variable:



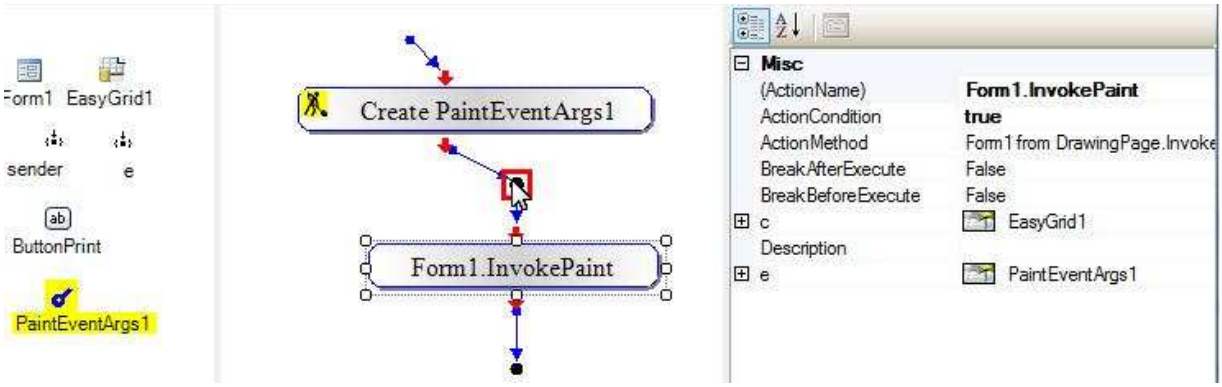
Select the PaintEventArgs variable:

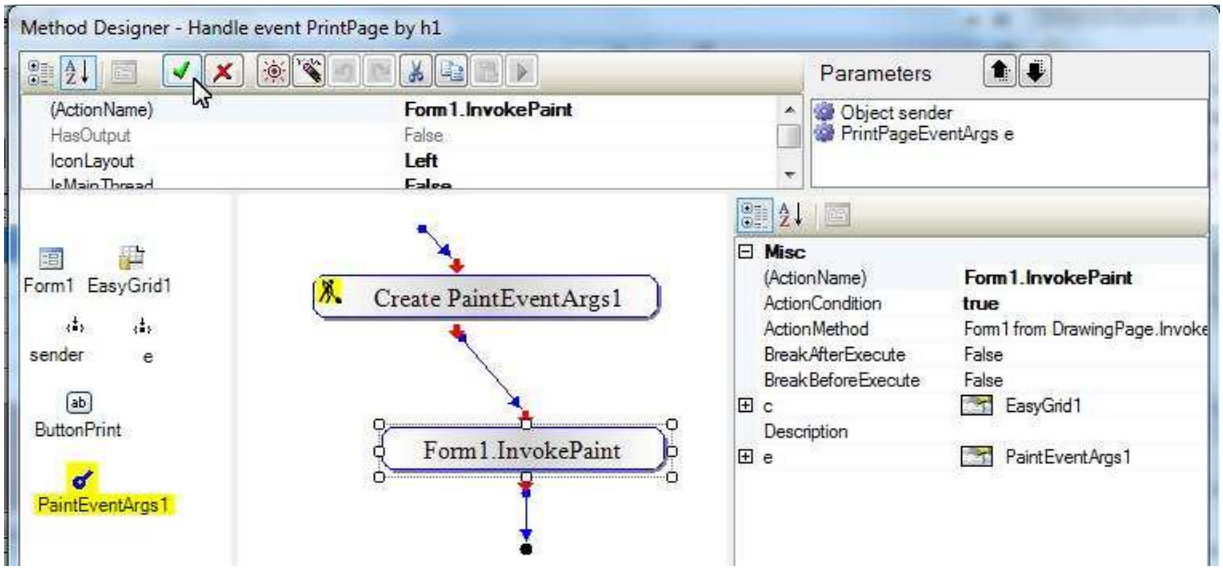


Click OK:



Link the two actions together:



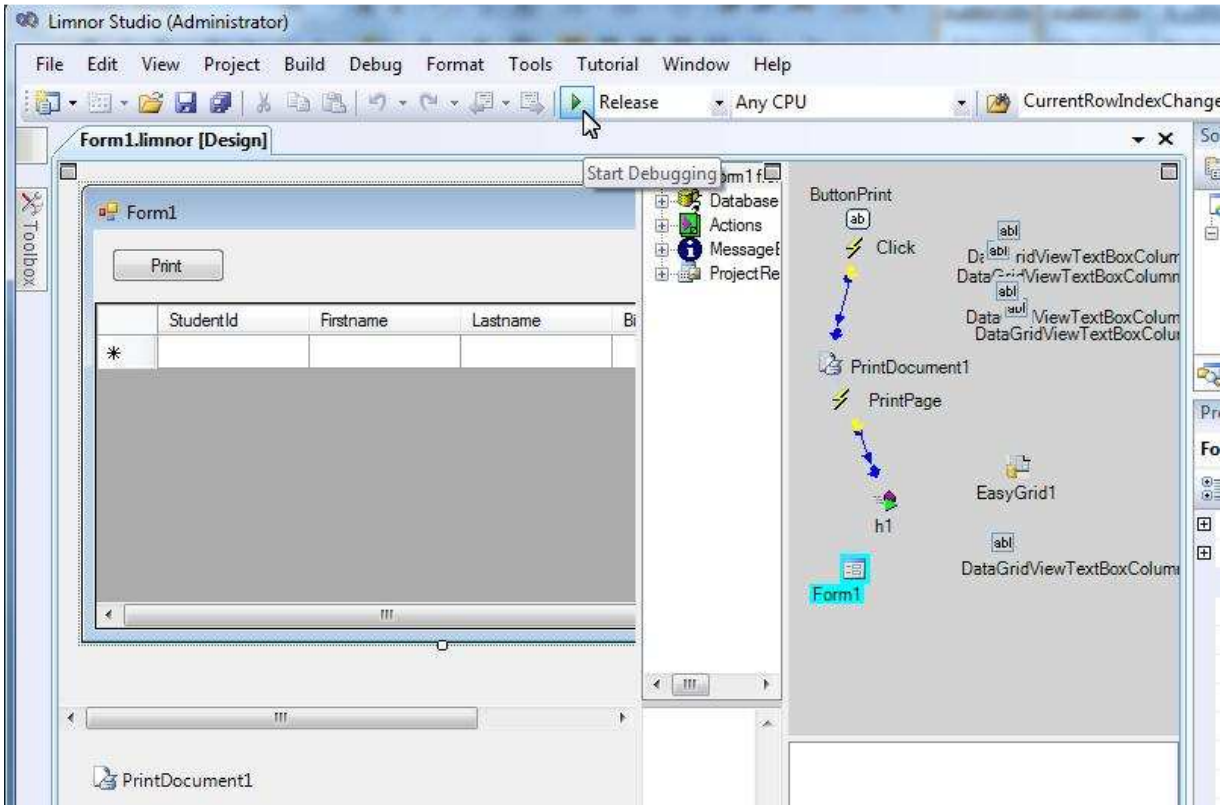


Taking over drawing

Note that in the above example, we do not do our own painting for the printing. We use a method, InvokePaint, provided by Microsoft .Net Framework to do the painting for the printing. This built-in method may not give us the ideal printing result. To take over the painting, we may use the Graphics property of the event parameter e to do various drawings. For all the drawing capabilities provided by the Graphics, see http://msdn.microsoft.com/en-us/library/system.drawing.graphics_methods(v=vs.90)

Test

We are done creating this sample. Click the Run button to compile and run the application.



The print out is weird:

The above result is generated by the InvokePaint method provided by the Microsoft .Net Framework. You may take over the printing by providing your own drawing actions to replace the InvokePaint action.

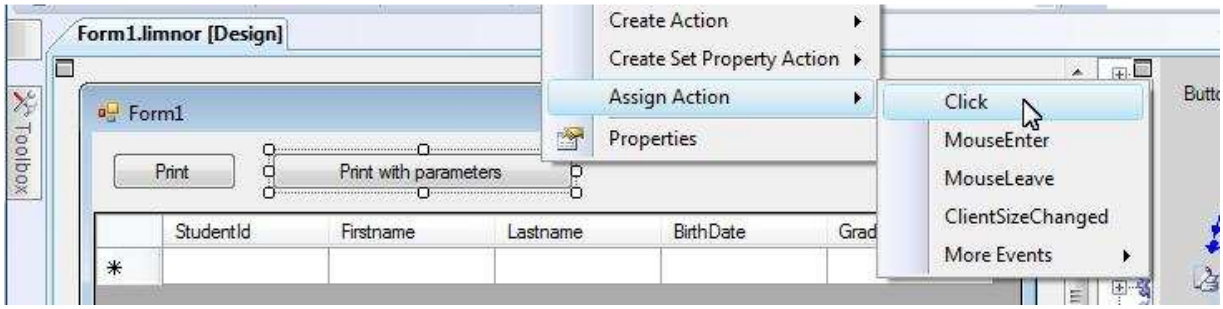
Use Built-in Printing Methods

This above sample demonstrates a generic printing process which can be used for printing all kinds of controls. But the result may not be acceptable by your scenarios. Some controls take over the printing process to give better print result.

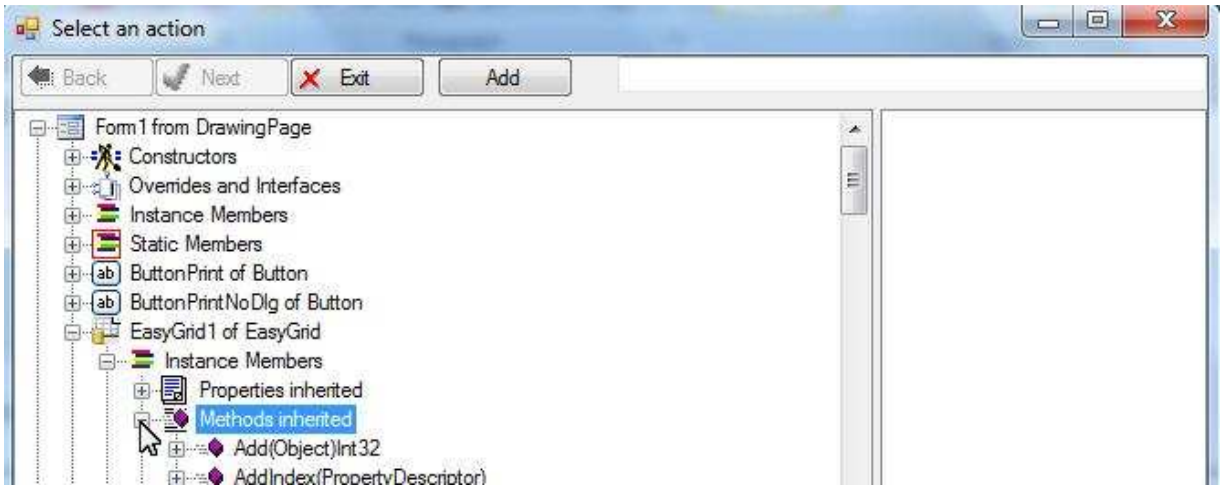
EasyGrid provides 2 printing methods. We'll demonstrate the use of both methods.

Print Action with preferences

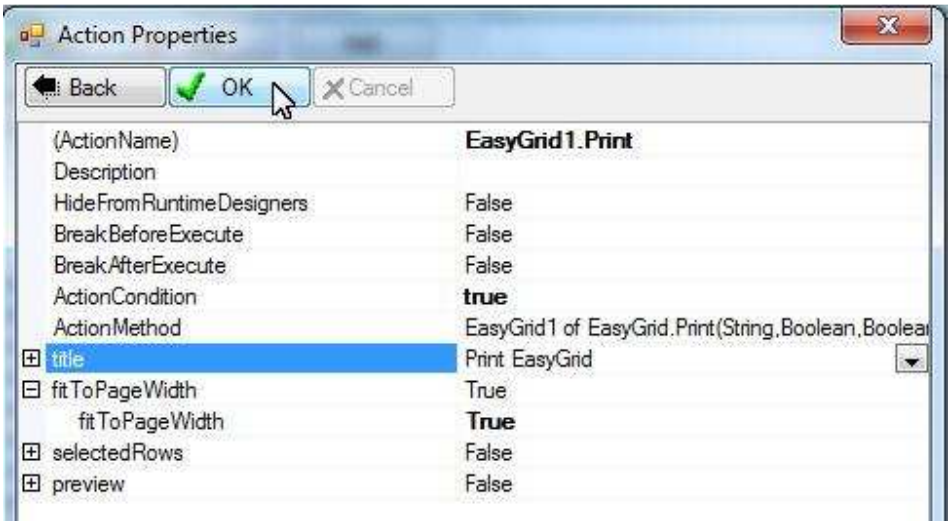
We use a button to invoke a print action:



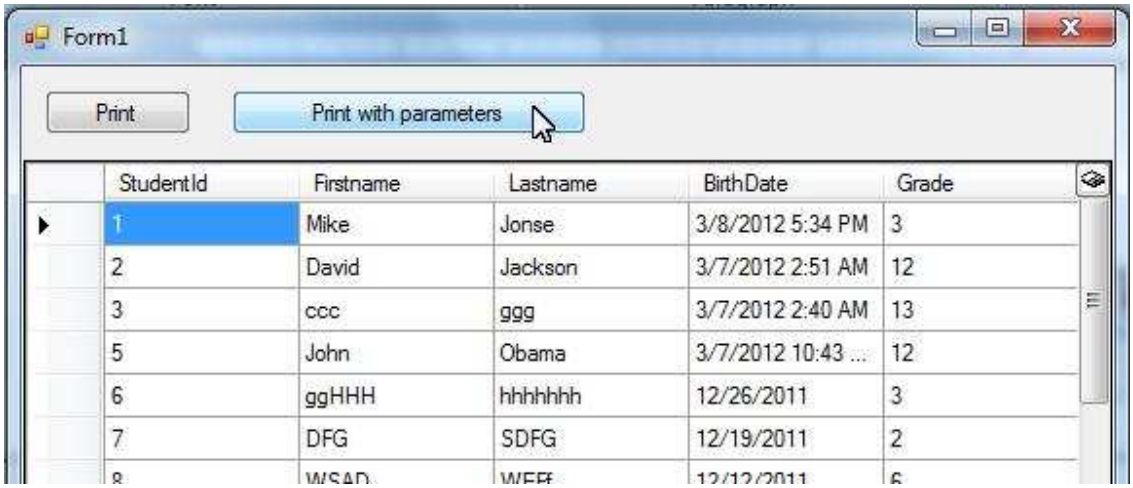
Select a Print method which accepts print parameters:



Give the print parameters for the action:



Compile and run the application. Click the button to do printing:



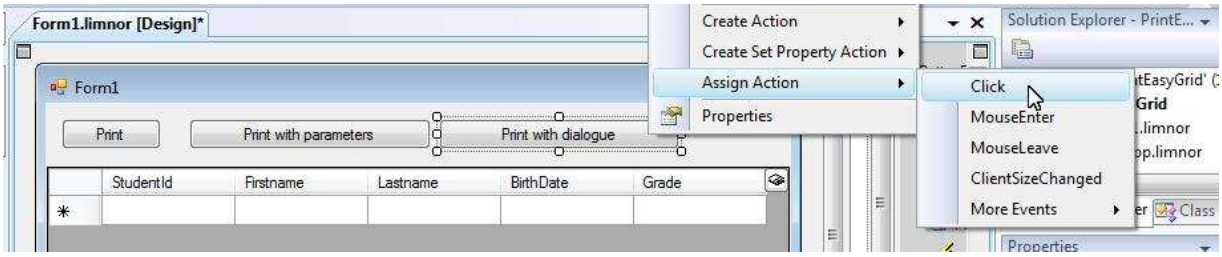
We get a better print out:

Print EasyGrid Tuesday, July 31, 2012 11:20 AM

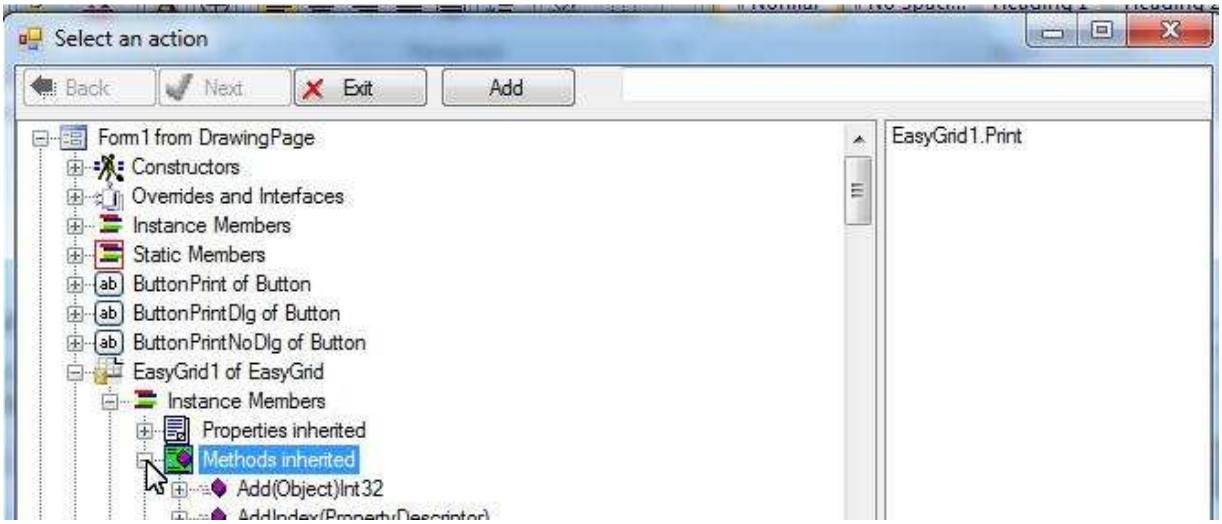
StudentId	Firstname	Lastname	BirthDate	Grade
1	Mike	Jonse	3/8/2012 5:34:58 PM	3
2	David	Jackson	3/7/2012 2:51:55 AM	12
3	ccc	ggg	3/7/2012 2:40:46 AM	13
5	John	Obama	3/7/2012 10:43:23 AM	12
6	ggHHH	hhhhhhh	12/26/2011 12:00:00 AM	3
7	DFG	SDFG	12/19/2011 12:00:00 AM	2
8	WSAD	WEFf	12/12/2011 12:00:00 AM	6
9	ssAAAX			

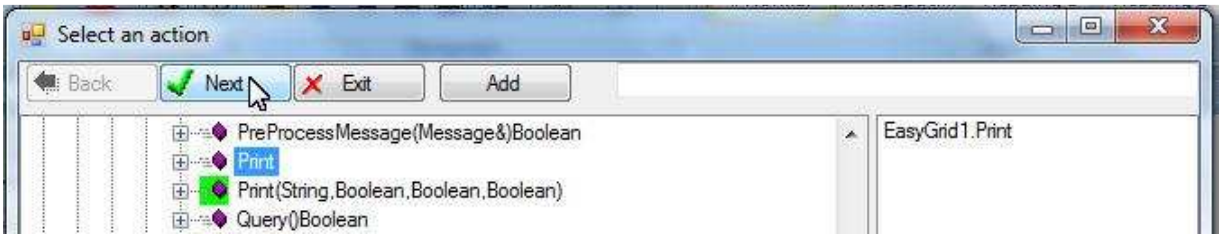
Use Print Dialogue

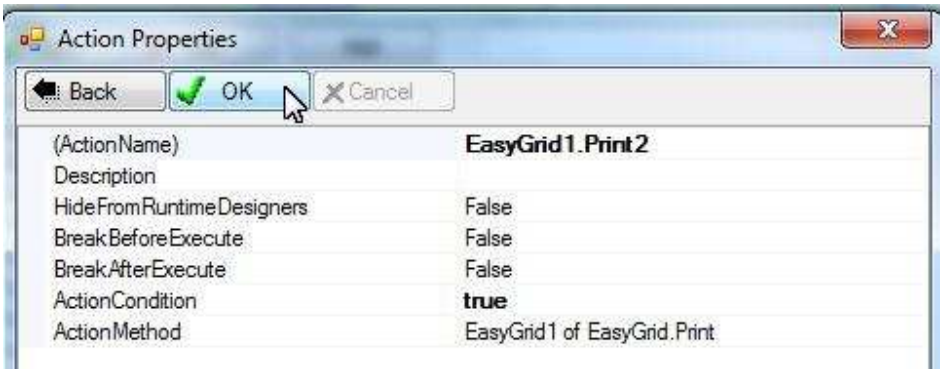
We use another button to invoke a print action:



Select the Print method which does not use print parameters:

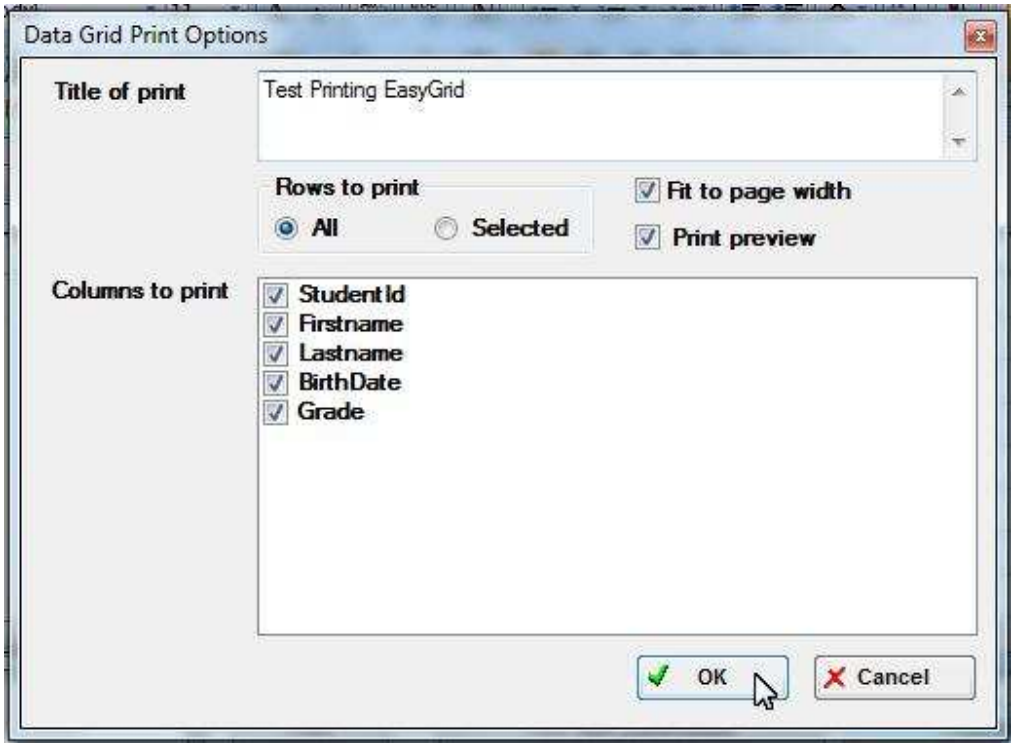






Compile and run the application. Click the button to invoke the printing:

A dialogue box appears to collect print parameters:



The print preview window appears:

The print out:

Test Printing EasyGrid — Tuesday, July 31, 2012 11:31 AM

StudentId	Firstname	Lastname	BirthDate	Grade
1	Mike	Jonse	3/8/2012 5:34:58 PM	3
2	David	Jackson	3/7/2012 2:51:55 AM	12
3	ccc	ggg	3/7/2012 2:40:46 AM	13
5	John	Obama	3/7/2012 10:43:23 AM	12
6	ggHHH	hhhhhhh	12/26/2011 12:00:00 AM	3
7	DFG	SDFG	12/19/2011 12:00:00 AM	2
8	WSAD	WEFf	12/12/2011 12:00:00	6