

# Set Row Color

---

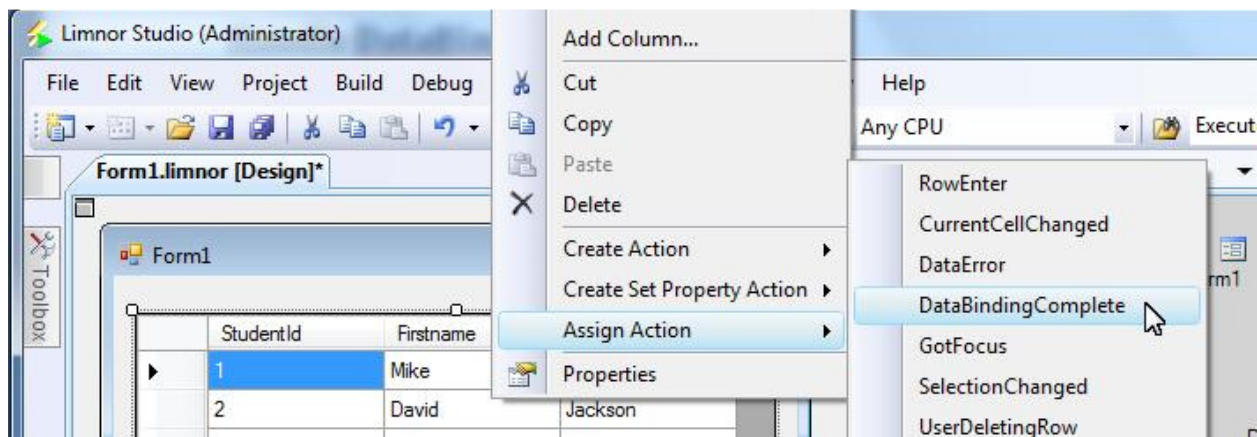
## Contents

Handle DataBindingComplete Event .....	1
Go through each row.....	2
Set Background Color for a row .....	3
Conditional Color Change .....	6

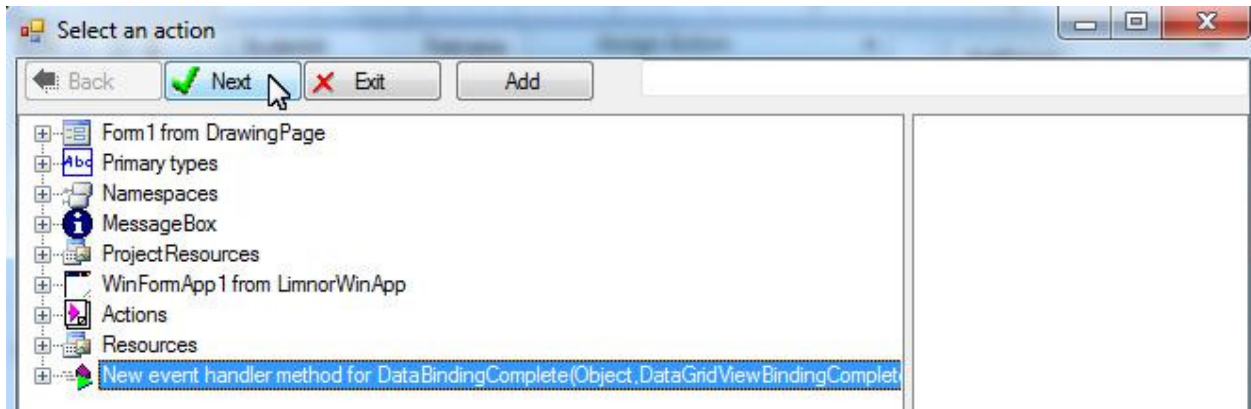
## Handle DataBindingComplete Event

The properties of each row of an EasyGrid can be changed. This sample does it at the event of DataBindingComplete.

Right-click on an EasyGrid; choose “Assign Action”; choose “DataBindingComplete”:



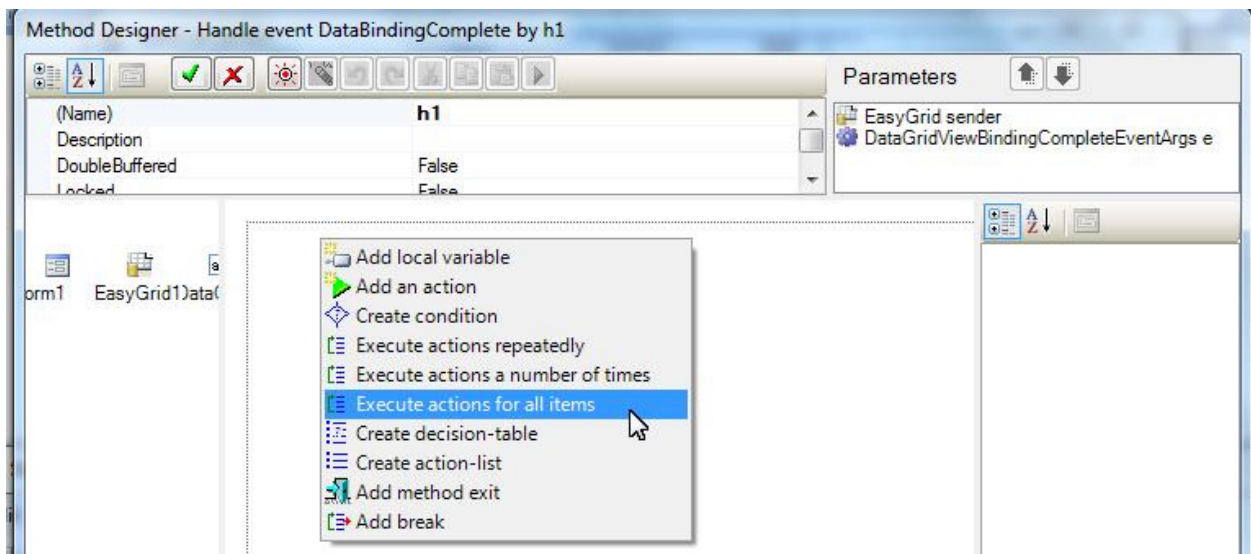
Select “New event handler method for DataBindingComplete “:



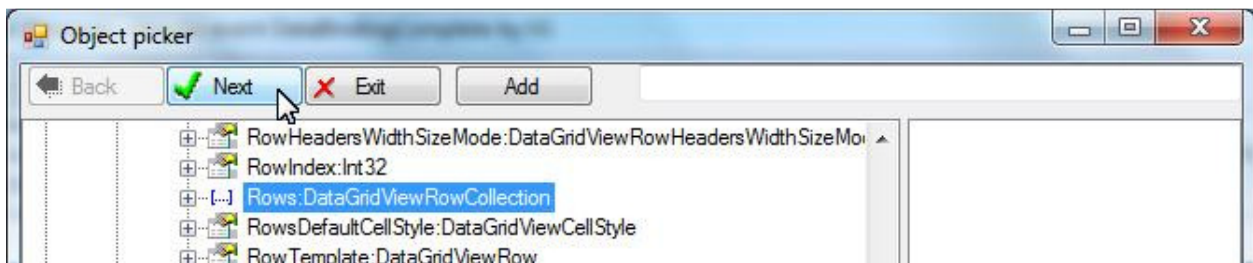
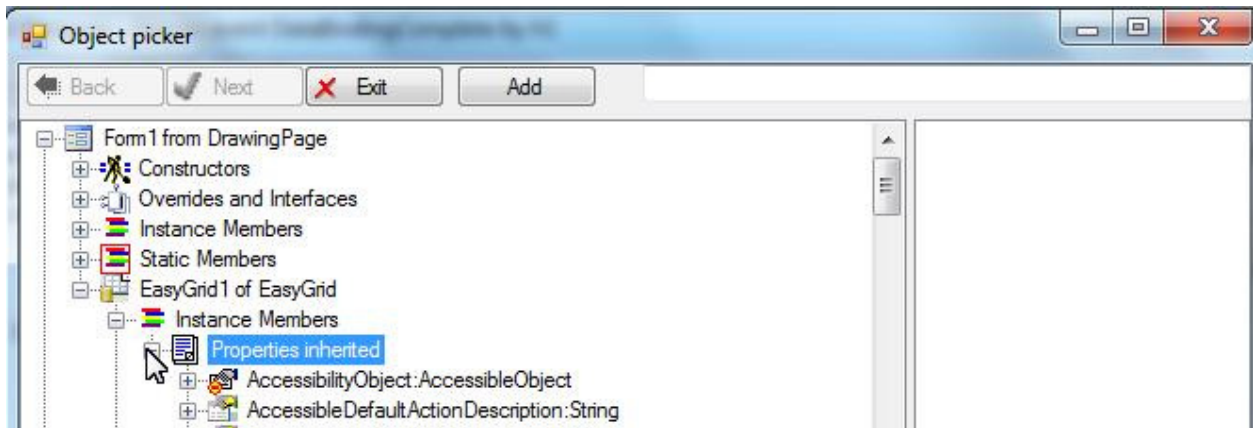
## Go through each row

A Method Editor appears for constructing the event handler method.

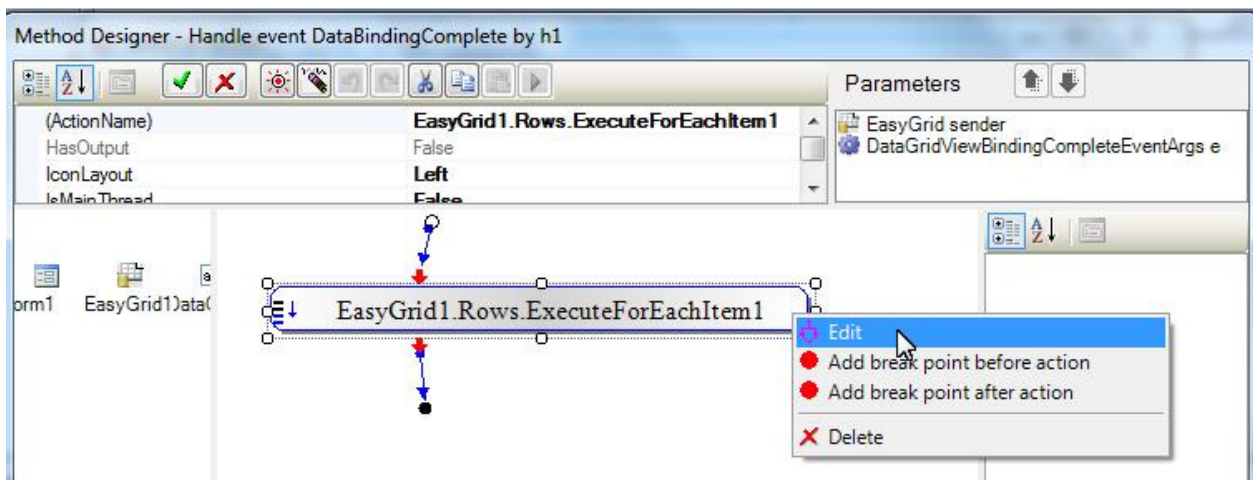
Right-click the Action Pane; choose "Execute actions for all items":



Select the Rows property of the EasyGrid:

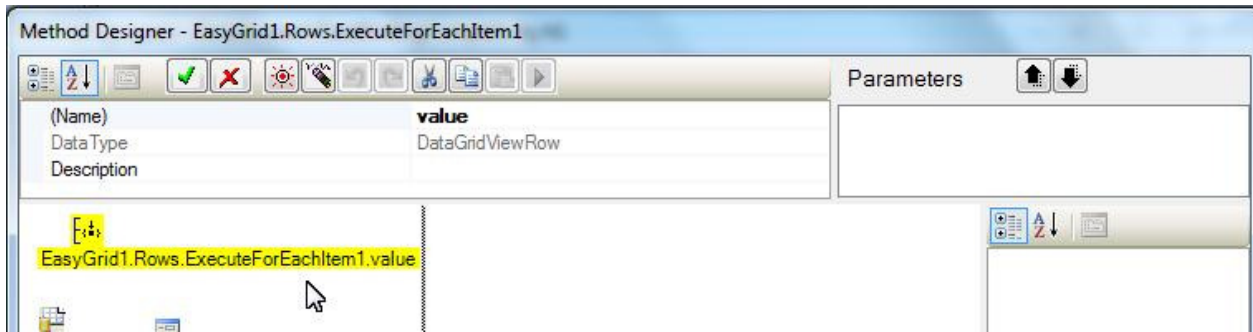


An action for going through each row is created. Right-click it and choose “Edit” to specify the actions for each row:



## Set Background Color for a row

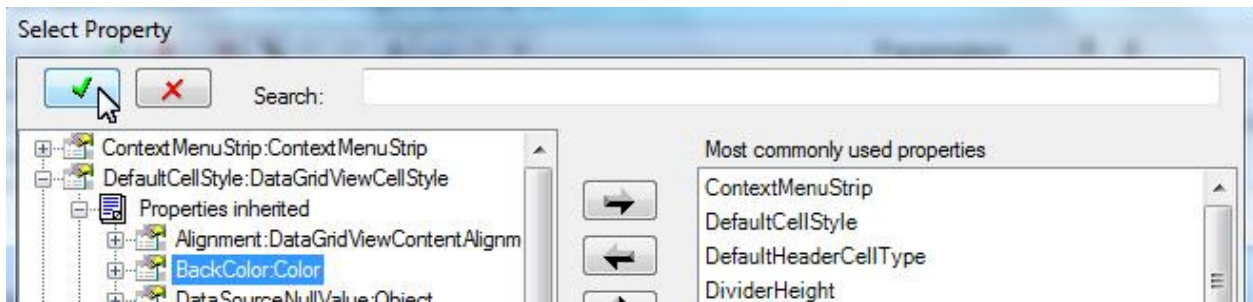
Another Method Editor appears for specifying actions for each row. Note that the icon “EasyGrid1.Rows.ExecuteForEachItem1.value” represents the row to act on:



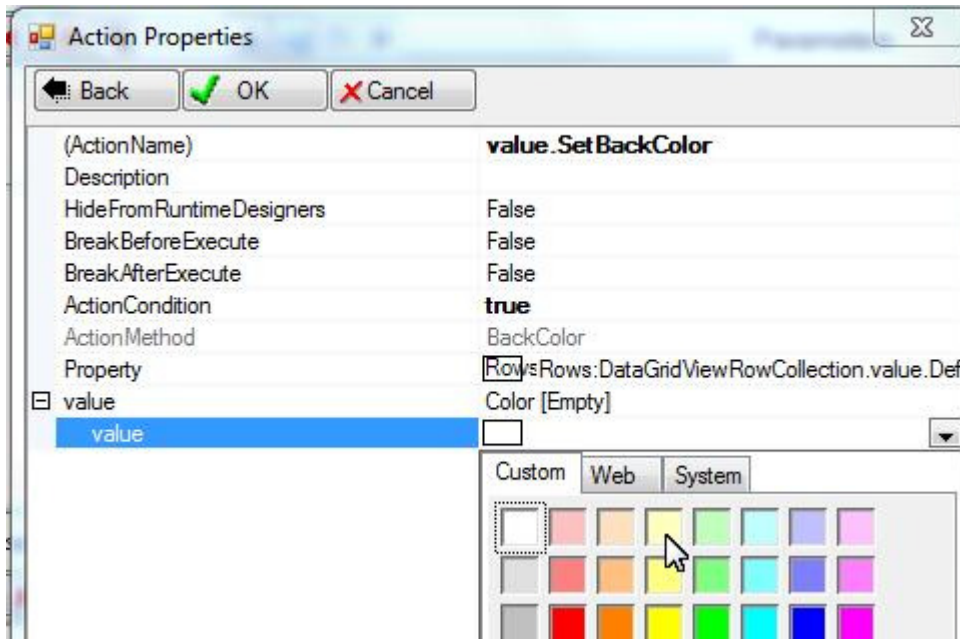
To set its background color, right-click “EasyGrid1.Rows.ExecuteForEachItem1.value”; choose “Create Set Property Action”; choose “\*All Properties\*=>”:



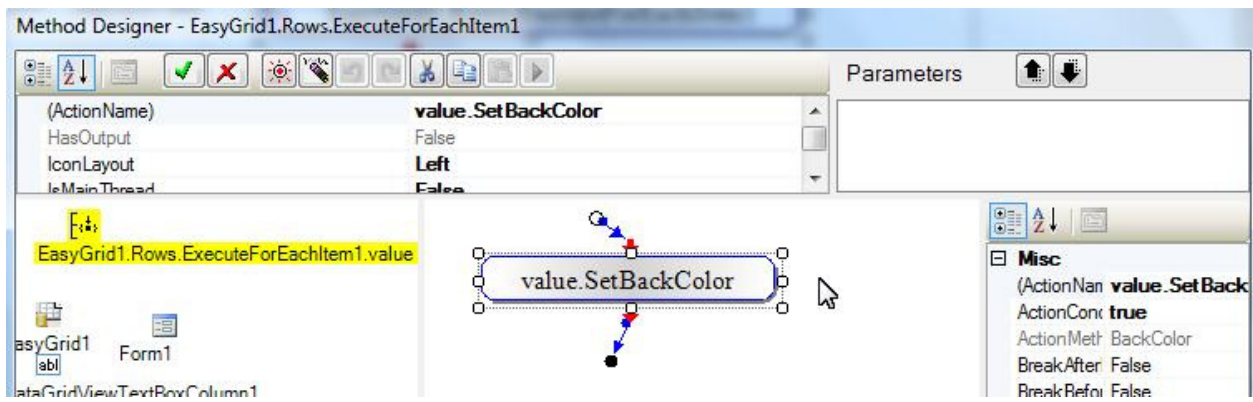
Select the BackColor property of theDefaultCellStyle:




Choose a color we want:

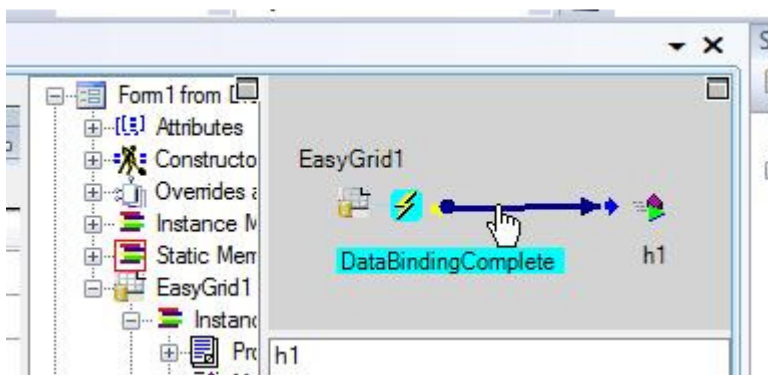


Click OK. The action appears in the Action Pane:

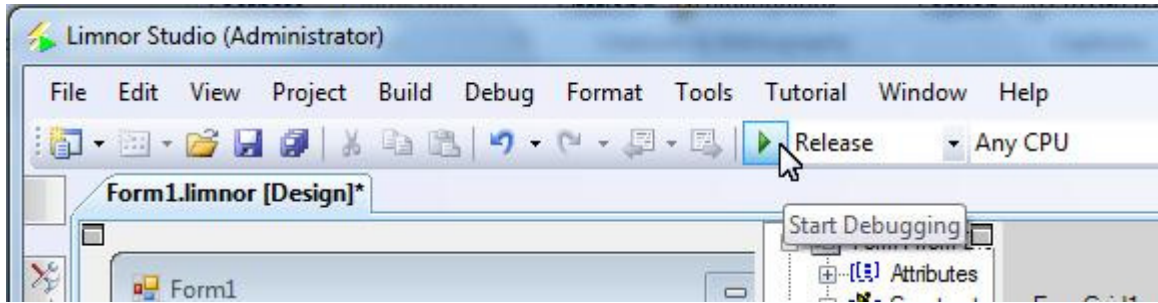


Click  to finish creating the methods.

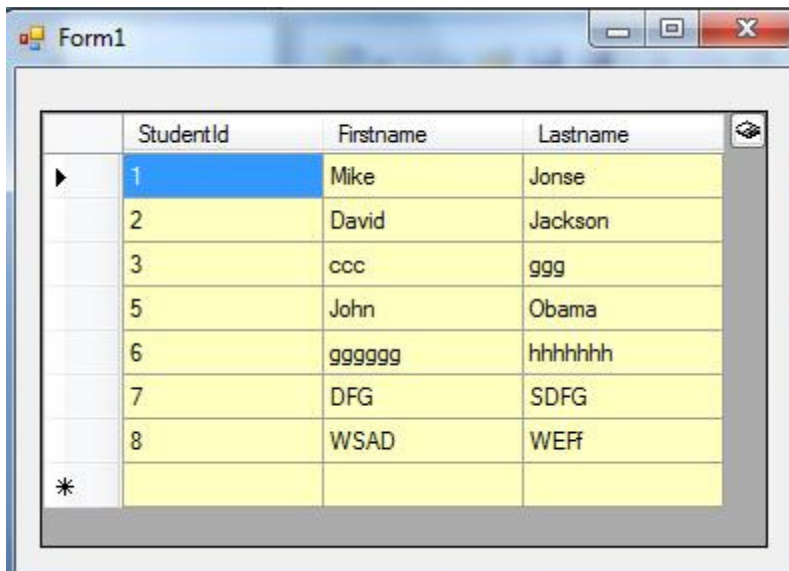
The event handler appears in the Event Path:



We may test the above programming now.



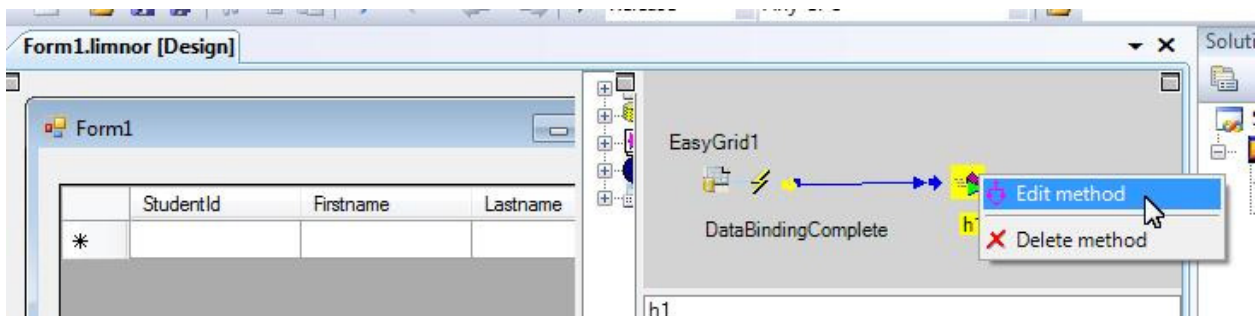
The Form appears. The background color is changed.

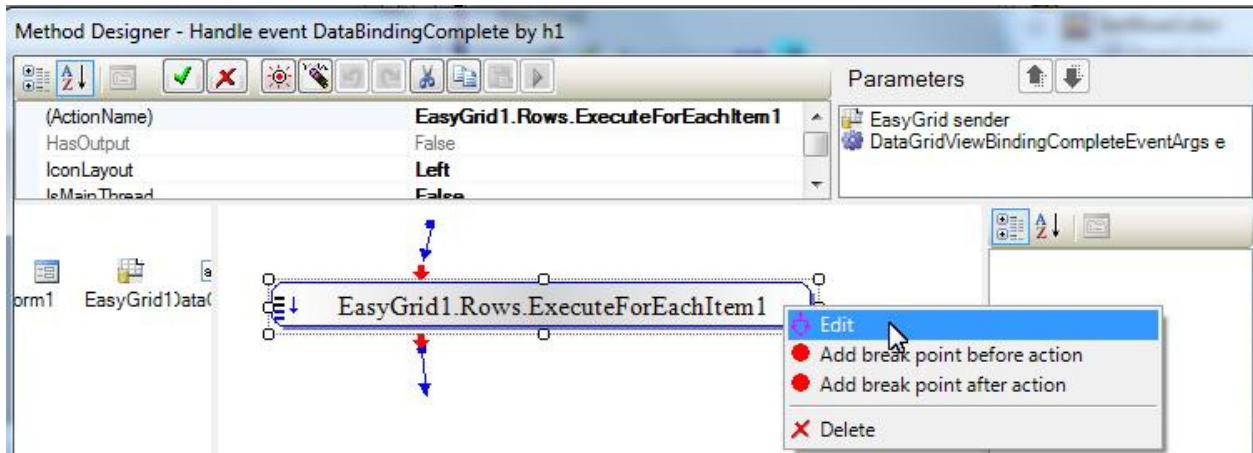


## Conditional Color Change

We may set the background color of each row based on the conditions. As an example, suppose we only want to apply the color for the row with "StudentId" field being 6.

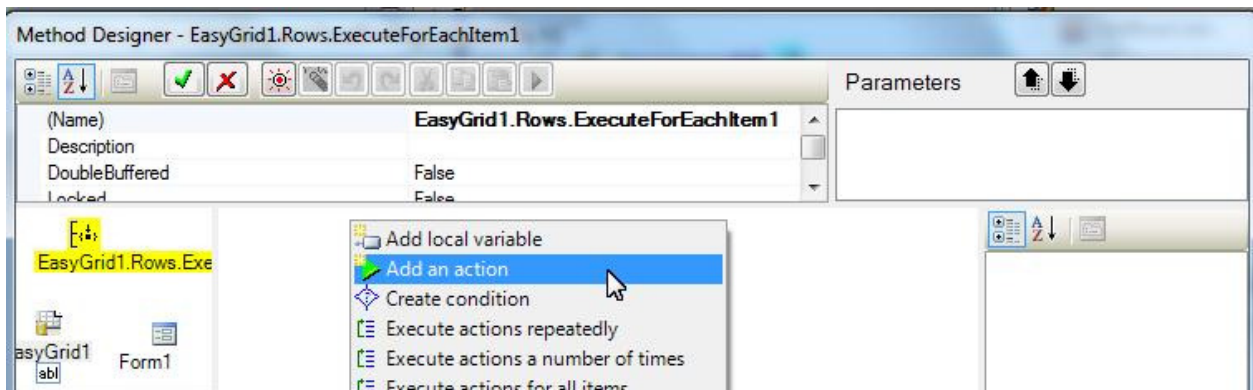
Open the event handler method to modify it:



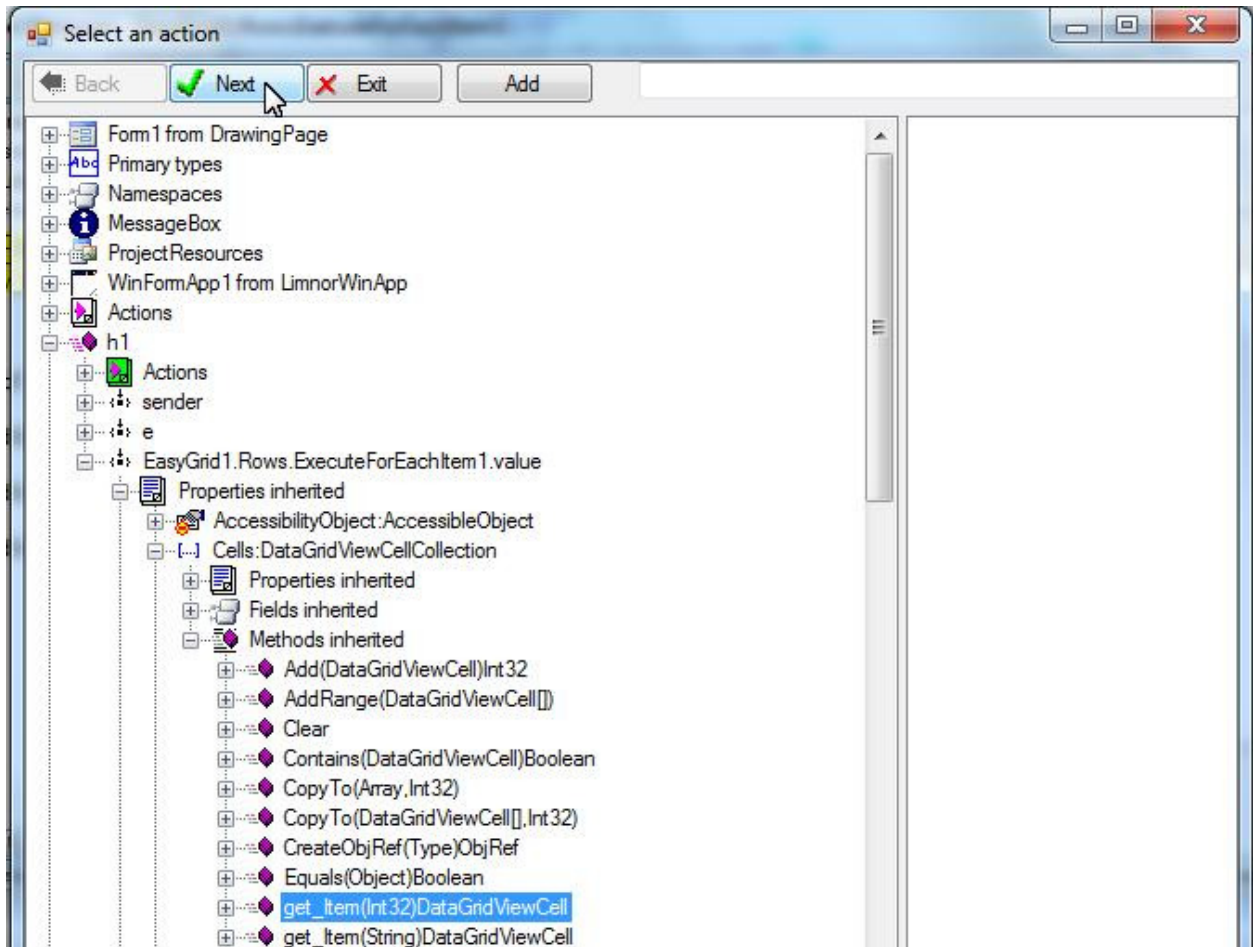


We want to access the first cell of the row. Each row has a Cells property. We may create an action to get the first cell, which is for the "StudentId" field.

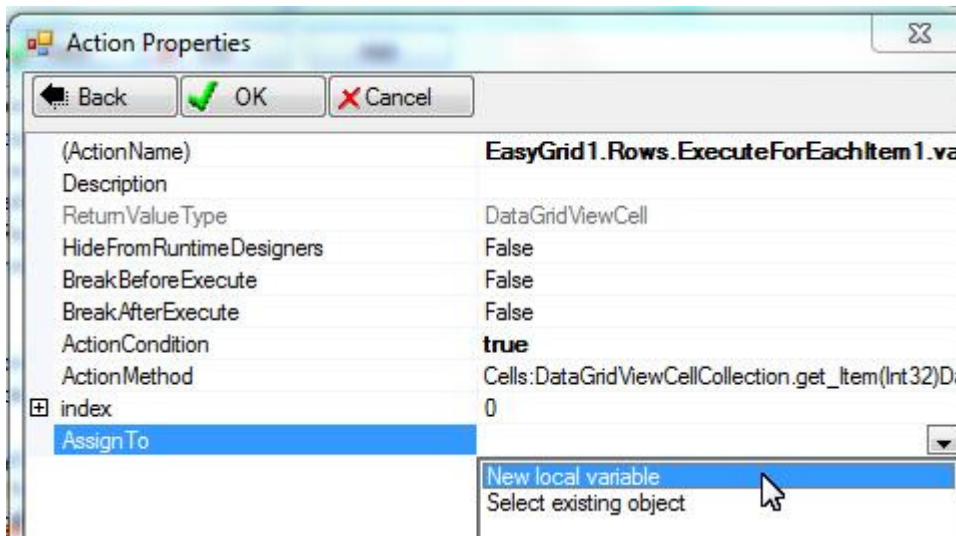
Right-click the Action Pane; choose "Add an action":



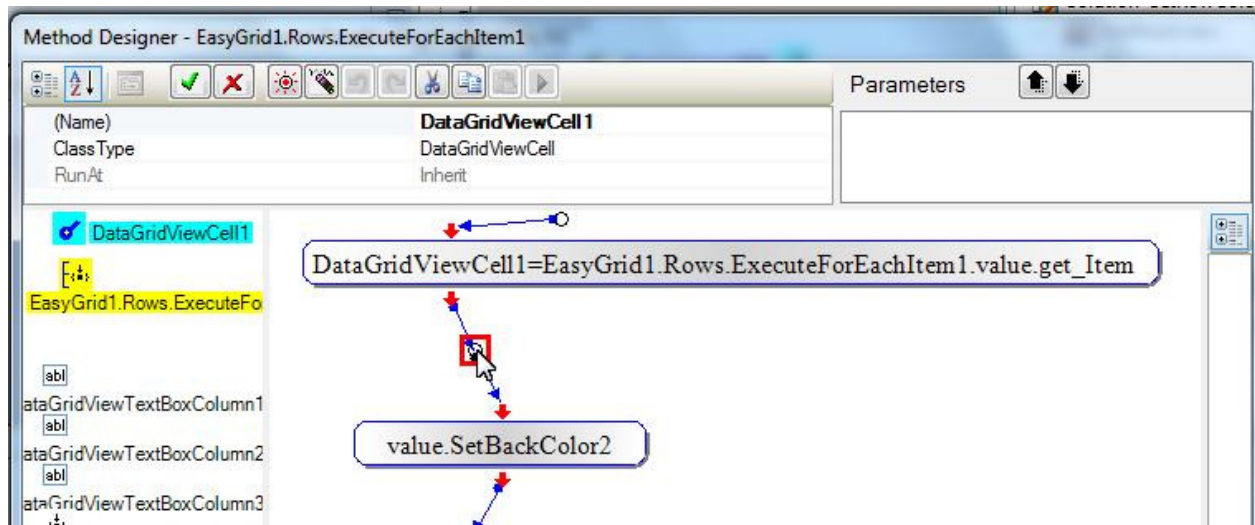
Select "get\_Item" method of the Cells:



0 for "index" indicates the first cell. Choose "New local variable" for "AssignTo":

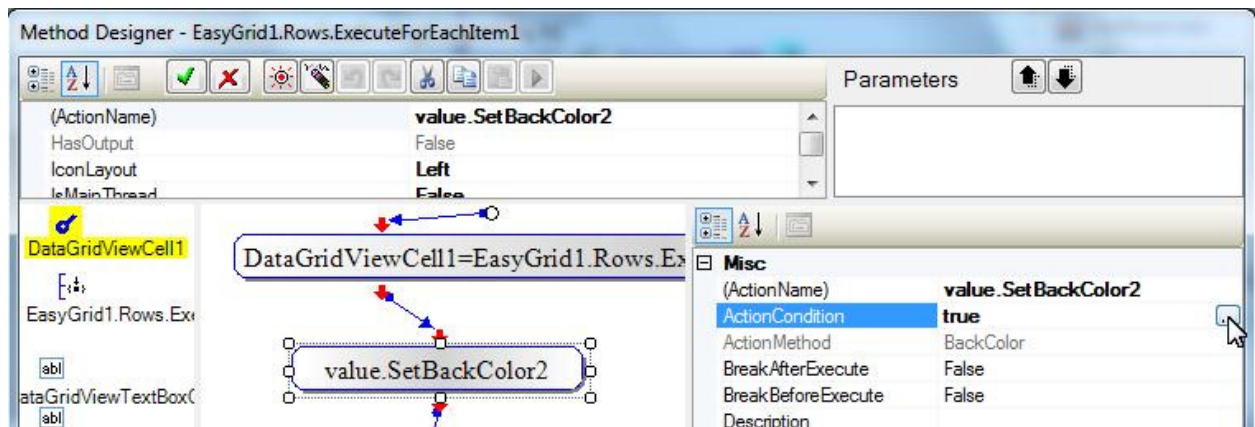



Click OK. The action appears in the Action Pane. Link the new action with the existing action:

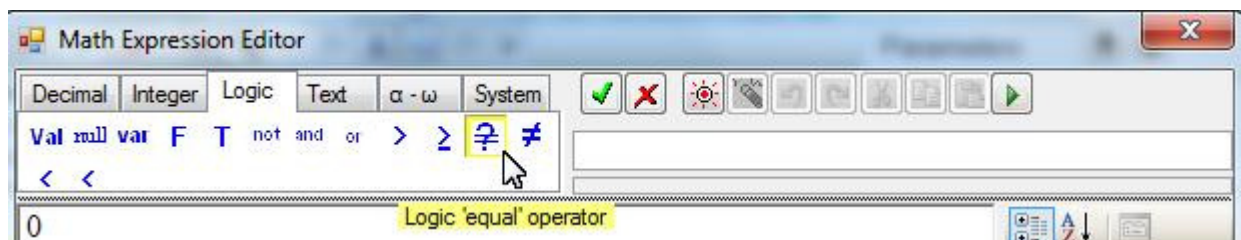


Note that a new variable “DataGridViewCell1” appears in the Variable Pane. The Value property of the variable represents the field value of the cell. We may use it to form conditions.

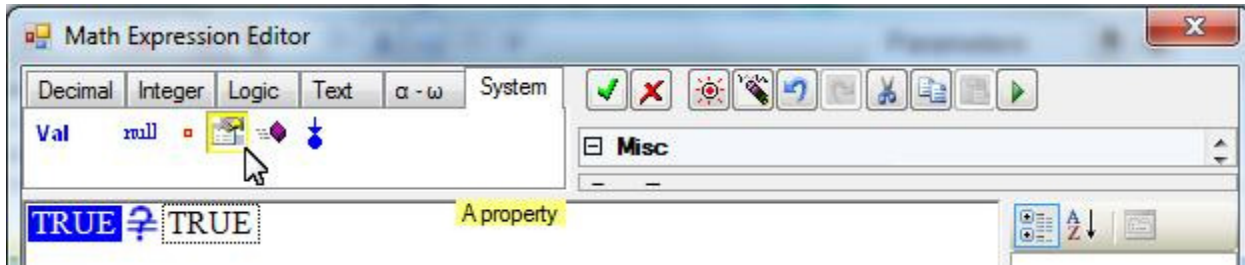
Set the ActionCondition property of the color-setting action:



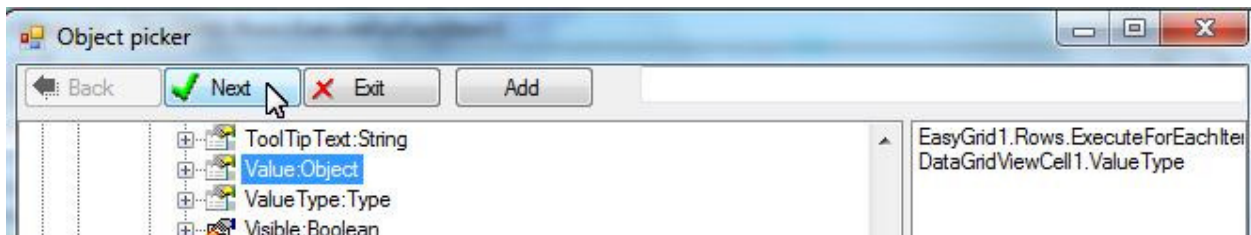
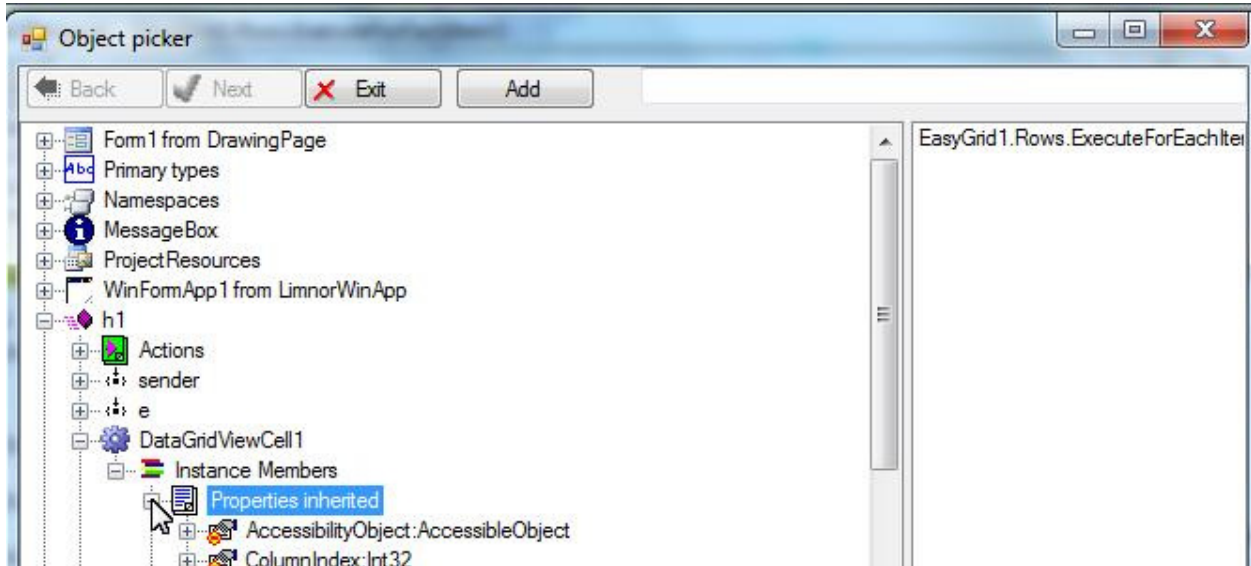
The Expression Editor appears. Suppose we want the condition to be that the cell value is 6. Click icon 



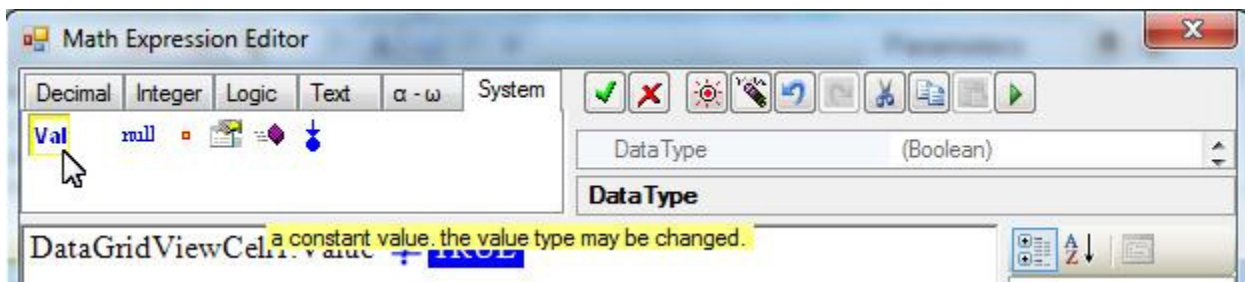
Select property icon for the first element:



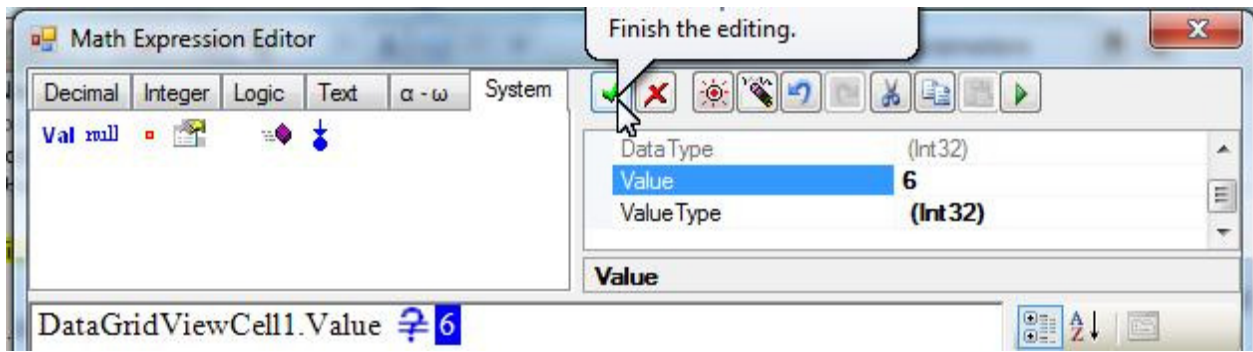
Select the Value property of the cell:




Select constant value icon for the second element:



Set the constant value to be 6. This is our sample condition:



Click  to finish modifying the methods. Test the application. This time only one row changes color.

