

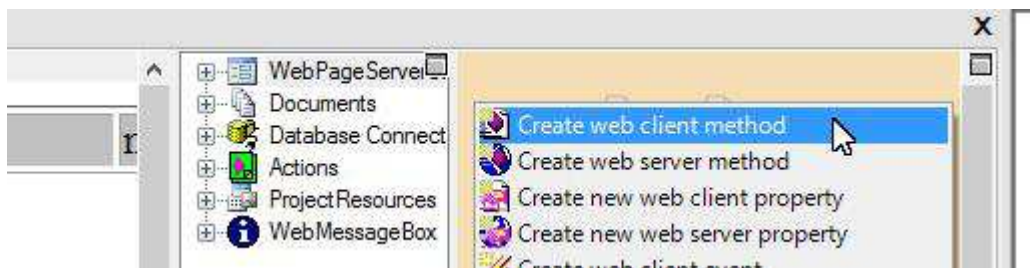
Execute Server Actions from Client Methods

Contents

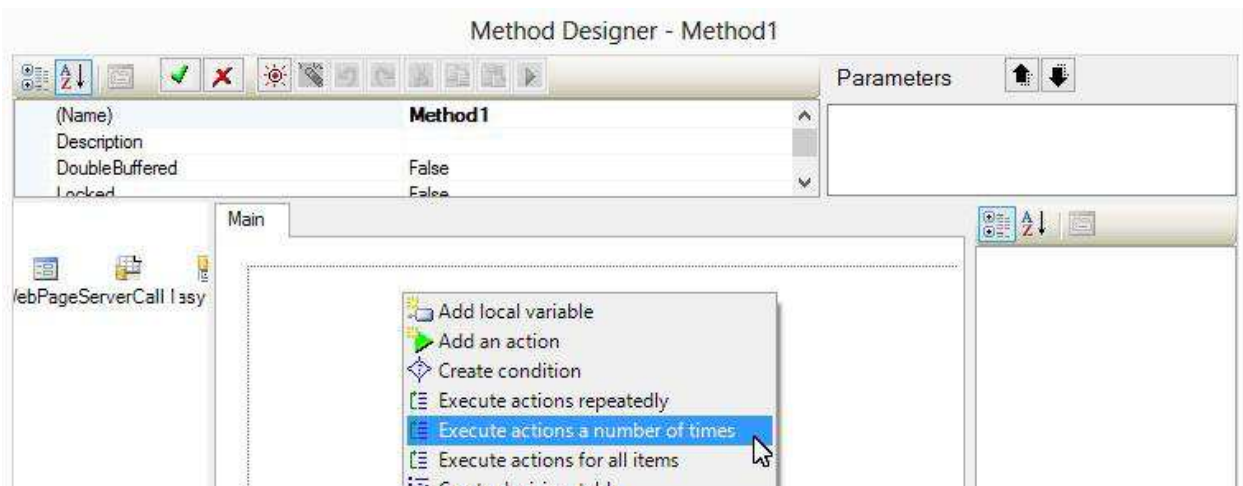
Scenario.....	1
One Problem	6
A Solution.....	8
Create a client event	8
Assign actions to new event	8
Fire event	11
Test.....	17
Conclusion.....	19

Scenario

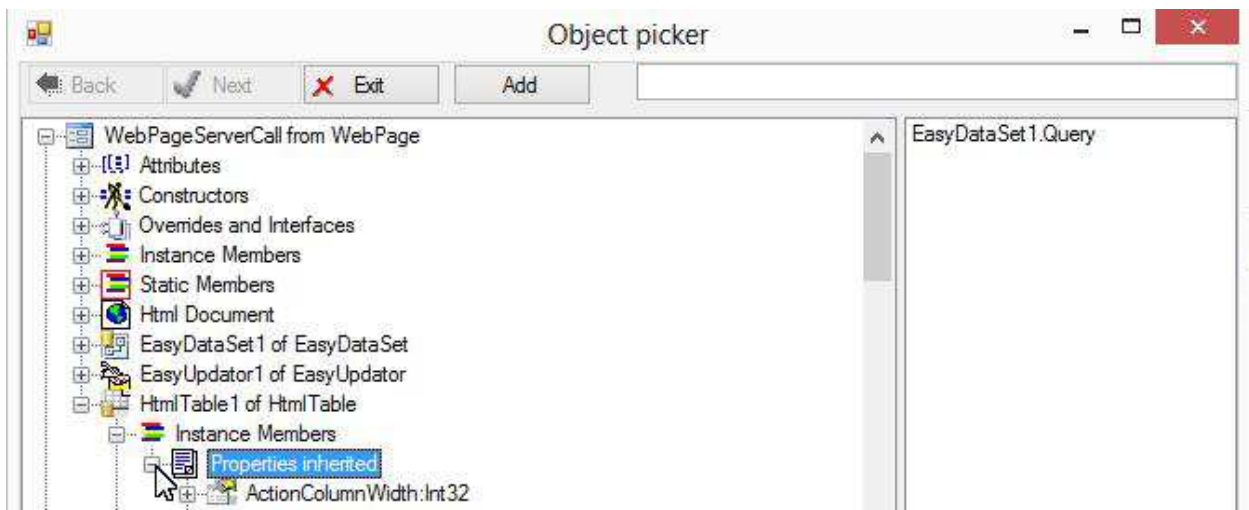
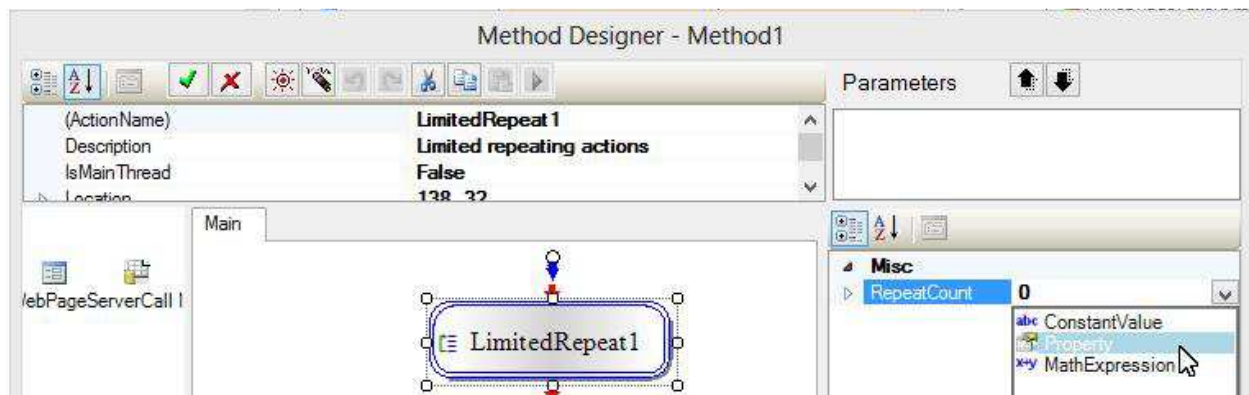
Consider such a scenario: a client method is created to do some processing:



Suppose we want to process all rows in a table. We may create a repeat action:

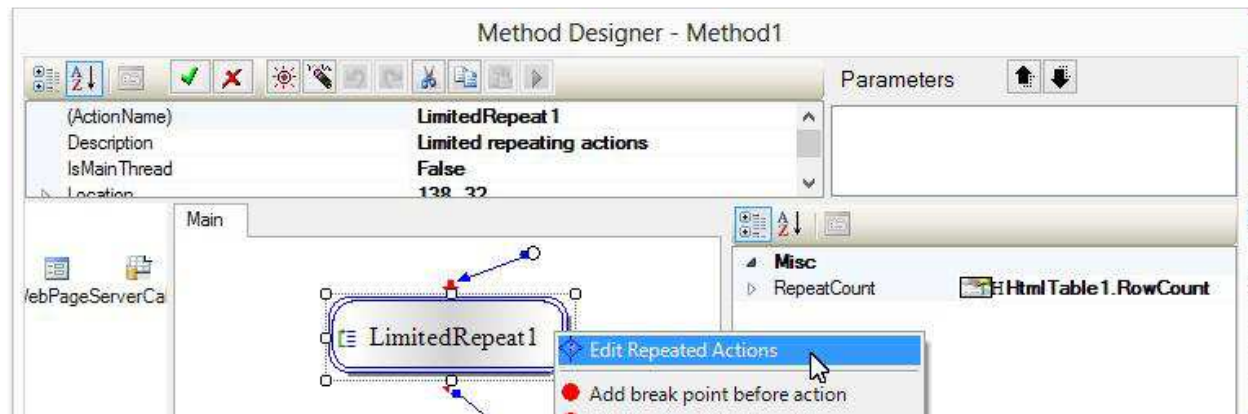


The repeat count can be the RowCount property of the table:

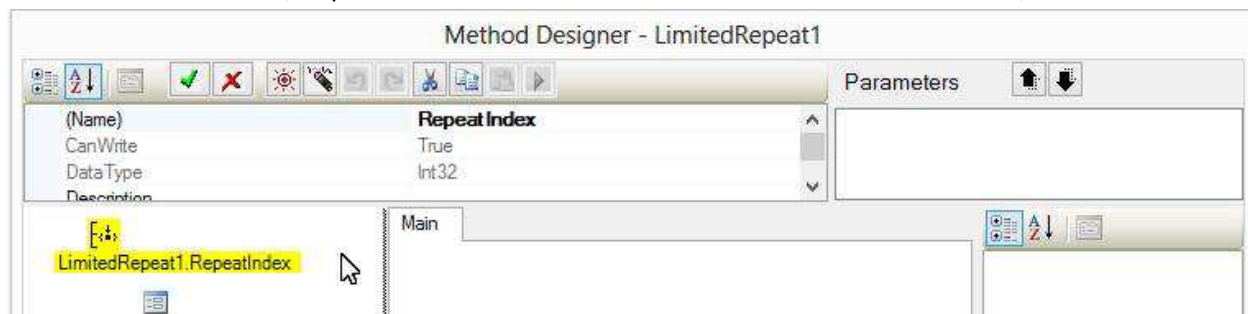




To specify actions to be repeatedly executed, right-click the repeat action, choose “Edit Repeated Actions”:

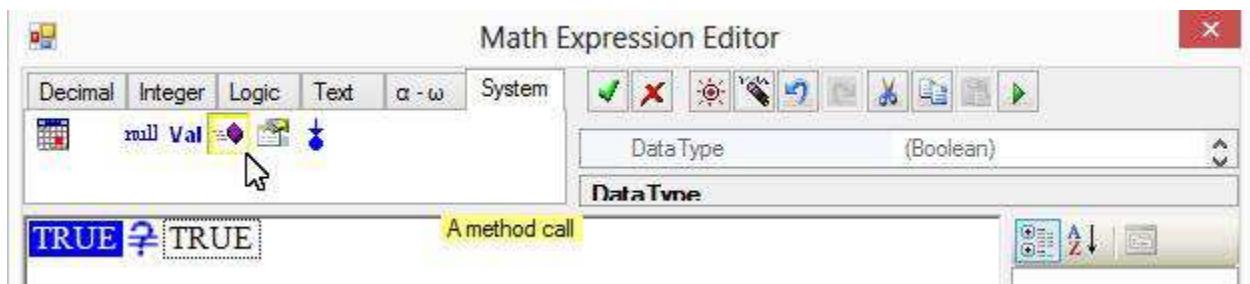
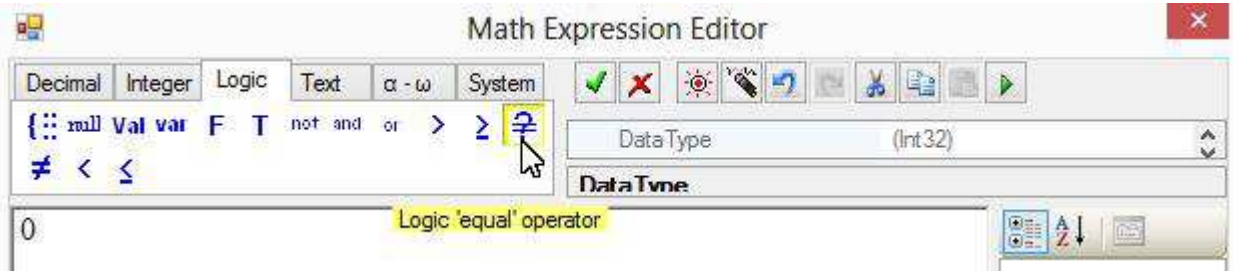
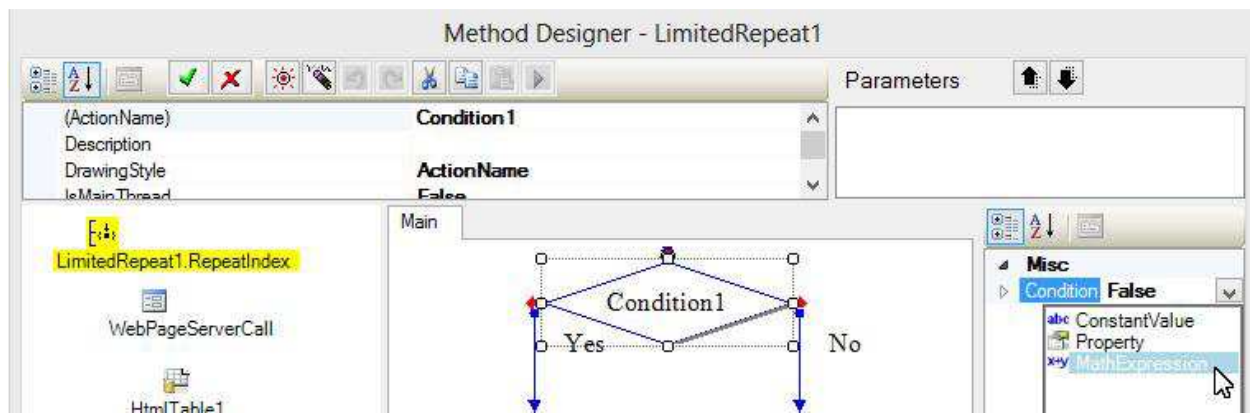
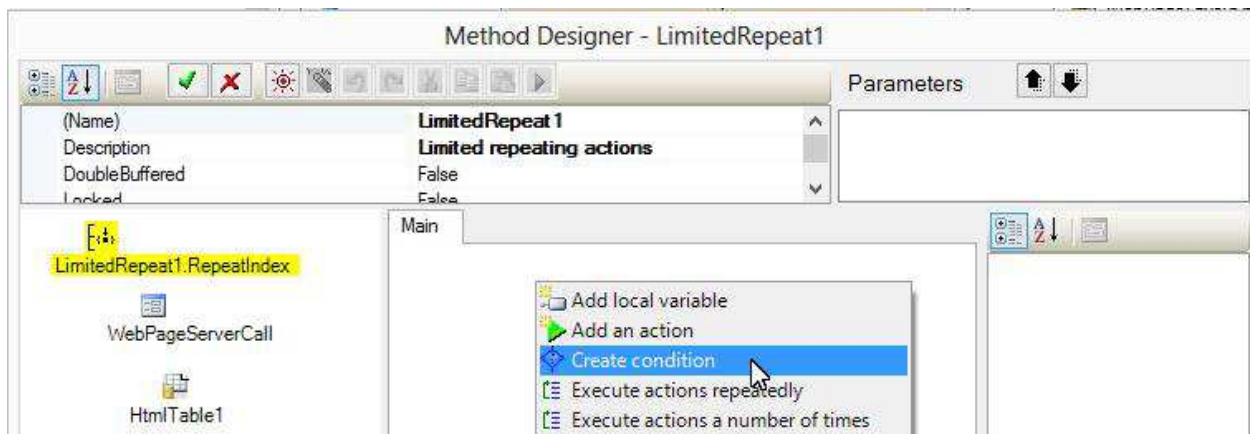


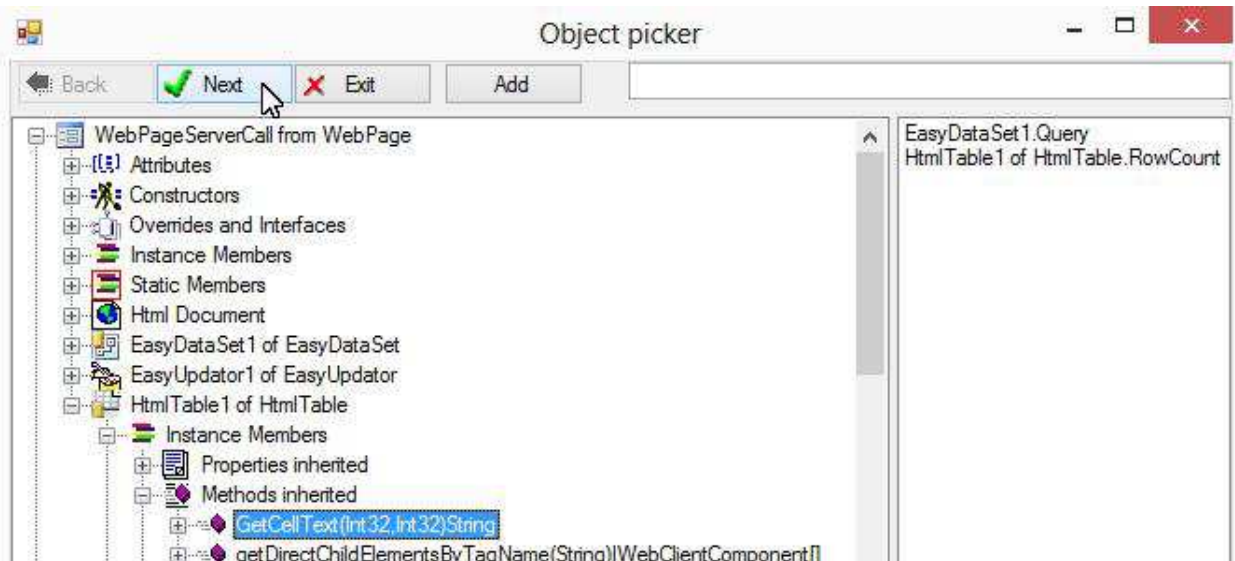
A new Method Editor appears for specifying actions to be repeatedly executed. Note a variable RepeatIndex. It indicates the repeat times. When the repeat action is executed, it will execute RepeatCount times, each time with a different value of RepeatIndex. RepeatIndex=0 indicates that it is the first time execution; RepeatIndex=1 indicates that it is the second time execution; and so on.



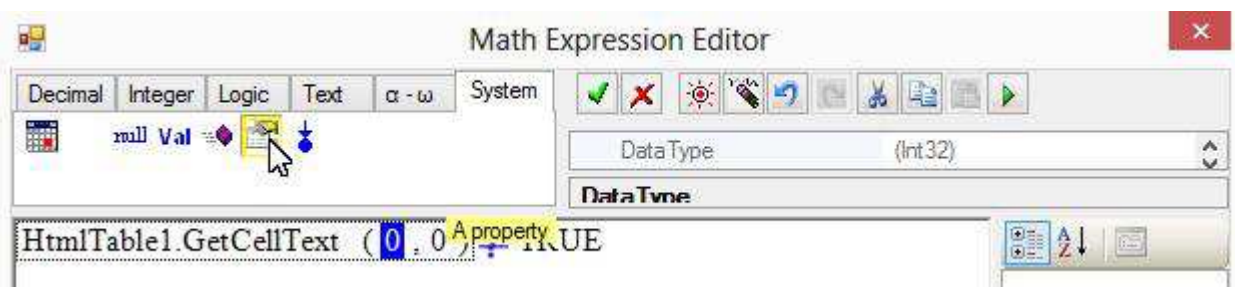
In our sample here, we may use RepeatIndex to indicate the row number of the table. RepeatIndex=0 indicates that we want to process the first row; RepeatIndex=1 indicates that we want to process the second row; and so on.

In our sample table, there is a field named “Active”. Suppose we only want to process the rows with field “Active” equals 1. Add a Condition action to check “Active” field:

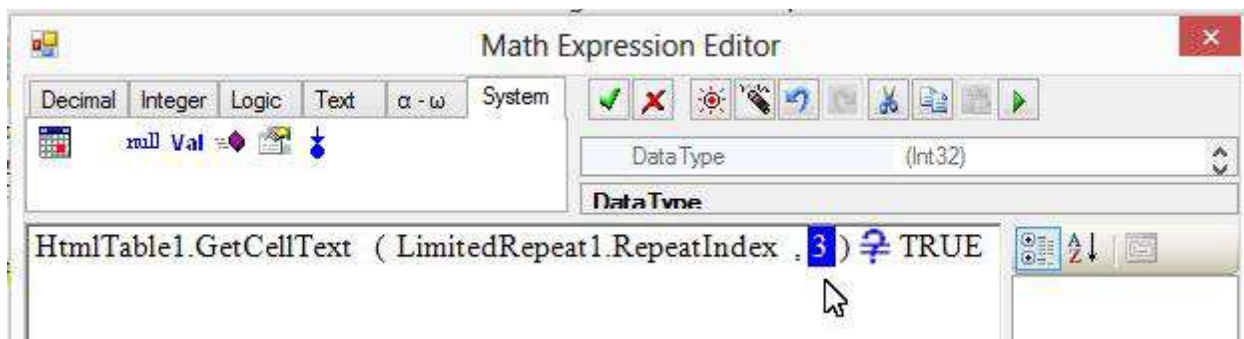




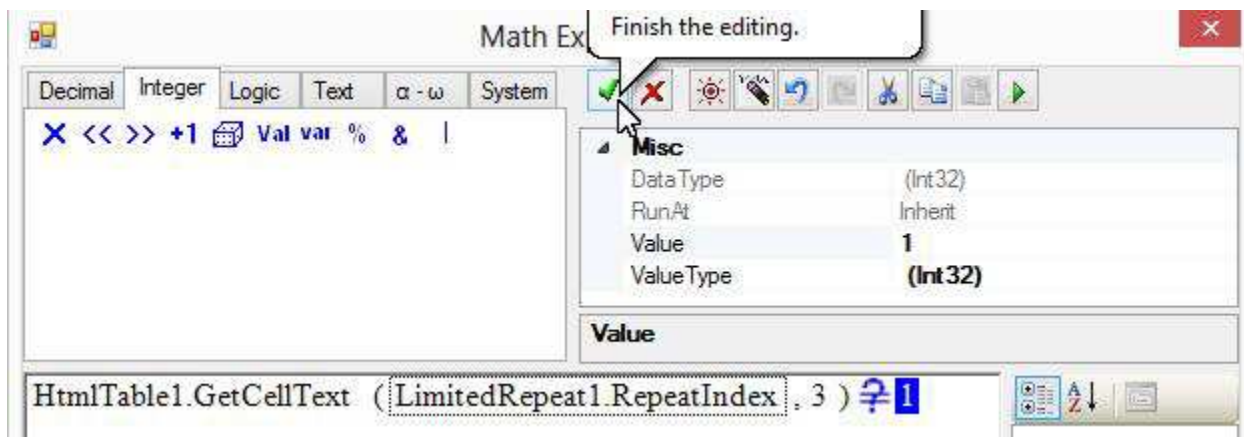
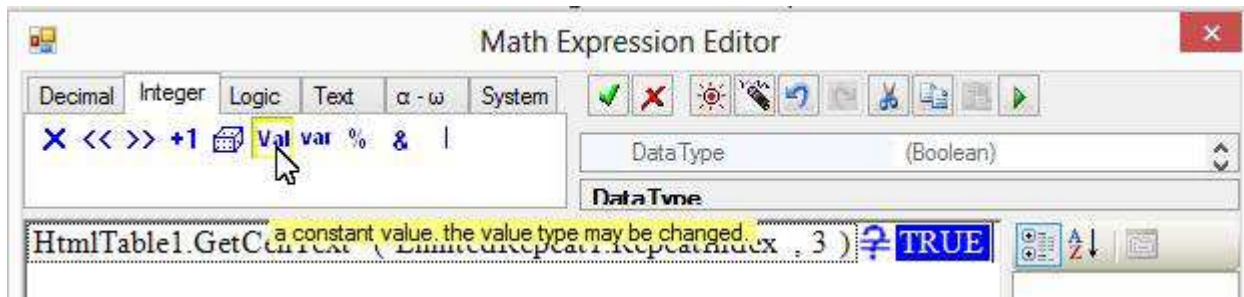
GetCellText requires row number and cell number. For the row number, use variable RepeatIndex:



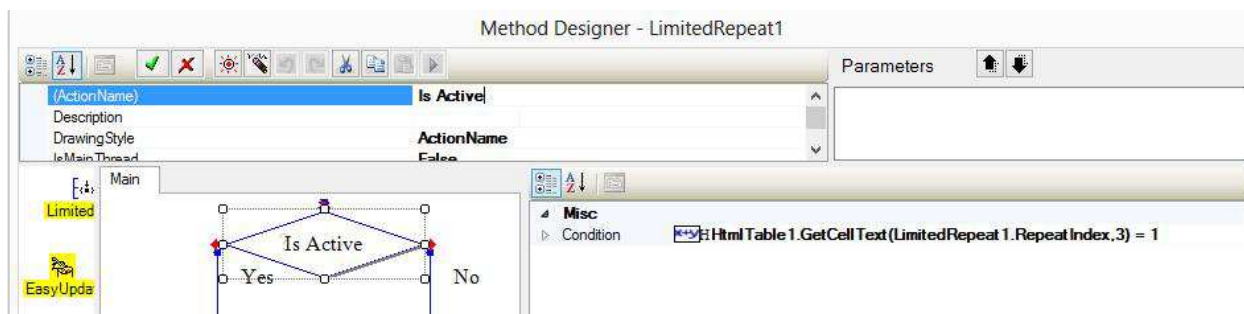
For the cell number, use 3 because in our sample, "Active" field is the third column:



Enter an integer 1 for the equation:



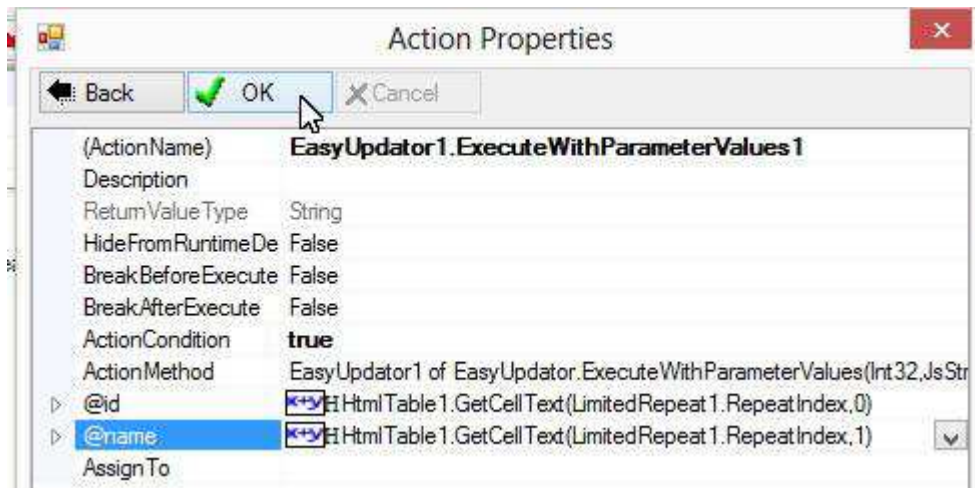
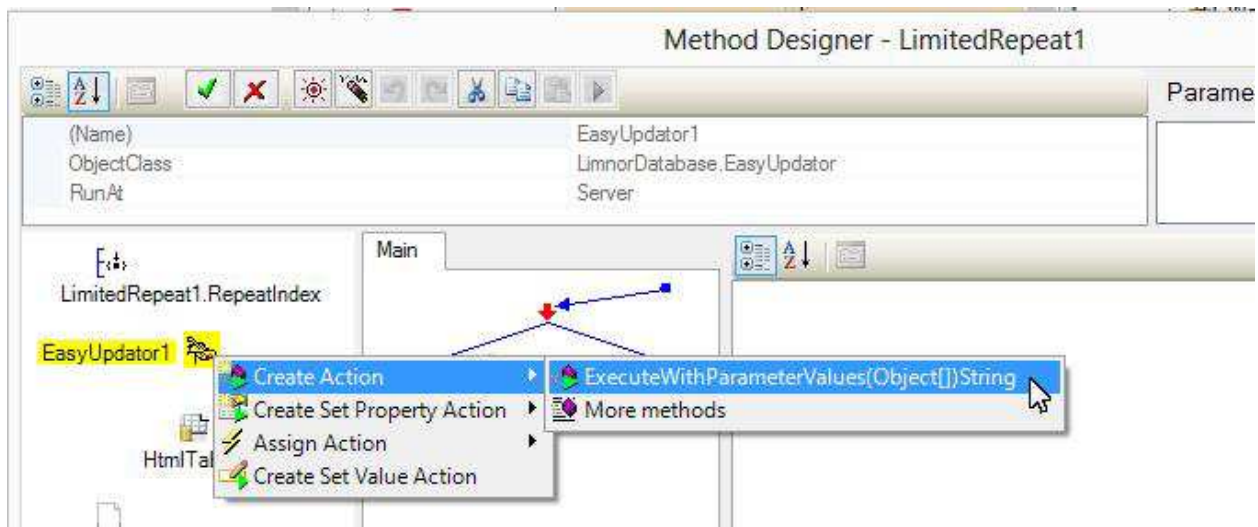
Rename the Condition action:



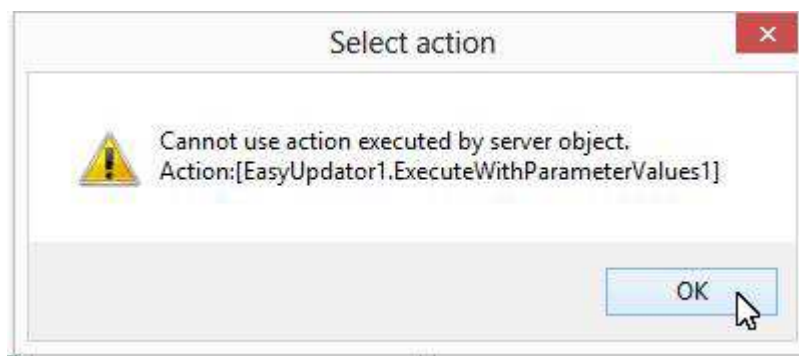
Now we are ready to add actions to be executed.

One Problem

Suppose one action we want to execute is to use values at the row to create a new record in a database. We know that the row number is RepeatIndex. We may use GetCellText of the table to get value from the row by specifying column number, as we did to get value for field "Active". We may use an EasyUpdator component to insert a new record and passing parameters to it via GetCellText. Let's do it and see what problem we will get.



It says we cannot use this action in this context:

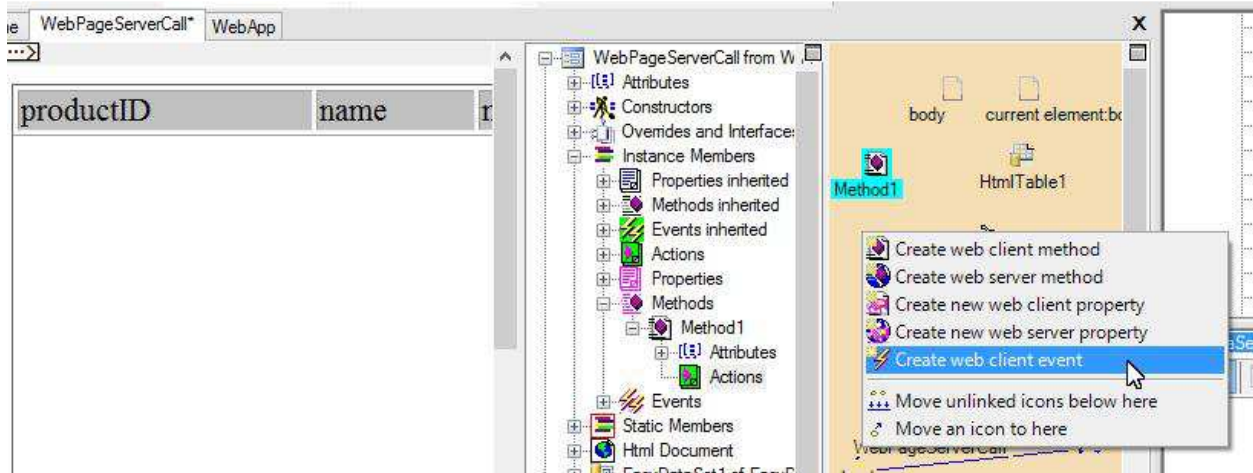


The above message is telling us that inside a client method, we may only execute client actions.

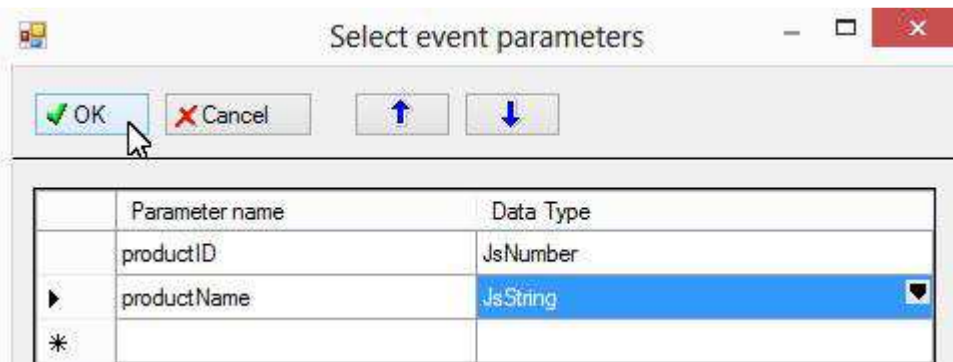
A Solution

Is there a workaround? Yes. The basic idea is that a client event may invoke any actions including server actions. Firing a client event is a client action. So, in a client method, we may execute an action to fire a client event. We may assign server actions to be executed to the client event.

Create a client event



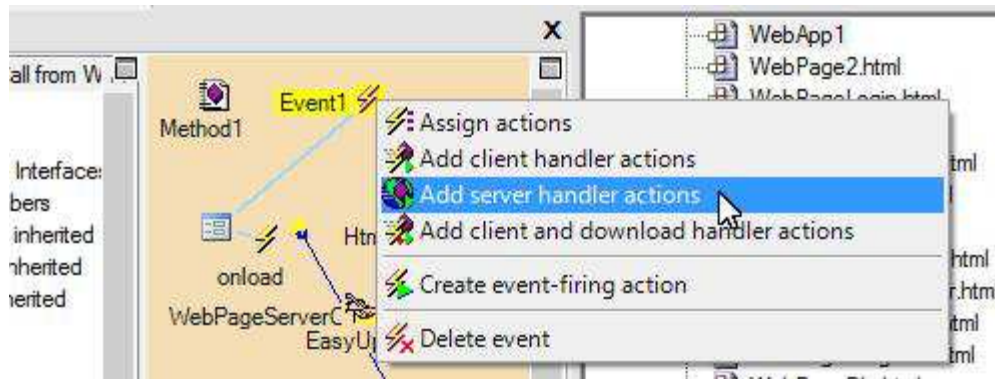
In our sample, we need to pass values necessary for creating a new record in this event. For our sample, we need an integer value and a string value for creating the new record. The values will be from the row being processed. So, we add two parameters for this new event:



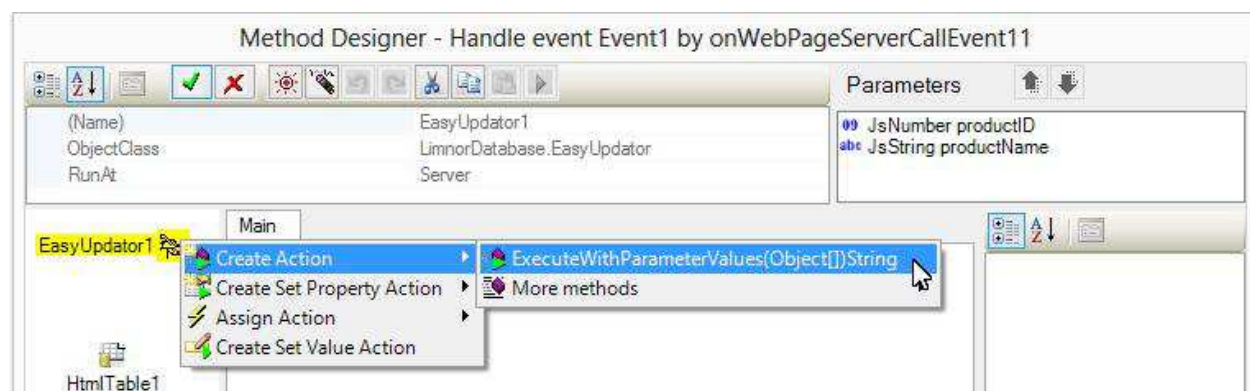
A new event, Event1, is created. We may rename it to make it more meaningful. For this sample, we leave it as Event1.

Assign actions to new event

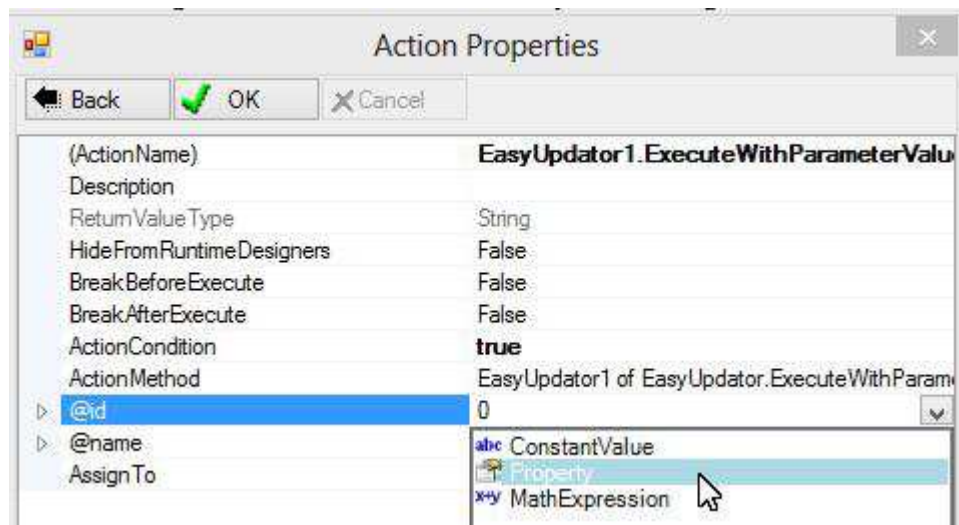
Add an event handler to Event1 so that its parameters can be used:

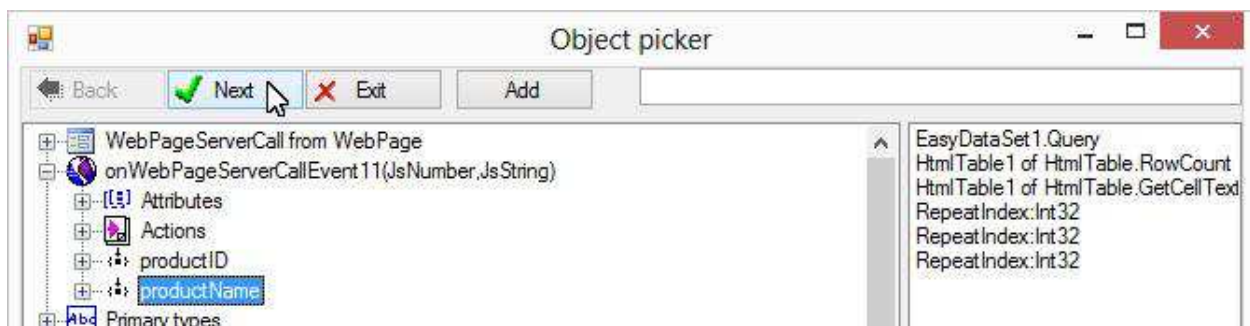
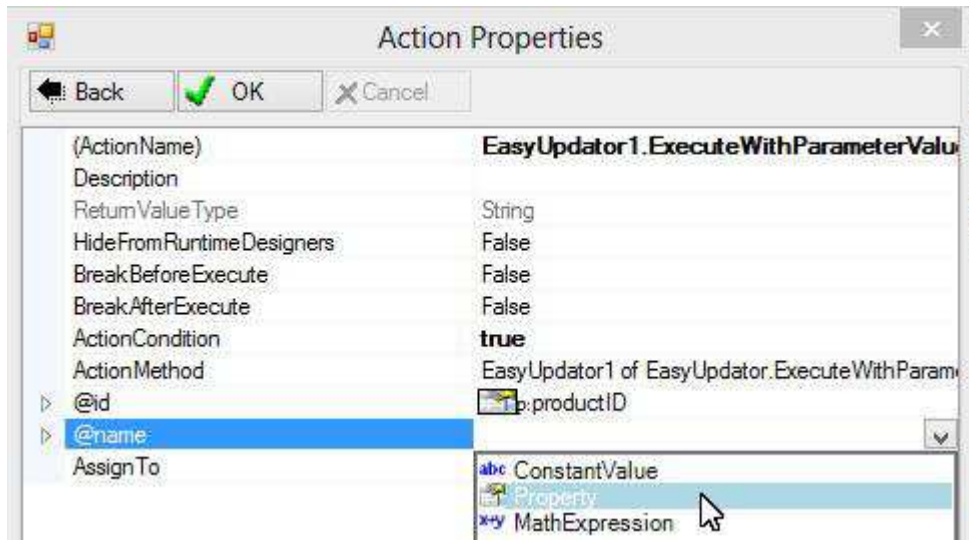
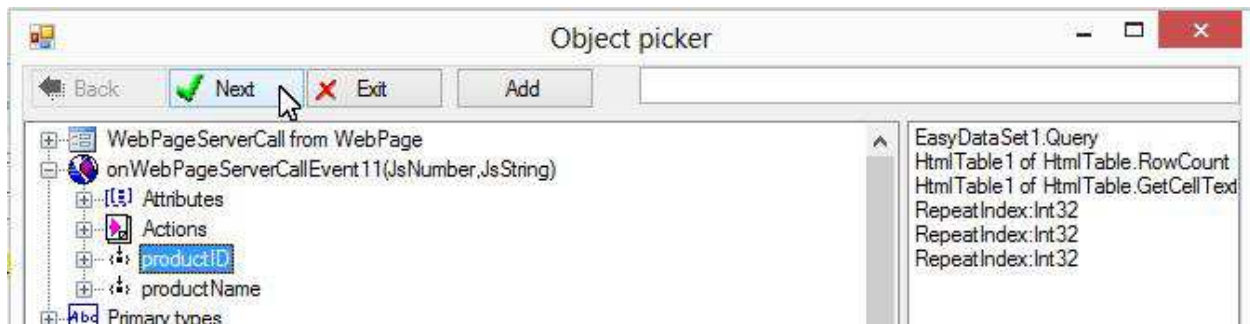


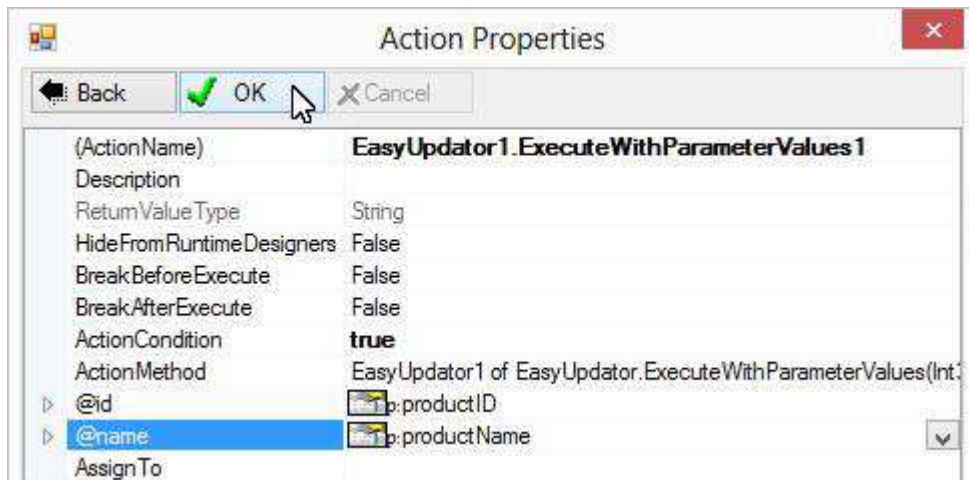
Add an action to create a new database record:



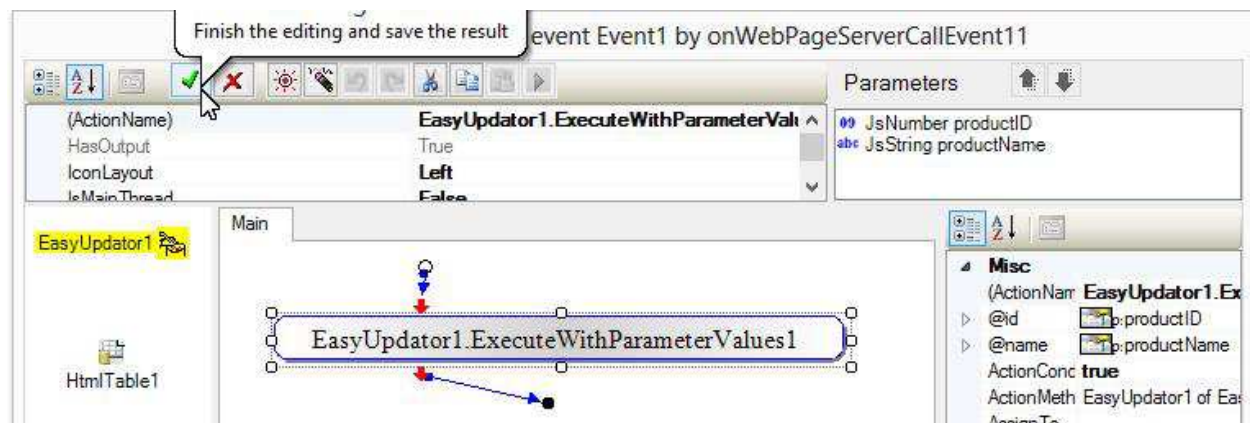
Pass event parameters to the action:







For this sample, we only use this one action for this event:

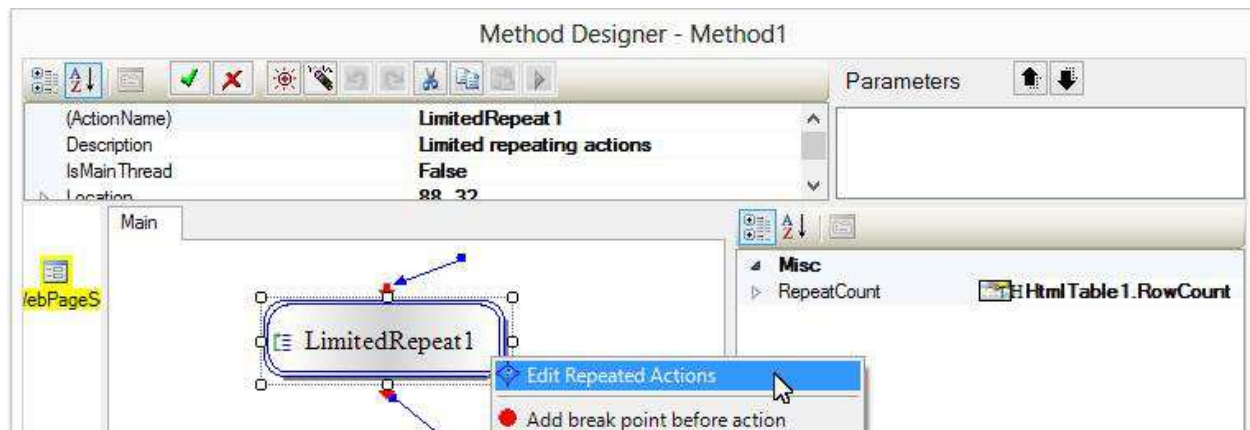
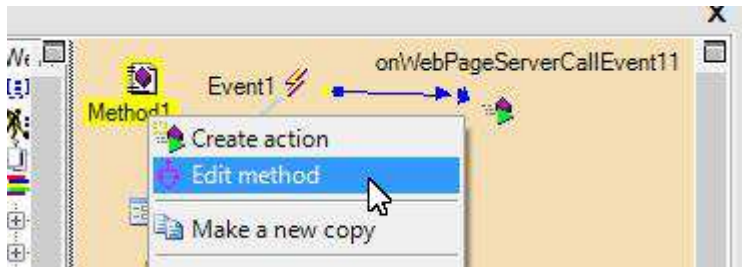


An event handler is assigned to Event1:

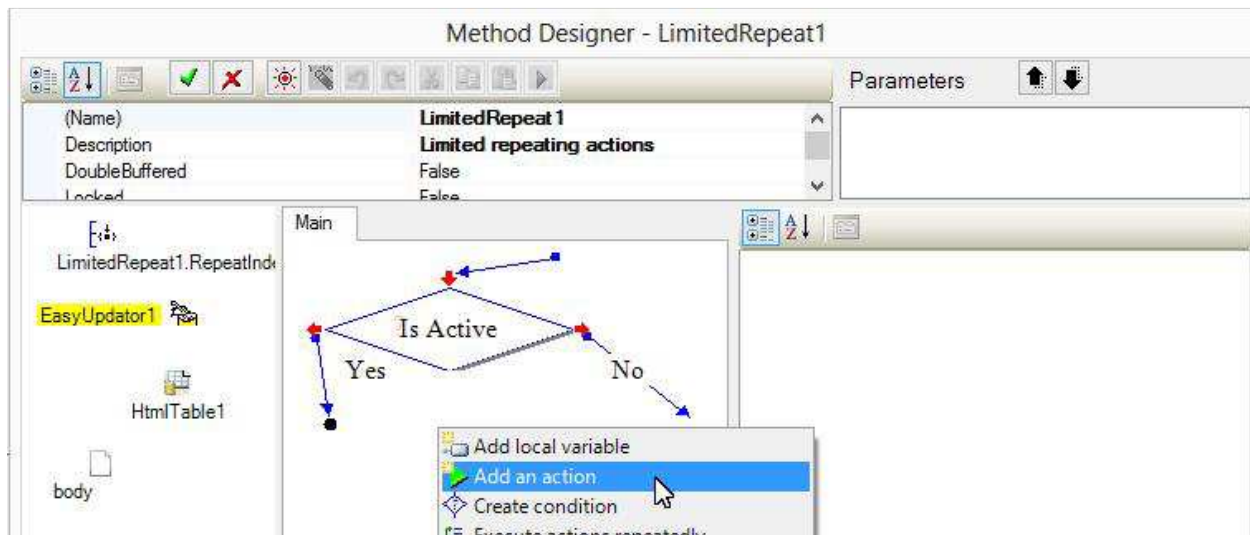


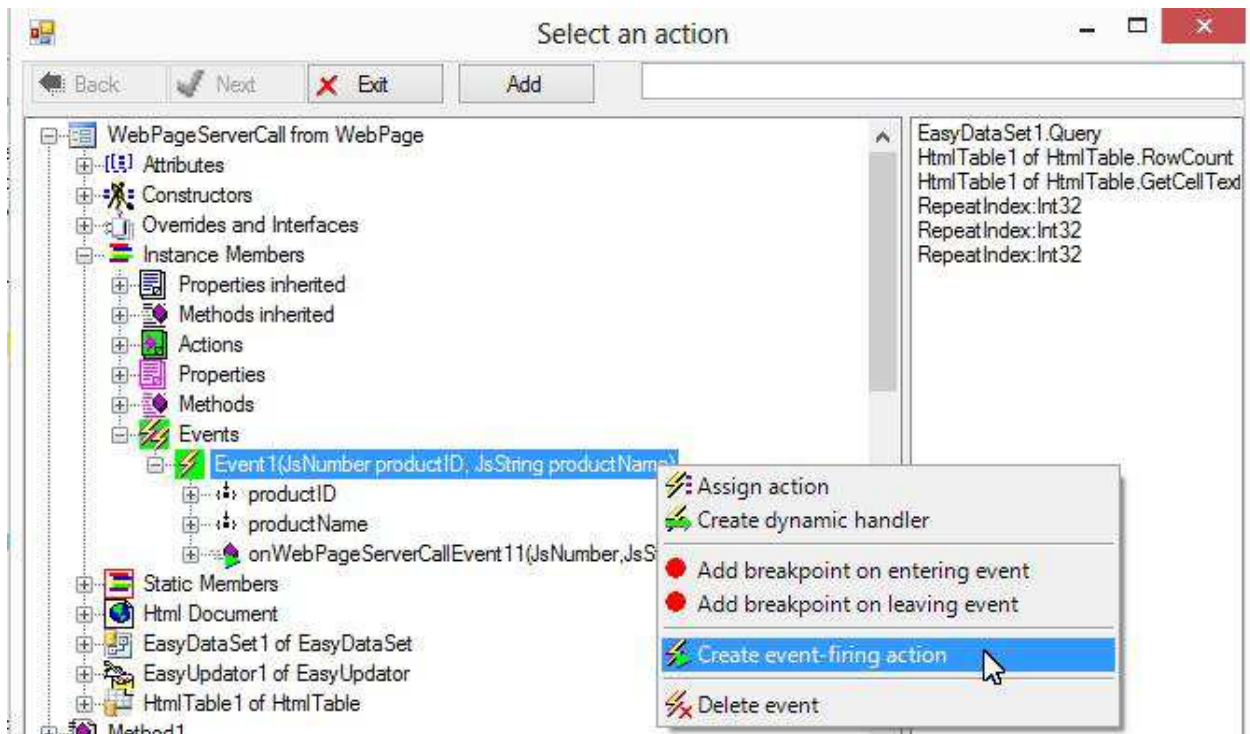
Fire event

In our original client method, instead of executing an action to create a database record, we execute an action to fire event Event1:

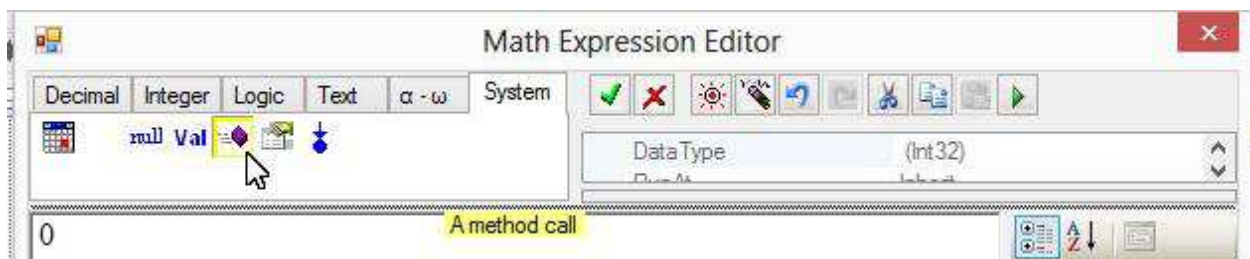
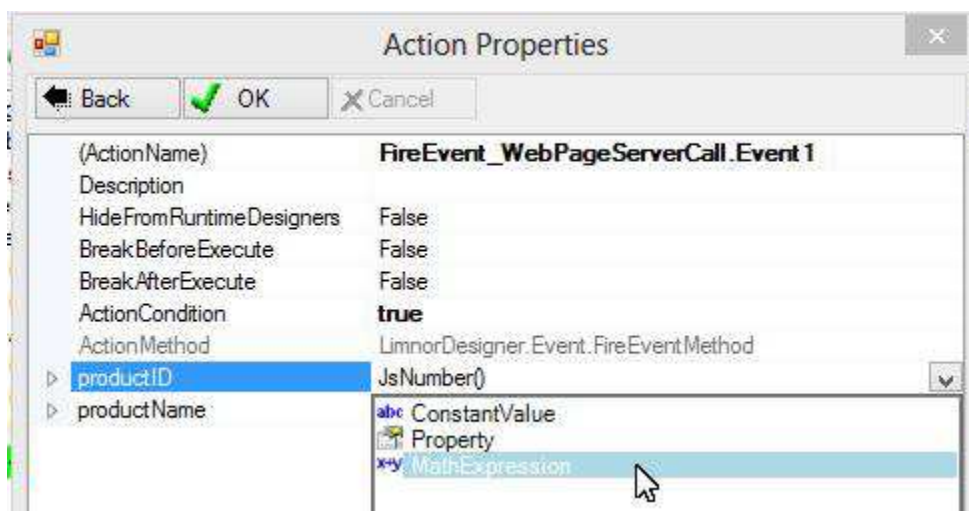


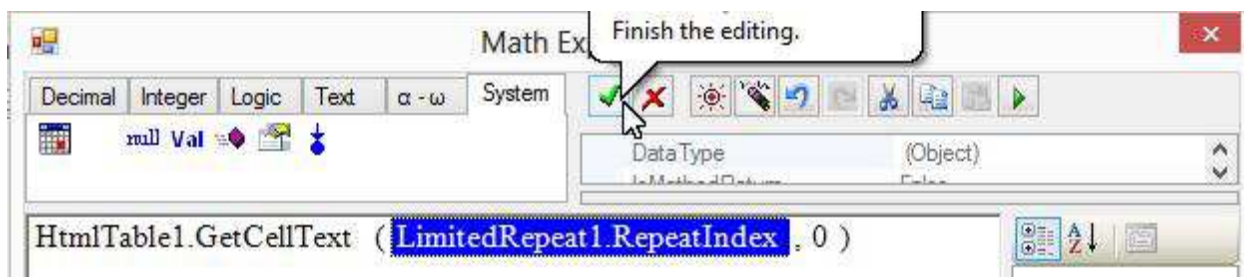
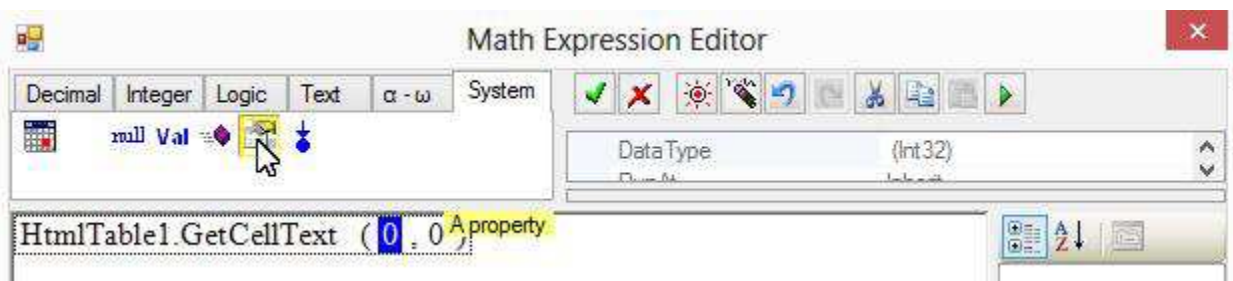
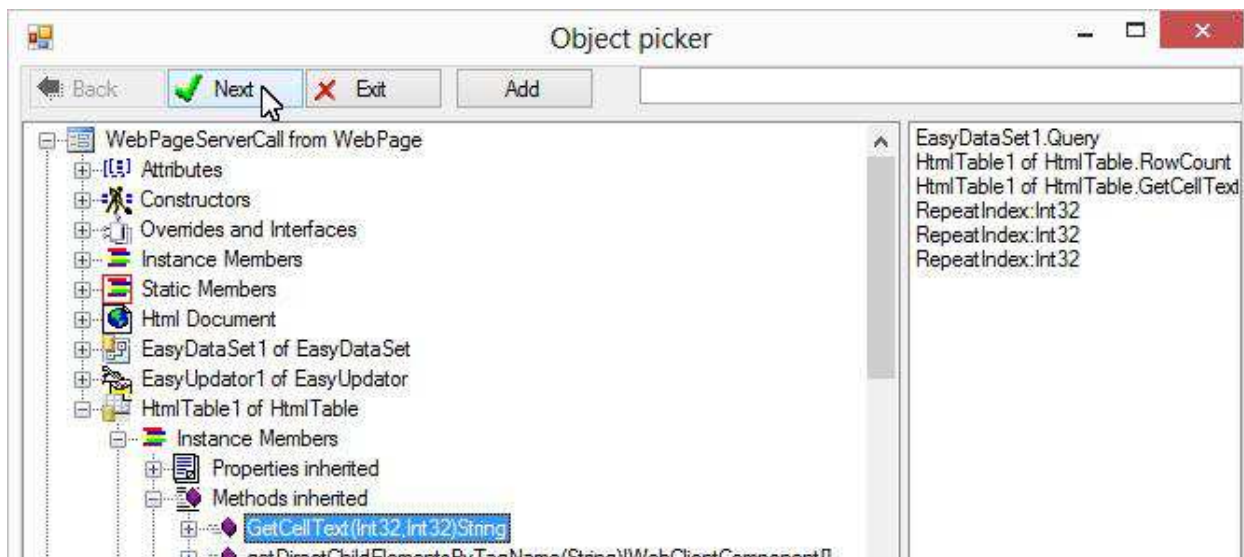
Add an action to fire Event1:

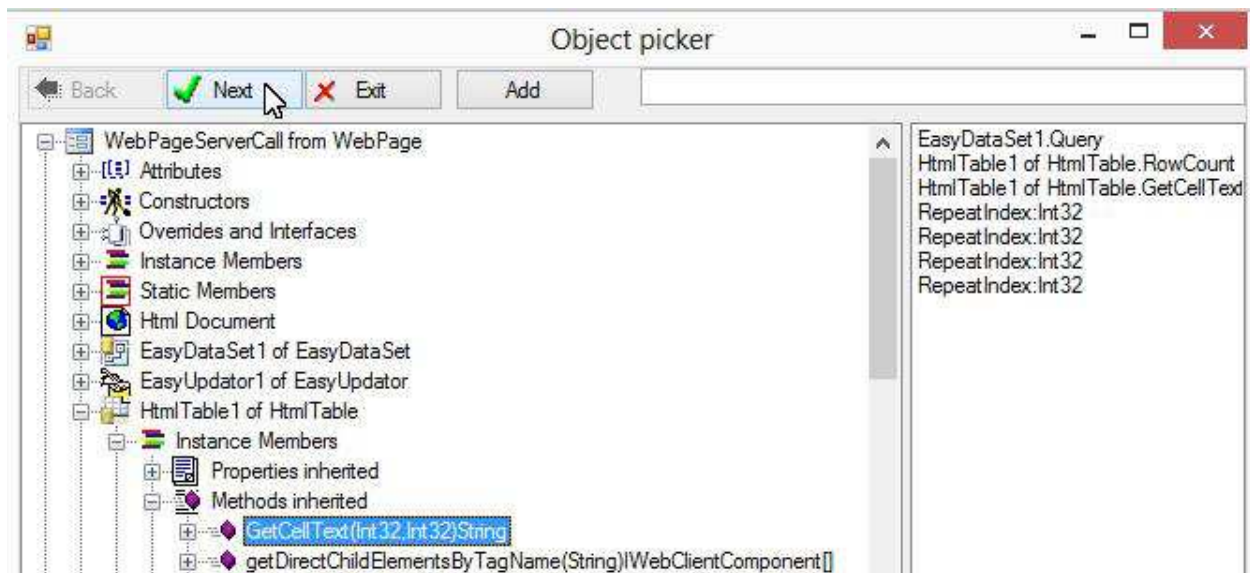
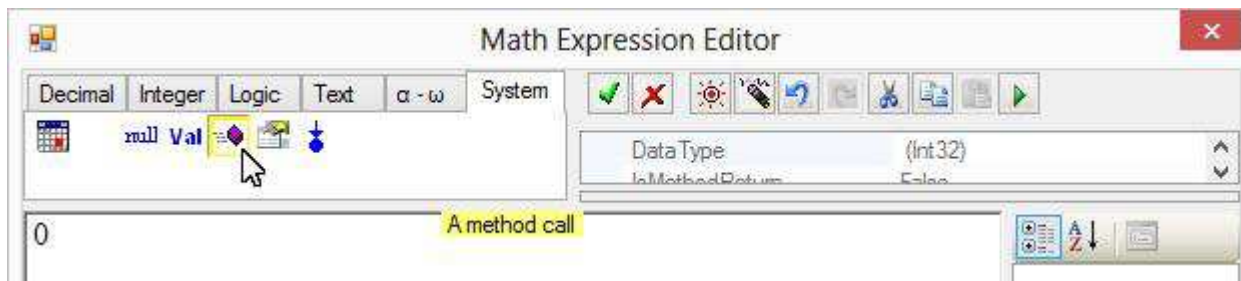
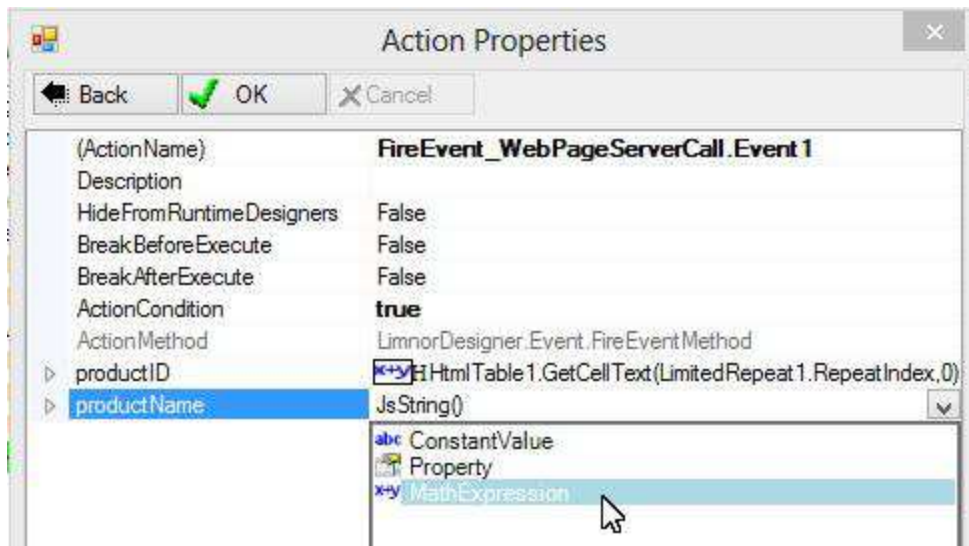


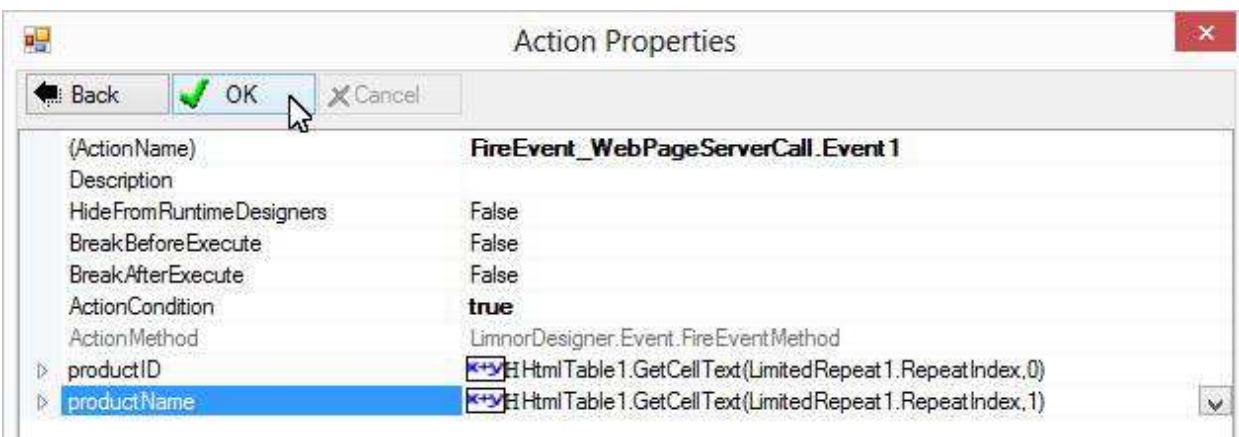
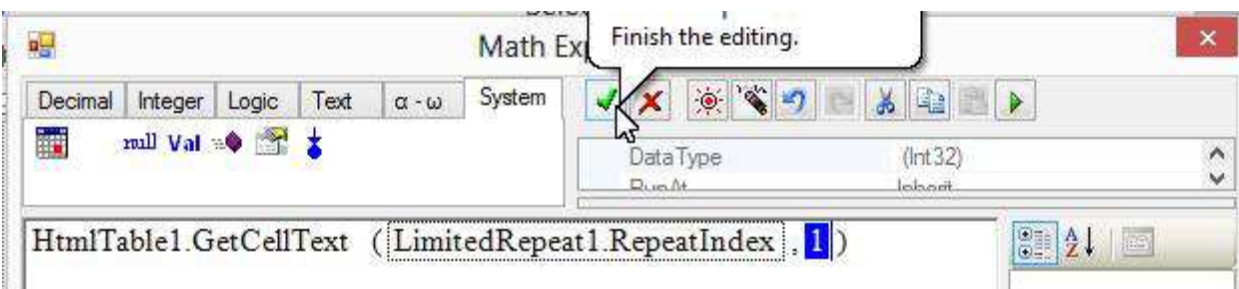
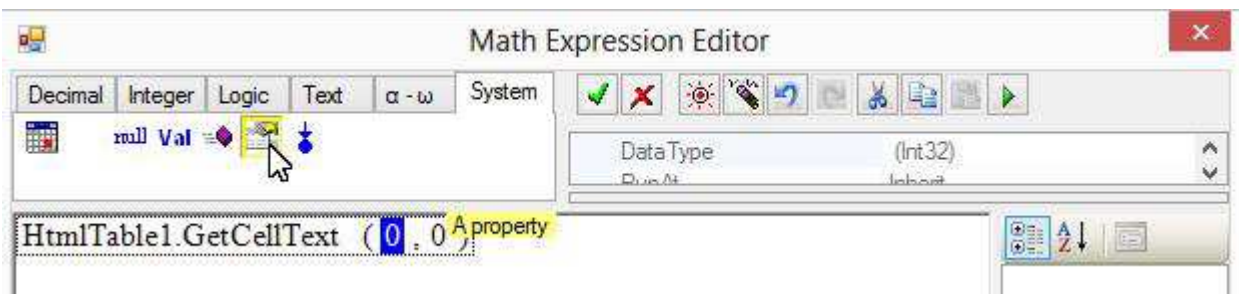


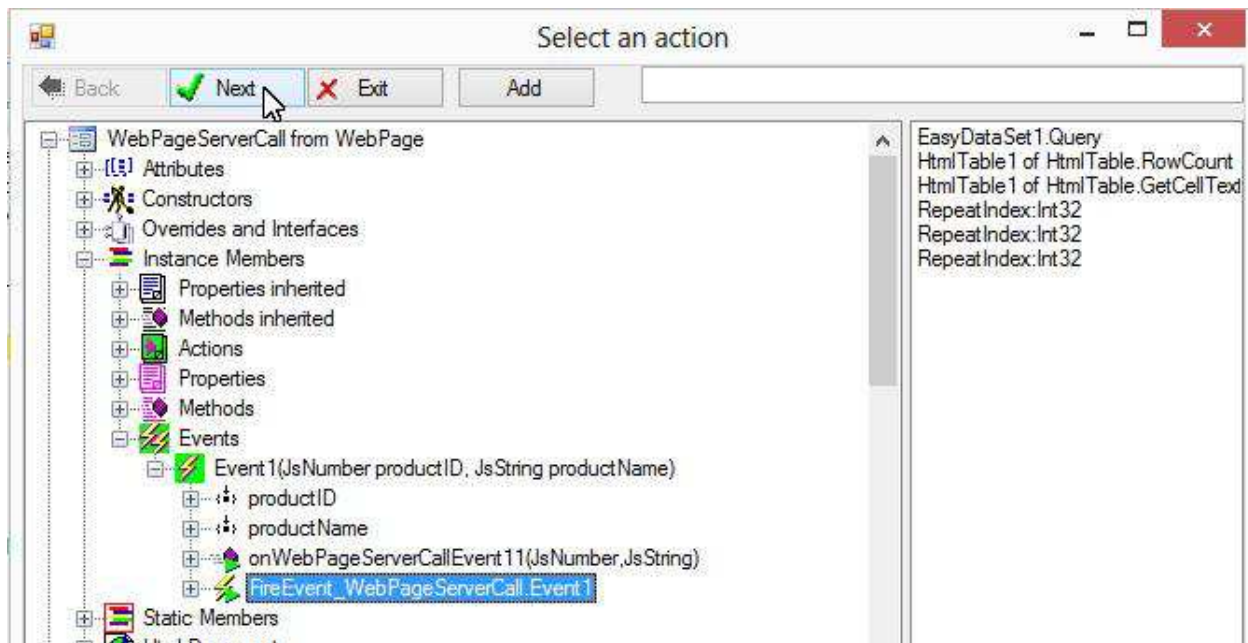
Use GetCellText to supply values for the event:



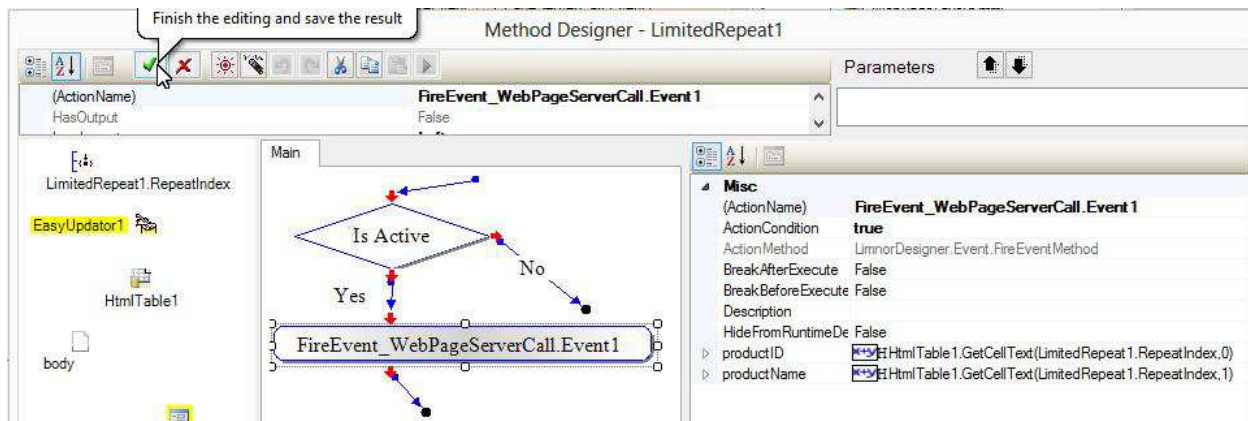






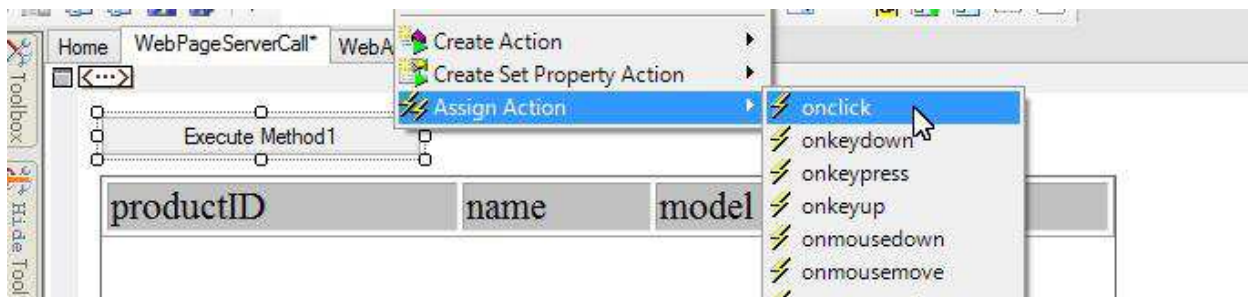


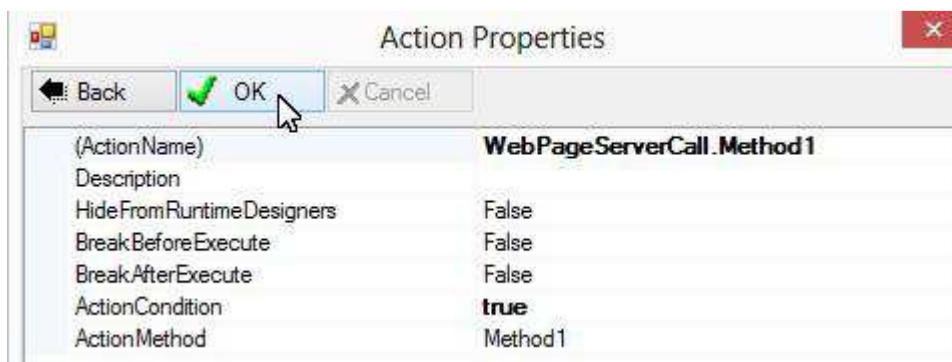
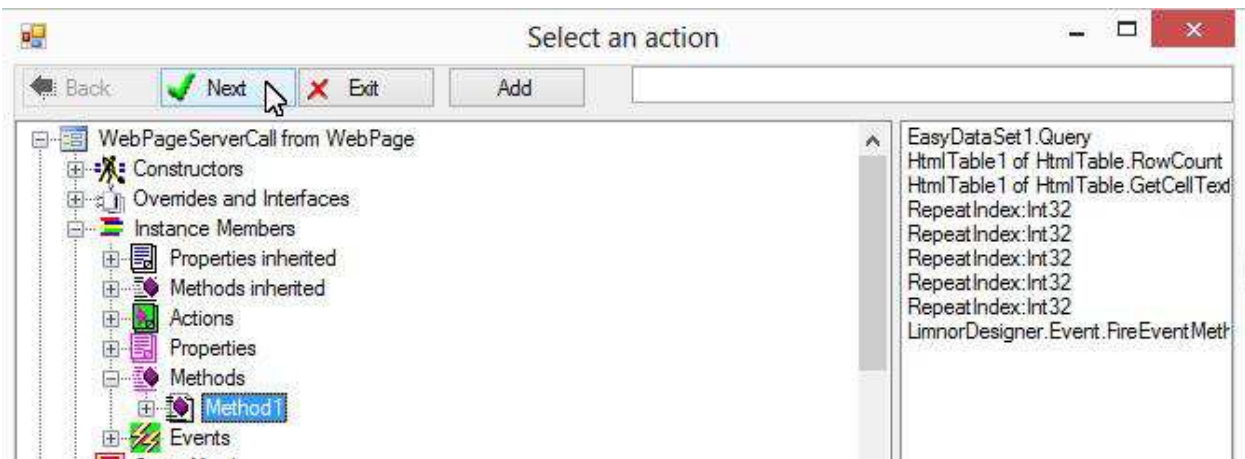
Link this new event-firing action to Yes port of the action:



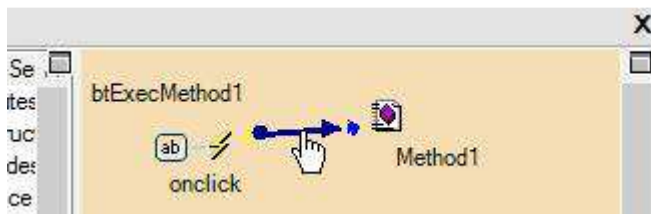
Test

We have created the method. We'll use a button to execute it.

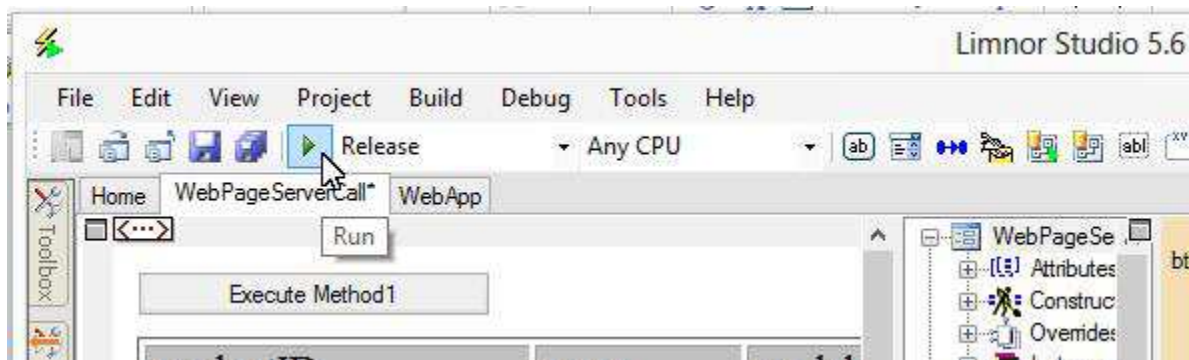




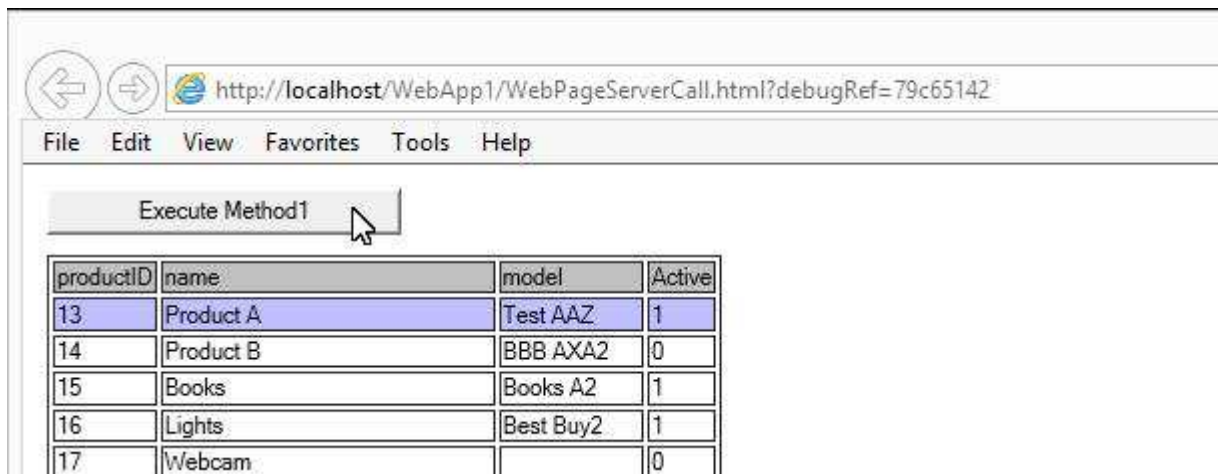
An action is created and assigned to the button:



Click Run button to test it:



A web page appears. We did not show how to add data in the table. Click the button to execute Method1:



If you enabled debug mode then you will see in the debug web page data are being sent to web server for creating new records in database. We can see new records are created:

```
mysql> select * from a;
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id: 38
Current database: test1

Empty set (1.08 sec)

mysql> select * from a limit 10;
+----+-----+
| ID | Name                |
+----+-----+
| 13 | Product A           |
| 15 | Books               |
| 16 | Lights              |
| 18 | waa                 |
| 19 | eosd                |
| 20 | <font color=red>Computer</font> |
| 21 | gshi*               |
| 22 | >iqF-               |
| 23 | <;&B1               |
| 24 | ?$<.<              |
+----+-----+
10 rows in set (0.00 sec)
```

Conclusion

The sample presented above is for showing a way of executing server actions from a client method. It is not for showing a best practice.

Please send your feedback to support@limnor.com