

---

# Control of Row Deletion

---

## Contents

Introduction .....	1
Check Record Existence .....	1
Handle Event UserDeletingRow .....	2
Create event handler .....	2
Get cell of the row .....	3
Query database.....	5
Check database record existence .....	7
Cancel row deletion .....	8
Feedback .....	9

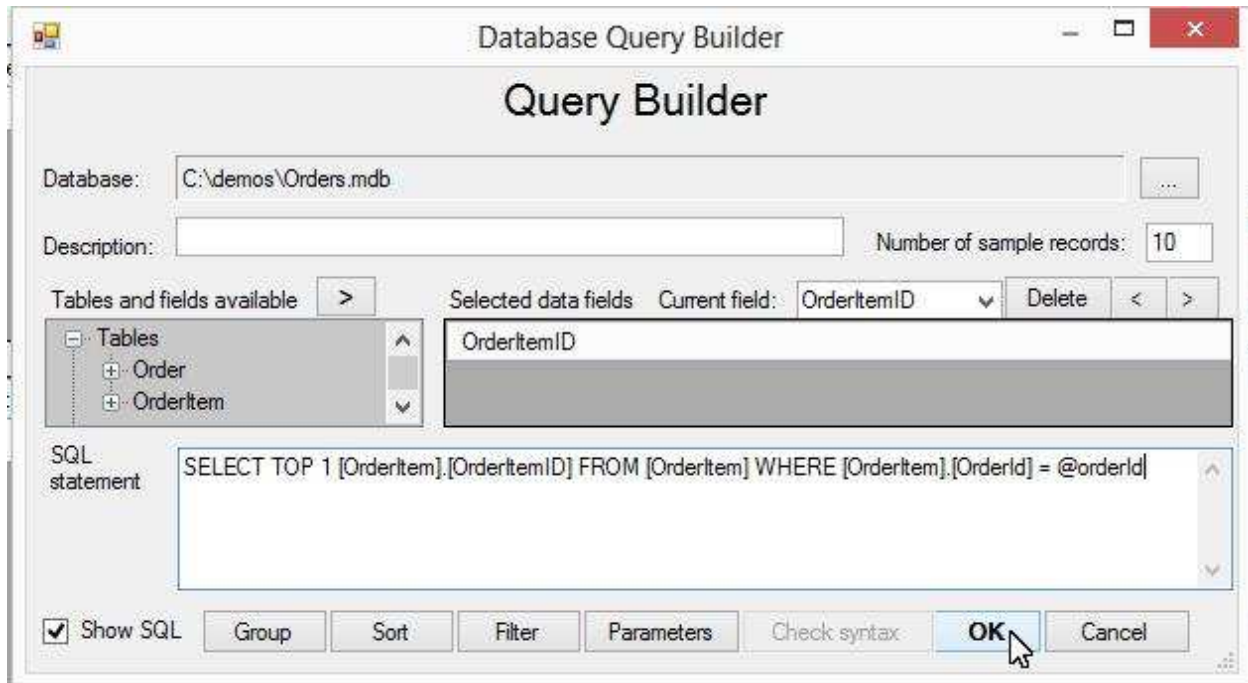
## Introduction

When you use a DataGridView or an EasyGrid to present data to users, suppose you want to allow or disallow deleting of a row based on contents of the row, you may handle event UserDeletingRow to check the contents of the row to determine whether to cancel the row deletion.

One of such scenarios can be that an EasyGrid displays records from an “Order” table; an “OrderItem” table uses an “OrderId” field to link to the “Order” table, forming an enforced one-to-many relationship; cascade-deletion is not enabled for the “Order” table. Deleting an “Order” record generates a database error if there are “OrderItem” records linked to the “Order” record. Therefore, we want to not allow deleting an “Order” record if there are “OrderItem” records linked to it and we want to allow deleting an “Order” record if there is not an “OrderItem” record linked to it. We can do such programming by handling event UserDeletingRow.

## Check Record Existence

We can use an EasyDataSet for finding out if there are “OrderItem” records linked to an “Order” record. Set its SQL to query for “OrderItem” records.



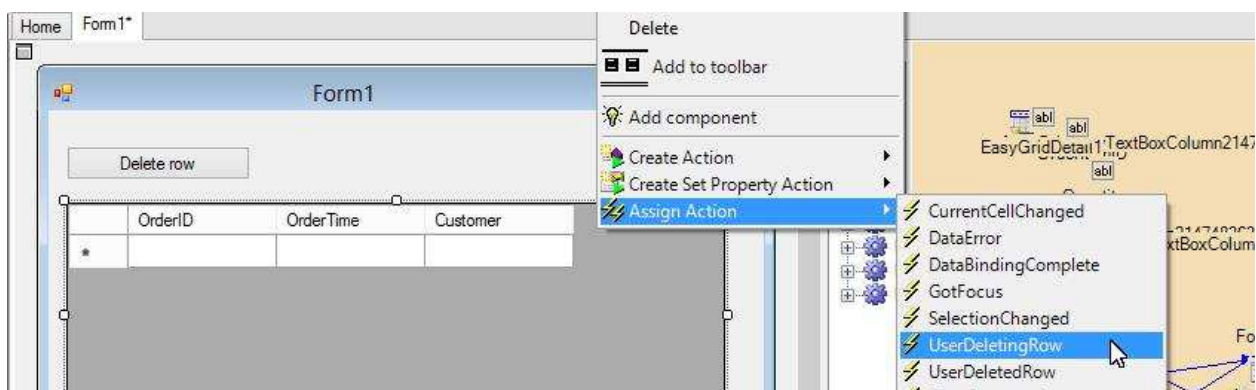
Note that because we just want to check if there is at least one “OrderItem” record linked to the “Order” record, we use “TOP 1” to just query for one record so that the query will run fast. “TOP 1” is specific for Microsoft Access, not a standard SQL syntax. For MySQL, “LIMIT 1” at the end of the SQL should be used.

Note that this is a query with a filter of OrderId=@orderid. “@orderid” is a parameter. Later when we handle event UserDeletingRow, the value of “@orderid” will be provided by the “Order” record to be deleted.

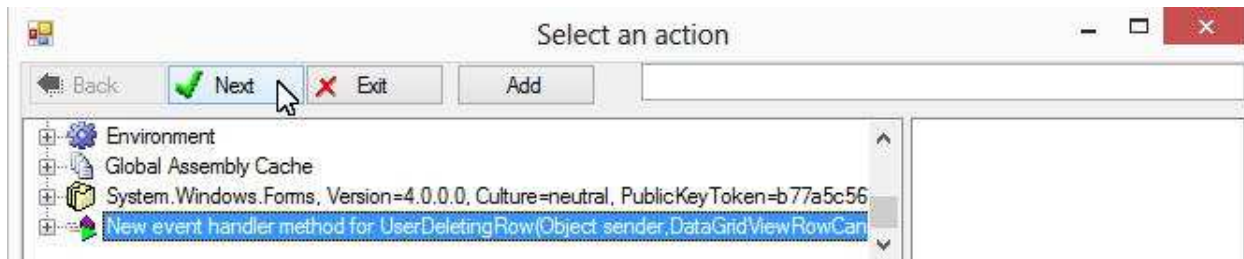
## Handle Event UserDeletingRow

### Create event handler

Right-click the EasyGrid showing “Order” records; choose “Assign Action”; choose event “UserDeletingRow”:



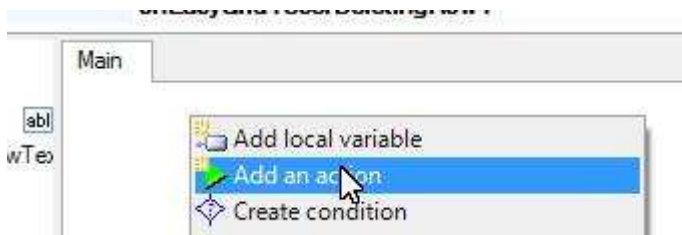
Select “New event handler method for UserDeletingRow”:



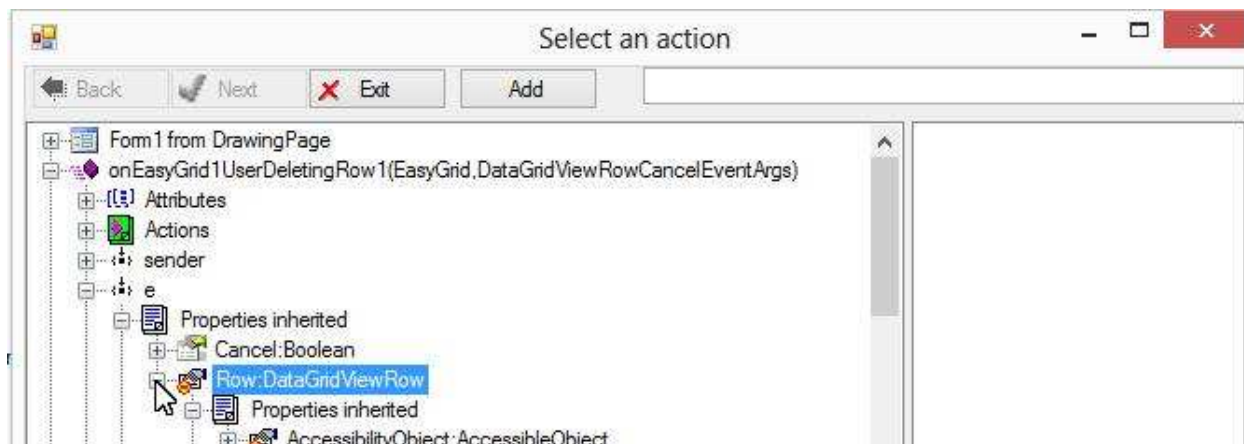
A method editor appears for editing the new handler. We are going to add actions in this handler to cancel row-deletion if there is an “OrderItem” record linked to the “Order” record being deleted.

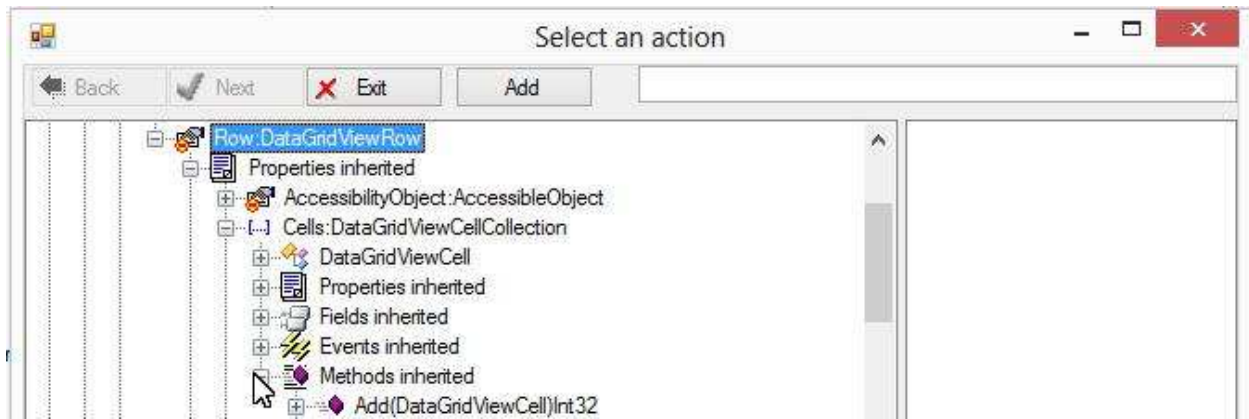
### Get cell of the row

In this sample project, the first column of the EasyGrid displaying “Order” records is for “OrderId” field. We want to use the value of “OrderId” to search “OrderItem” table to see if at least one record exists. The value of “OrderId” is the value of the first cell of the row being deleted. We need to get the first cell first. Add an action to get the first cell:

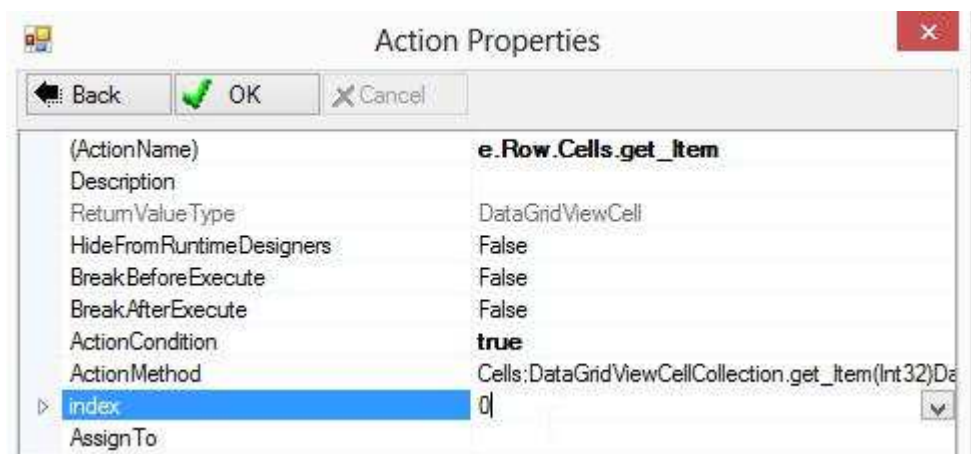


Select get\_item of a Cells property of row being deleted:

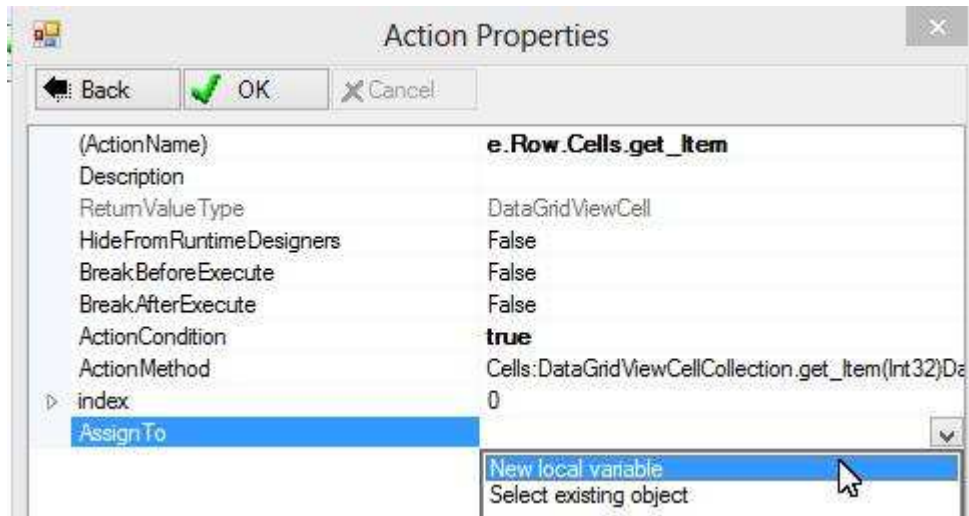




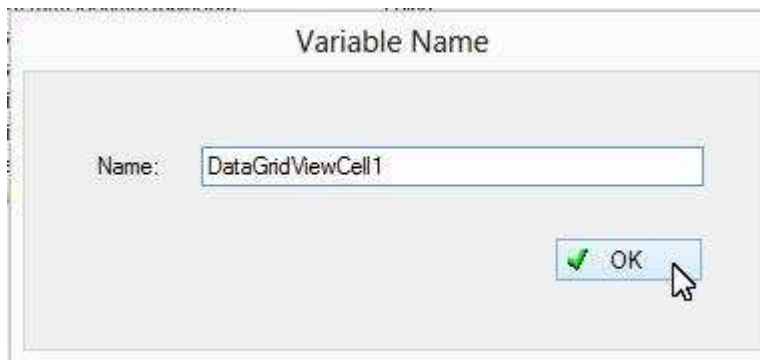
“index” indicates the index of the cell. 0 indicates the first cell; 1 indicates the second cell; etc.



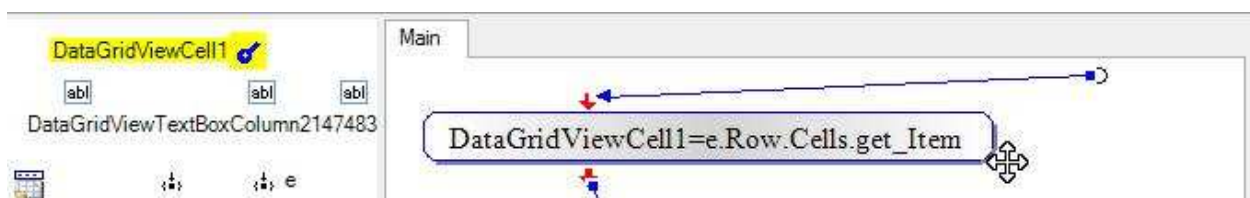
Choose “New local variable” for “AssignTo”:



Choose a variable name:

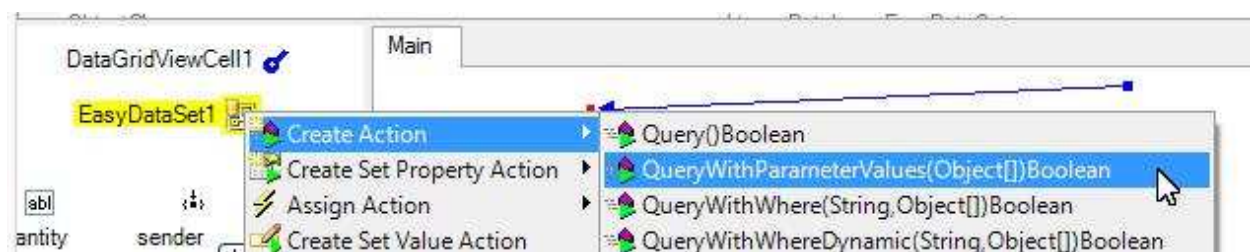


The action and a variable appear. The variable is the first cell:

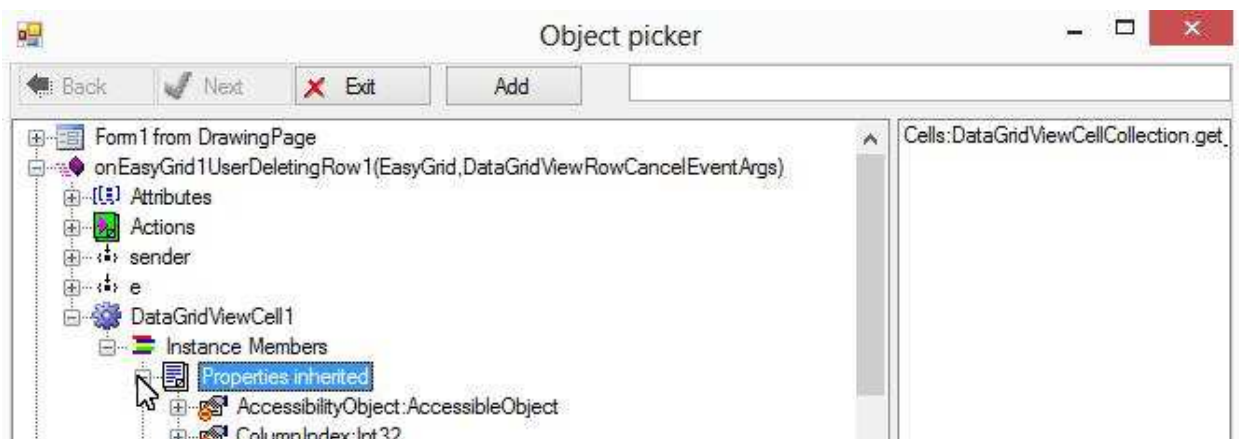
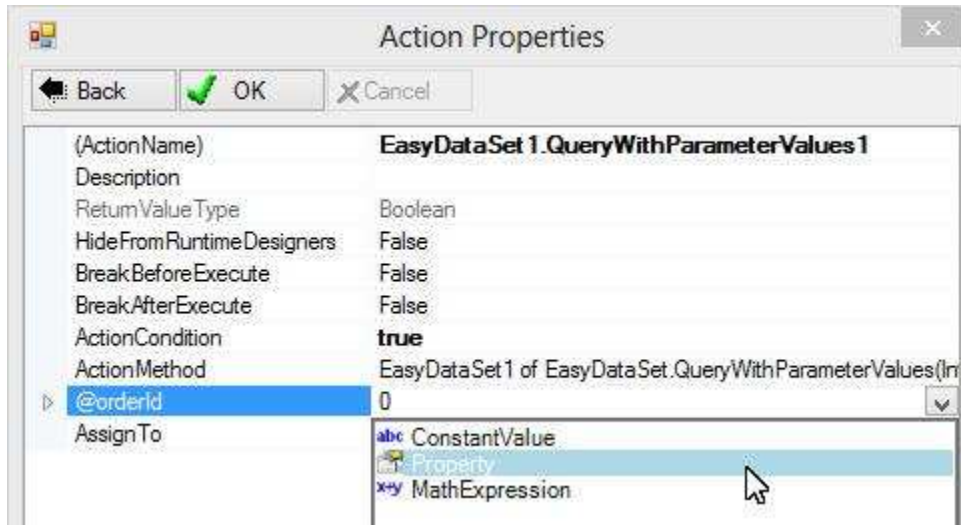


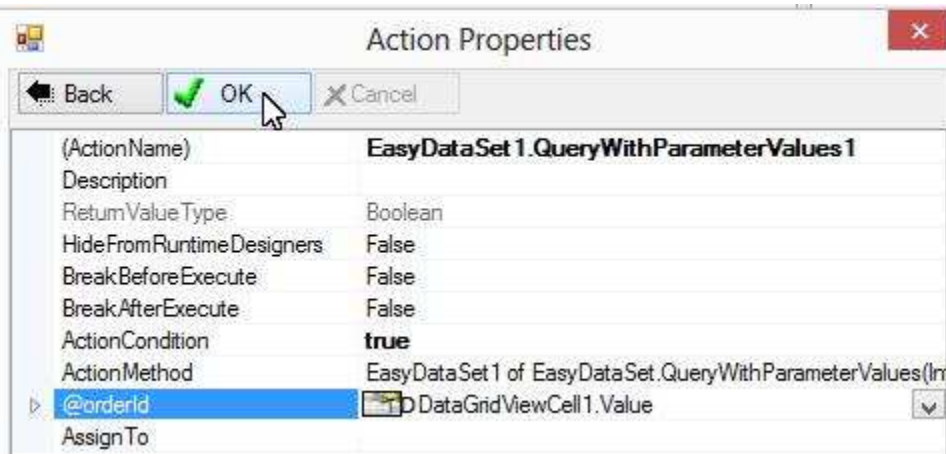
## Query database

We can create a "QueryWithParameterValues" to check if there is an "OrderItem" record with the order id:



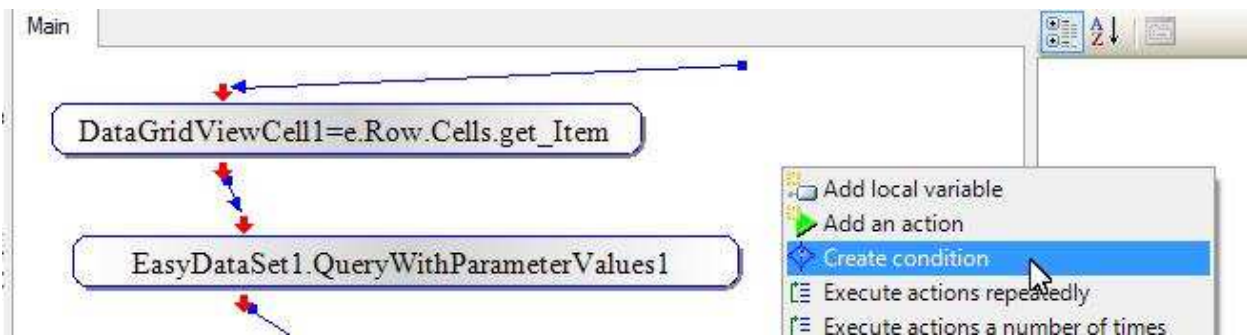
Pass the cell value to “@orderId”:



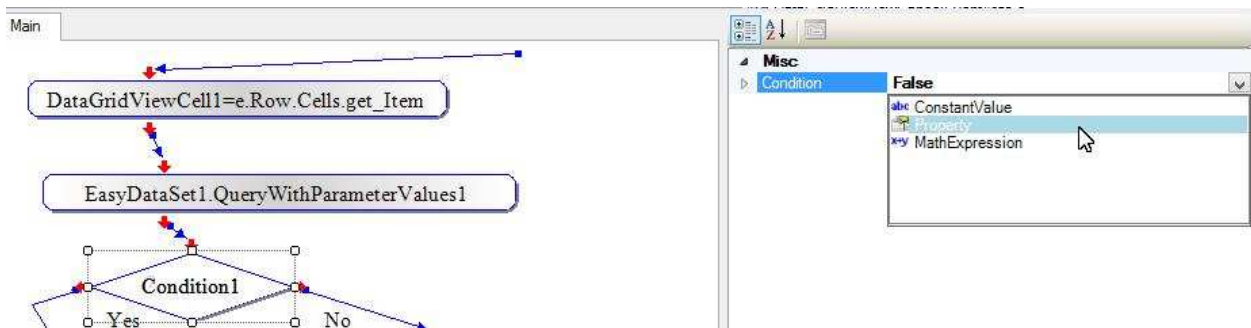


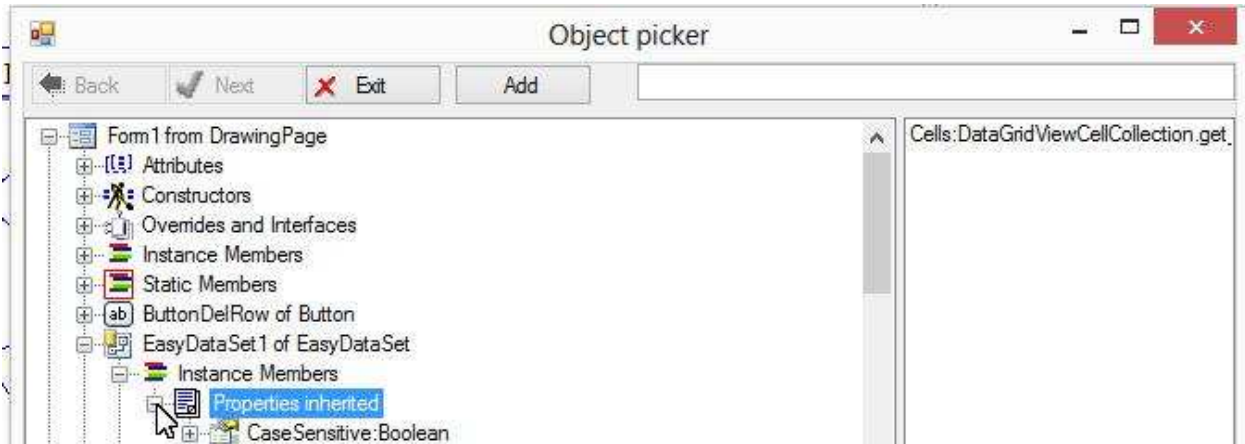
## Check database record existence

Add a Condition to check existence of an "OrderItem" record:



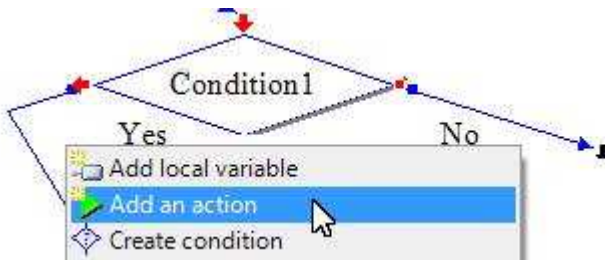
We use "HasData" property of the EasyDataSet as the condition:





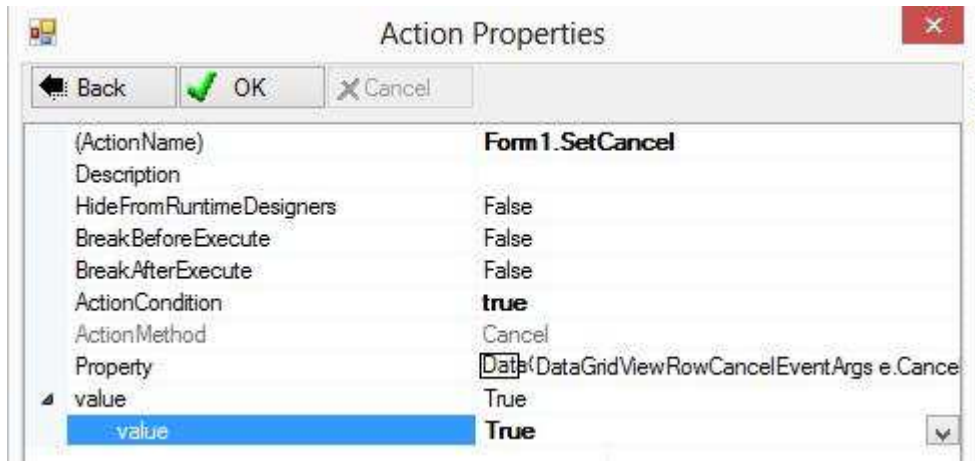
## Cancel row deletion

Add an action to set Cancel property to True to cancel the row deletion:

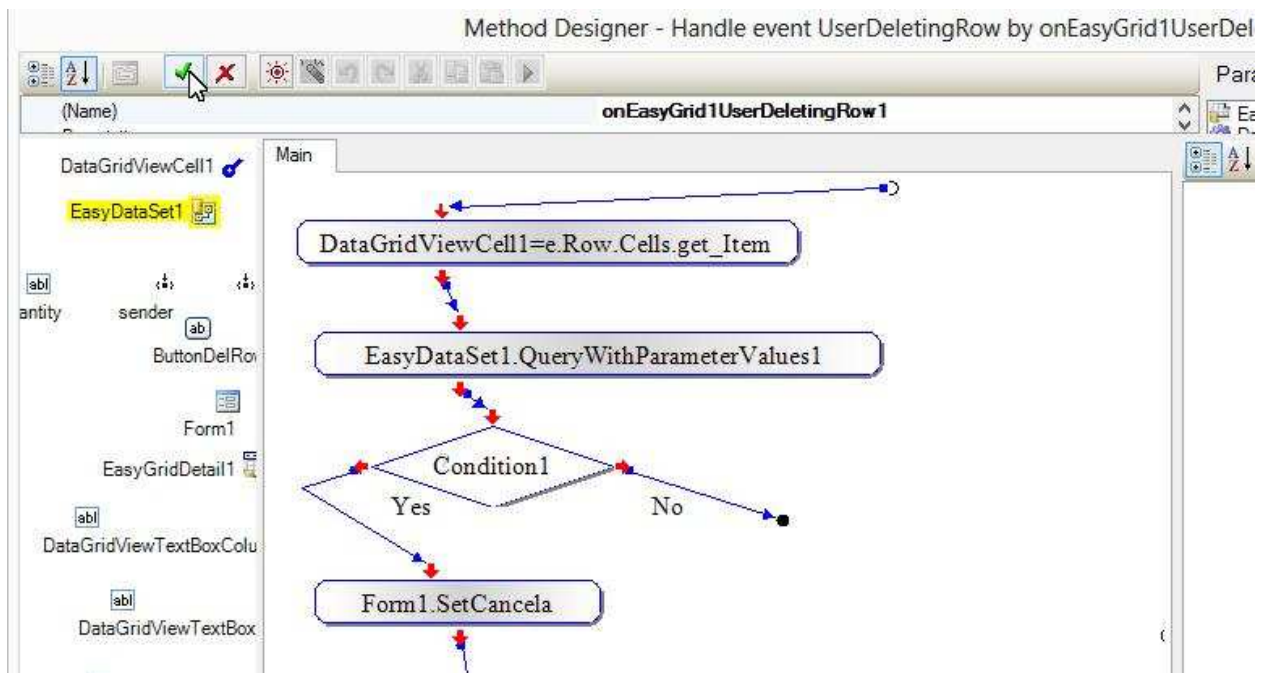


Set "value" to True:





That is all we need for this event handler:



## Feedback

Please send your feedback to [support@limnor.com](mailto:support@limnor.com)