

# Make Report

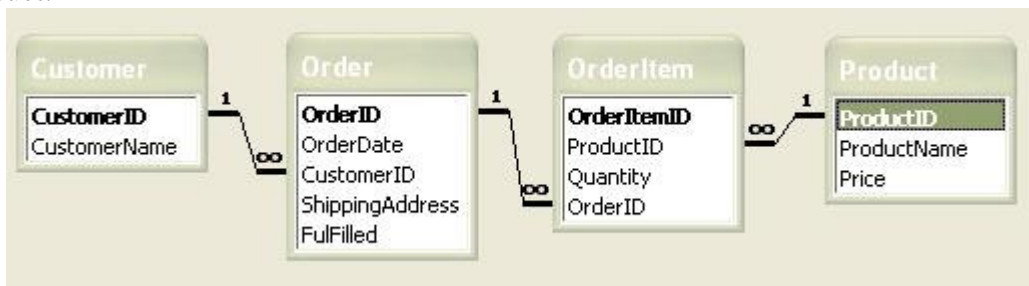
1	Introduction.....	1
2	Sample Database.....	1
3	Make Invoice .....	3
3.1	Create an Invoice Page.....	3
3.2	Make Invoice Query .....	3
3.3	Calculate Total Amount.....	6
3.4	Design Report .....	10
4	Show Report.....	12
5	Print Report.....	13

## 1 Introduction

If you can use Limnor to develop database applications then you do not have much to learn to use Limnor to make reports. You use the same database query building skills and use data-binding to display data.

## 2 Sample Database

This sample application use a very simple database to do our demo: a customer may make many orders. One order may have many order items. Each order item is for one product:



Note that the first field of each table (in bold) is an auto-number. The value of an auto-number is given by the database and cannot be changed. This is for easy of enforcing the table relationship. You may add your own “code” or “id” fields to tables. But then you are responsible to maintain their uniqueness.

These tables are very simple in that very few fields are used in each table. Of course you may add more fields to each table to save more attributes for each record.

We enter two customers and 4 products for our demo:

Customers		
CustomerID	CustomerName	
1	Mike	
2	John	
*		

Products			
ProductID	ProductName	Price	
1	Apple	1.2	
2	Orange	0.8	
3	Banana	1.5	
4	Pear	2.1	
*			

Make one order to order 3 products:

Orders					
OrderID	OrderDate	CustomerID	ShippingAddre	FulFilled	
1	1/11/1960	1	123 Main Stre	<input type="checkbox"/>	
2	1/18/1960	2	7788 Broadwa	<input type="checkbox"/>	
*					

Order Items				
OrderItemID	OrderID	ProductID	Quantity	
1	1	1	3	
2	1	2	2	
3	1	3	6	
*				

Make another order to order 2 products:

Orders					
OrderID	OrderDate	CustomerID	ShippingAddre	FulFilled	
1	1/11/1960	1	123 Main Stre	<input type="checkbox"/>	
2	1/18/1960	2	7788 Broadwa	<input type="checkbox"/>	
*					

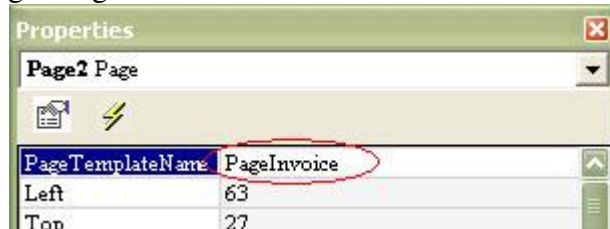
  

Order Items				
OrderItemID	OrderID	ProductID	Quantity	
1	2	2	5	
5	2	3	8	
*				

### 3 Make Invoice

#### 3.1 Create an Invoice Page

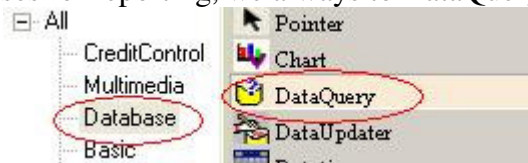
Every page can be used as a report page. We create a new page to show invoice. For creating new pages, see <http://www.limnor.com/downloads/Lesson4.pdf>  
We name the new page “PageInvoice”:



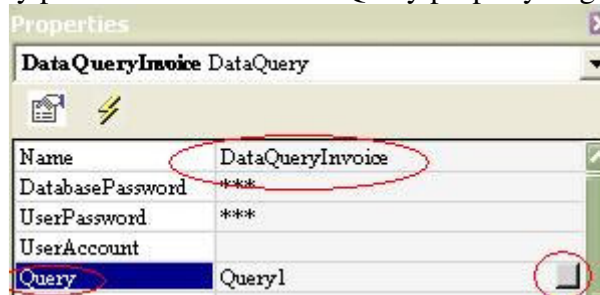
#### 3.2 Make Invoice Query

Each business has its own unique requirements for an invoice. Typically, you want to show product name, price, quantity and amount.

To get data from databases for reporting, we always use DataQuery performers:



Let create a DataQuery performer and build its Query property to get invoice data:



First, double-click OrderID because for an invoice we must have an order:



Double-click OrderItemID because we want to list order items:



Because OrderItemID is from OrderItem table, and OrderID is from Order table, it will ask you how you want to link these two tables. We must link them with OrderID:

### Select join fields for the new table

New table name:

Join fields:

	New table Join field	Join-to table	Join-to field
▶	OrderID	Order	OrderID
*			

Join type:

Records in the new table must exist  
 If you select this option, the query will only return those records matching the join fields in the new table

We may add other fields as the invoice requires:

### Database Query Builder

#### Query Builder

Query name:  Database:

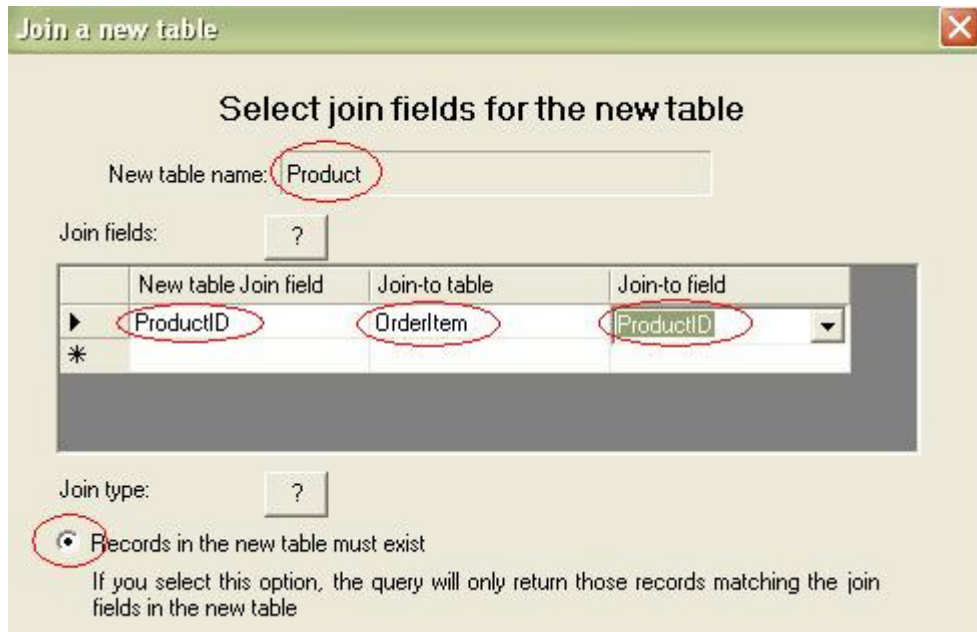
Number of sample records:

Tables and fields available:  Selected data fields:  Current field:

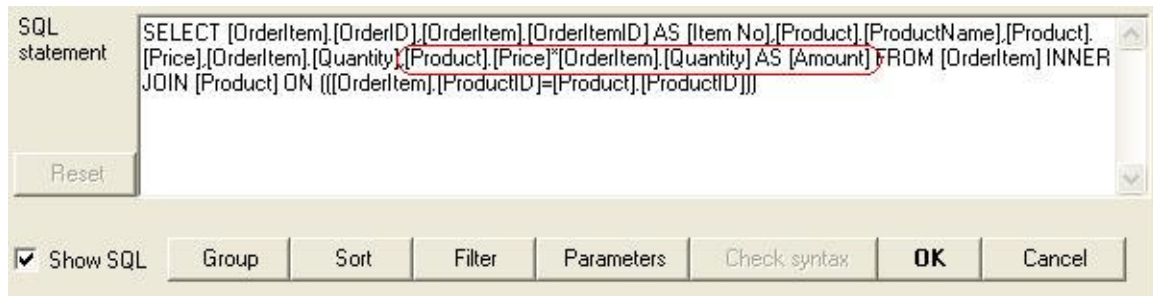
	OrderID	Item No	ProductName	Price	Quantity	Amount
▶	1	1	Apple	1.2	3	3.6
	1	2	Orange	0.8	2	1.6
	1	3	Banana	1.5	6	9
*						

Show SQL

Note that when you add ProductName, it will ask you for table link again. You must use ProductID to link the Product table with the OrderItem table:



The “Amount” field is a calculated field. You need check “Show SQL” and manually enter this field:



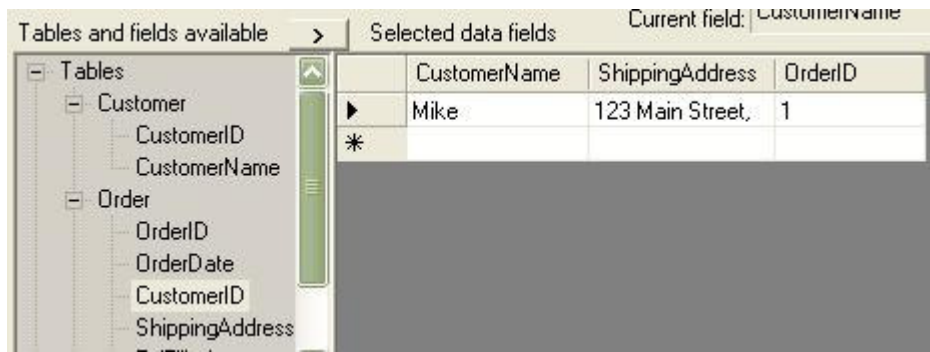
You can see that after the Quantity field, we type in the following text:

**,[Product].[Price]\*[OrderItem].[Quantity] AS [Amount]**

This query lists all orders. If we want to search by OrderID, we may set a filter to do that. Click “Filter” button, choose OrderItem table, choose OrderID field, choose “=” for Operator; and select “Param”, type in a parameter name, say, “OrderID”, click “Add filter” button:



We use another DataQuery performer to get customer name and shipping address. We also use OrderID to do the filtering:



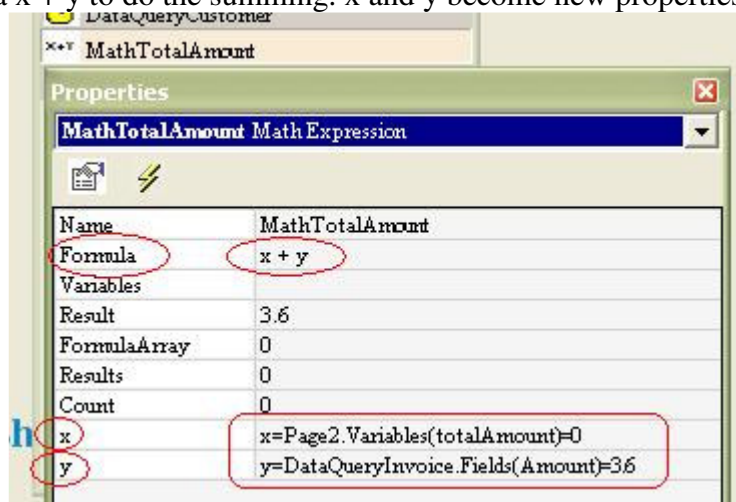
At runtime, we may use actions to set this parameter, @OrderID, to different values to get different invoices.

Now we have invoice data from the database. We will design page layout to show the data.

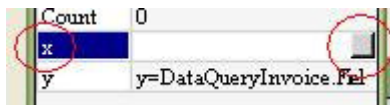
### 3.3 Calculate Total Amount

A Math Expression performer is used to sum up all Amount field of the order item query. The calculation result is saved in a variable named totalAmount. For more information on using variables, see Lesson 12 of the Tutorial.

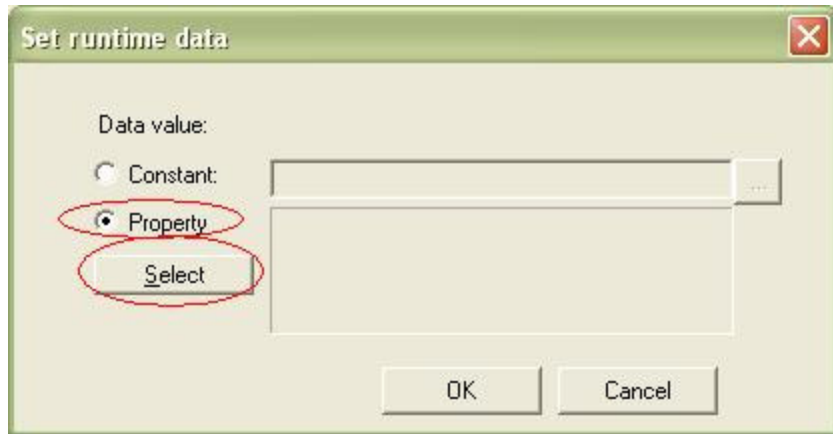
We use formula  $x + y$  to do the summing.  $x$  and  $y$  become new properties:



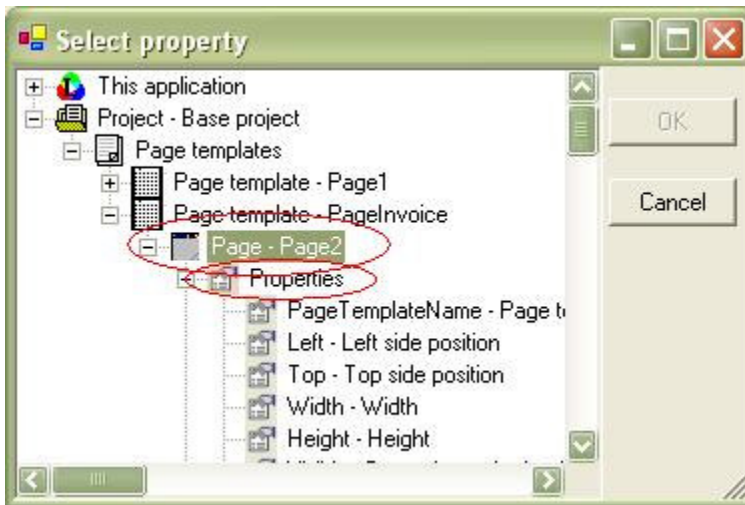
We link property  $x$  to the variable totalAmount, and  $y$  to the Amount field of the invoice query.



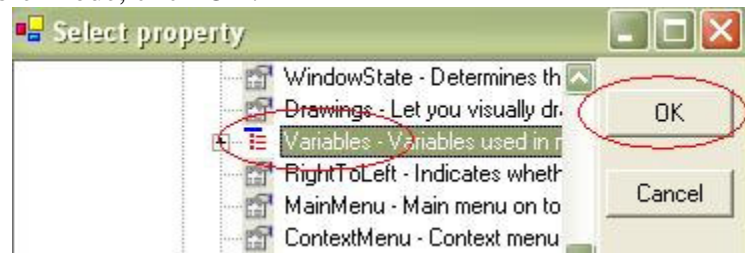
Check "Property", click "Select" button.



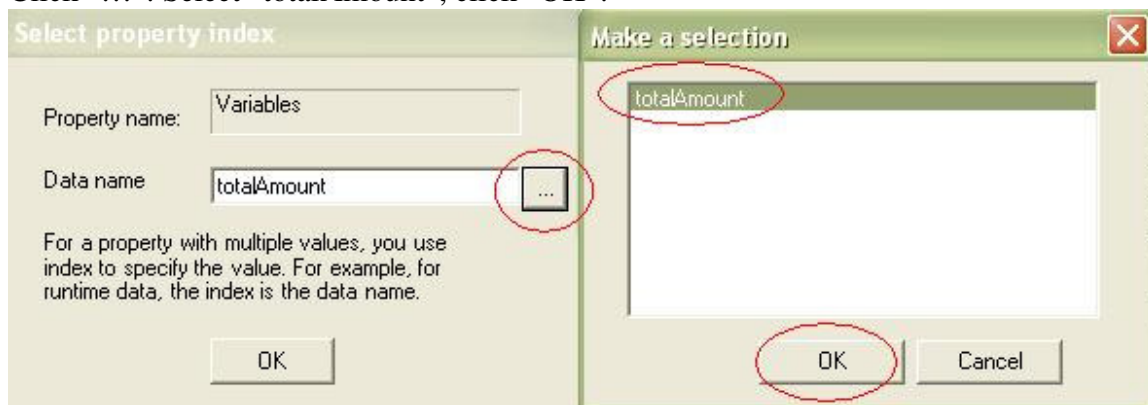
Choose "Page2":




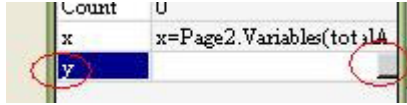
Find its "Variable" node, click OK:



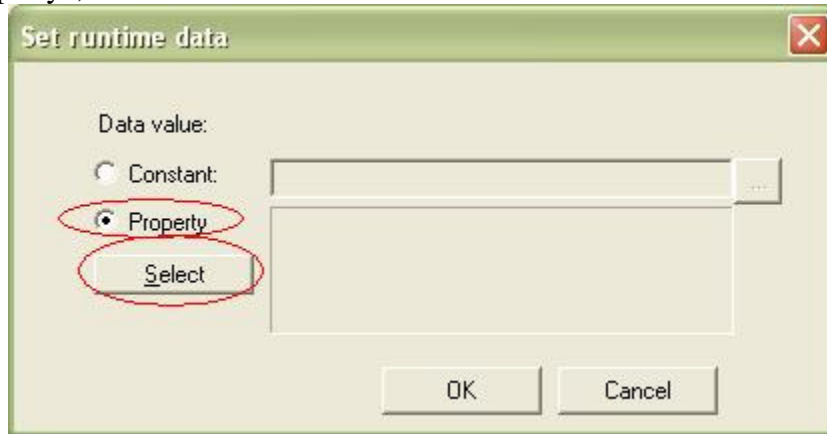
Click "...". Select "totalAmount", click "OK":



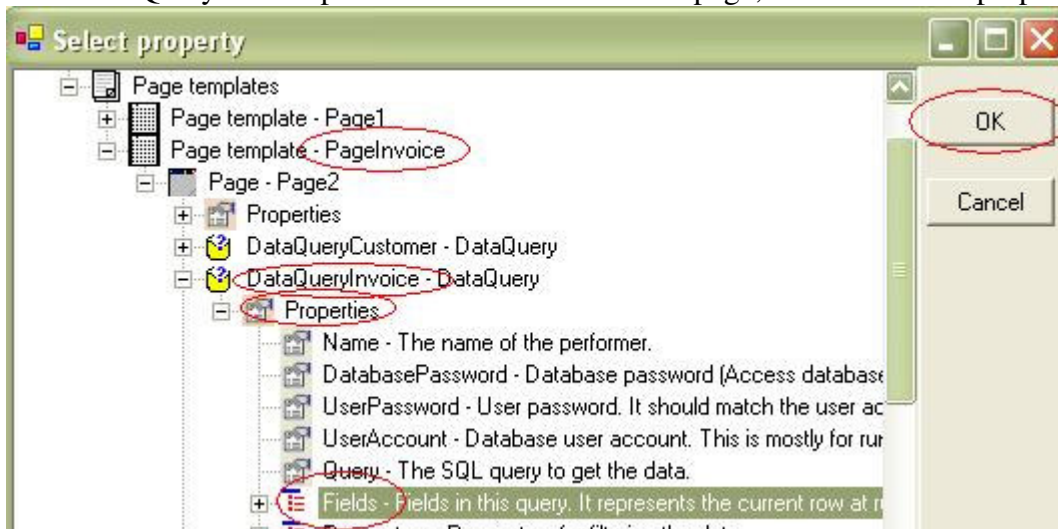
Now variable x is linked to variable “totalAmount”.  
Choose variable y and click :



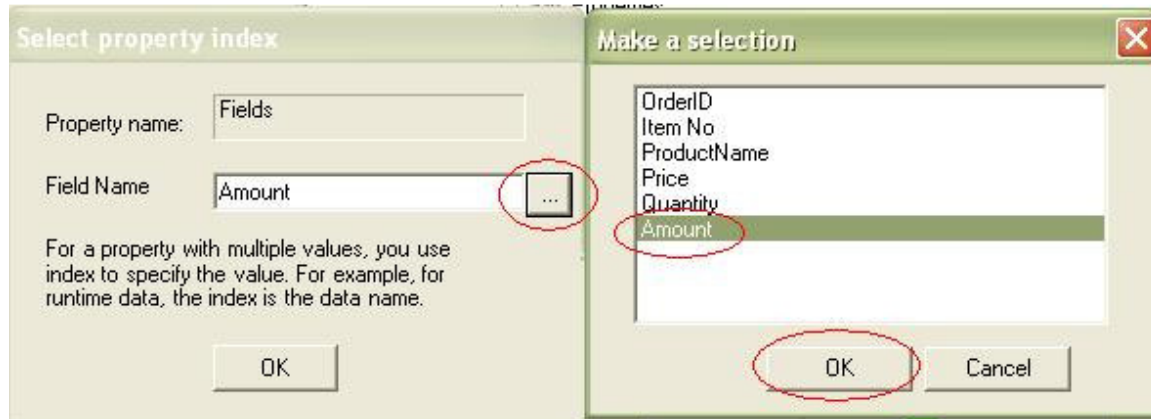
Check “Property”, click “Select” button:



Find the DataQueryInvoice performer under and invoice page; select its Fields property:

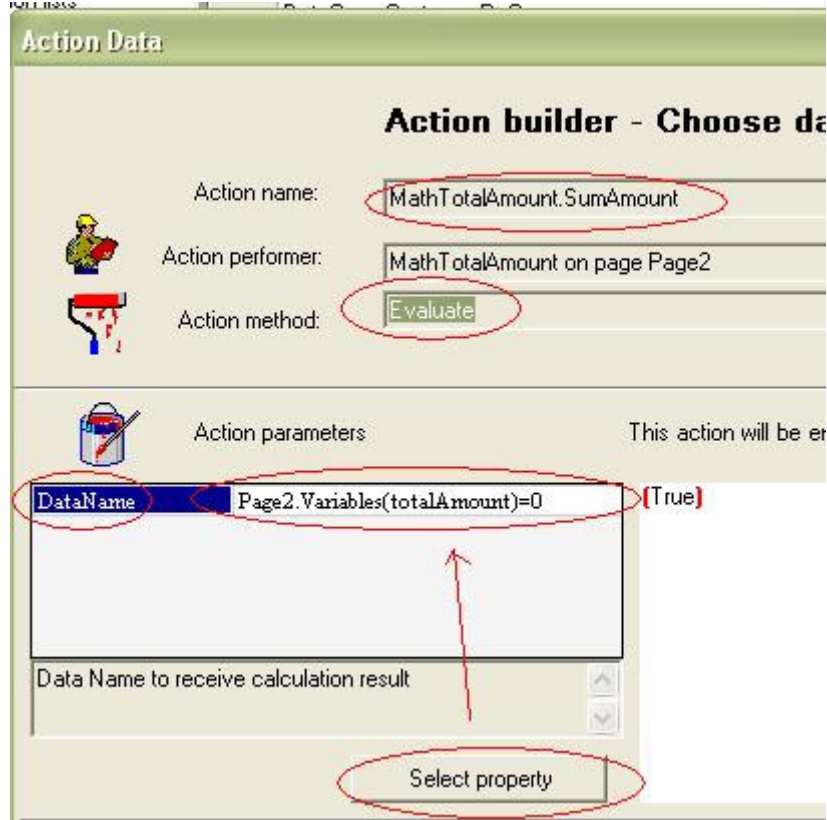


Choose Amount field:

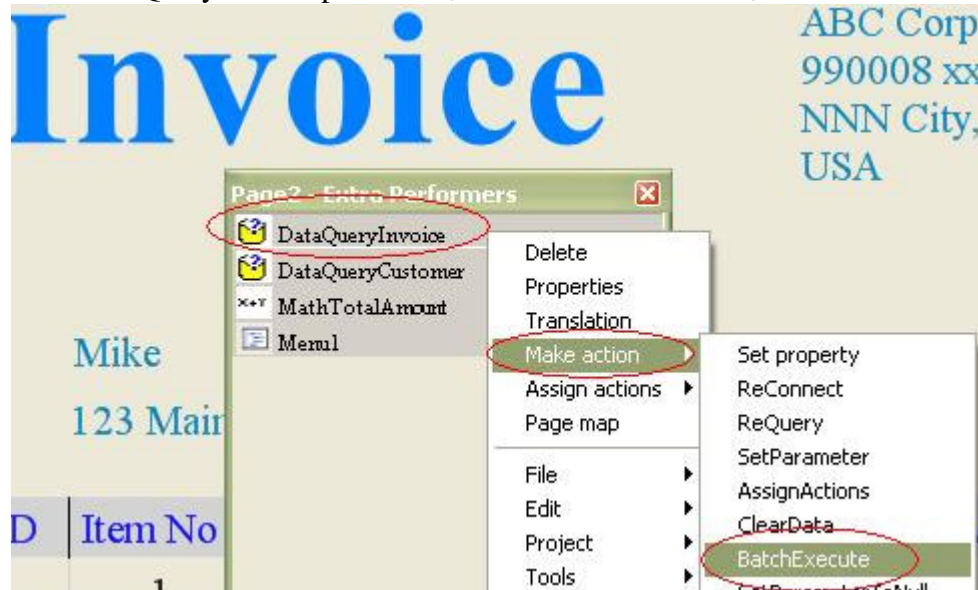


Now variable y is linked to field “Amount” in the invoice item.

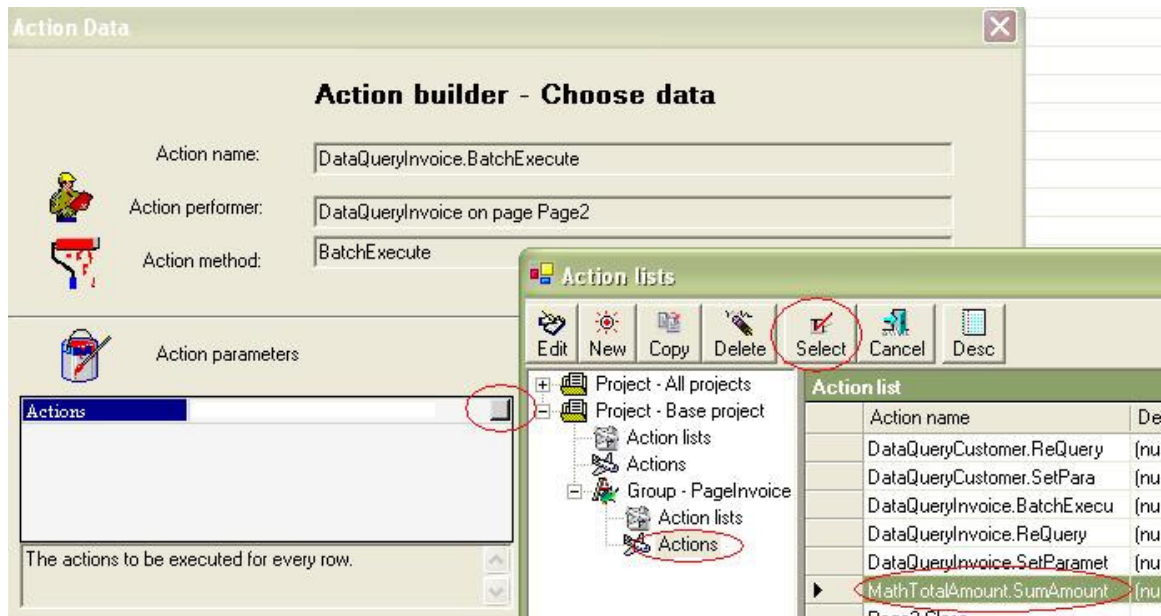
We use the Evaluate method of the Math Expression to create an action to evaluate the formula. The action requires a DataName parameter. This parameter indicates where to save the evaluation result. We want to save the result back to the variable totalAmount:



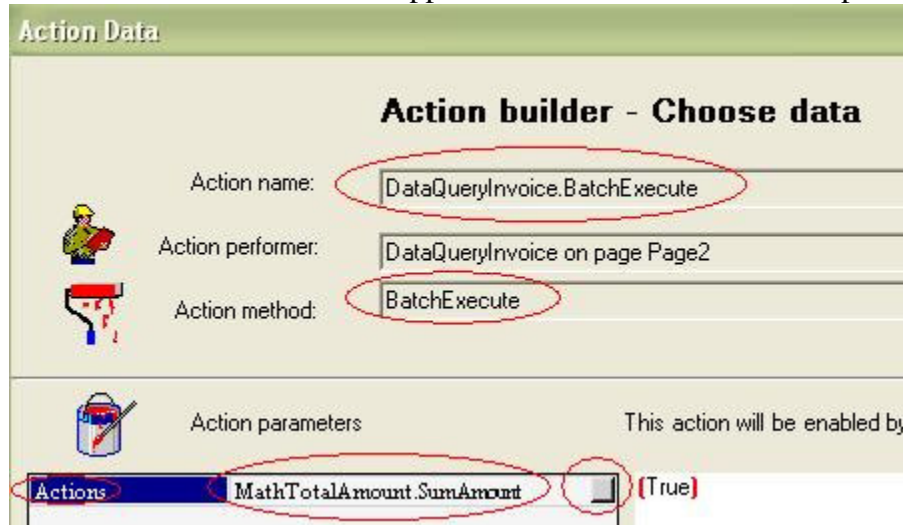
Now, if for every order item we execute this action then all of the order item amounts will be summed up into the variable totalAmount. This can be done using the BatchExecute method of the DataQuery. We use this method to create an action. Right-click on the DataQueryInvoice performer, choose “Make action”; select “BatchExecute”:



Accept the default action name. The Action Data dialogue box appears. This action requires an “Actions” parameter. We use the summing action we just created as this parameter. Click  to select the method MathTotalAmount.SumAmount. Click Select button:



Action MathTotalAmount.SumAmount appears as the value for “Actions” parameter:

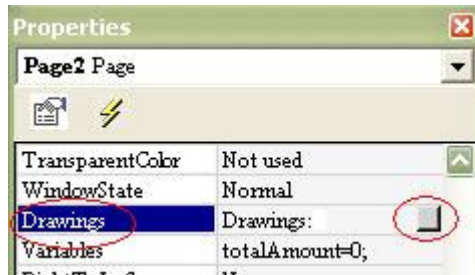



When this BatchExecute action is executed, it iterates through all order items, for each order item, it executes action, MathTotalAmount.SumAmount, once.

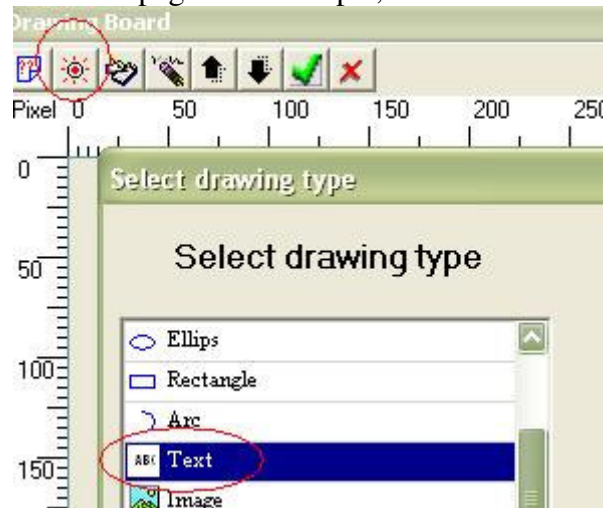
This action is included in the action list, LoadInvoice. When loading the invoice, this action, DataQueryInvoice.BatchExecute, will be executed. See [Show Report](#).

### 3.4 Design Report

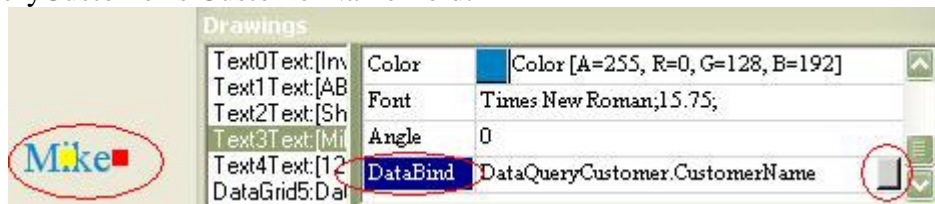
To design report layout, set the page’s Drawings property:



Click  to add controls to the page. For example, add Text controls:

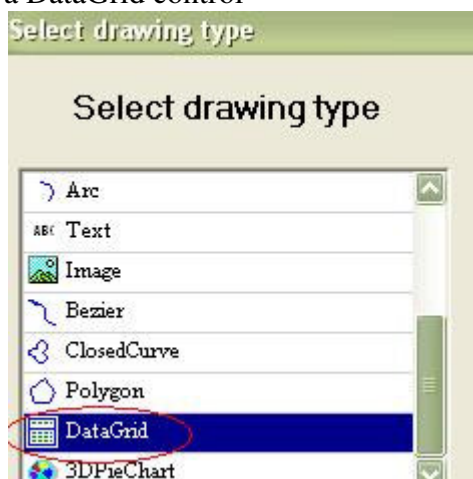


To make a Text control display customer name, set its DataBind property to the DataQueryCustomer's CustomerName field:



In the same way, you can show shipping address.

To show order items, add a DataGrid control

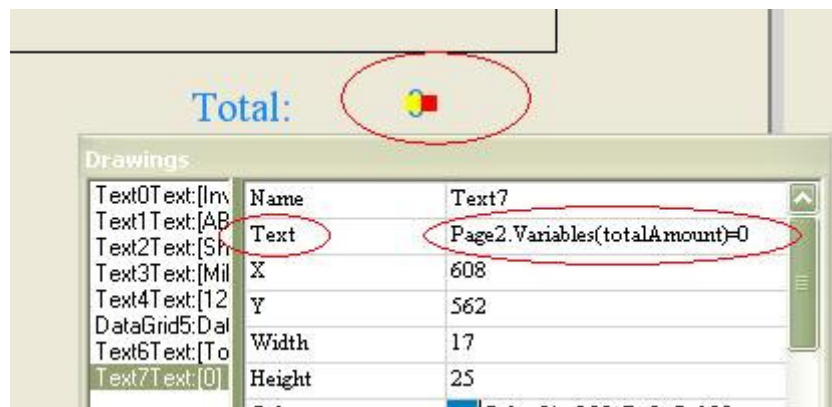


and set its DataQuery property to DataQueryInvoice:

OrderID	Item No	ProductName	Price	Quantity	Amount
1	1	Apple	1.2	3	3.6
	2	Orange	0.8	2	1.6
	3	Banana	1.5	6	9

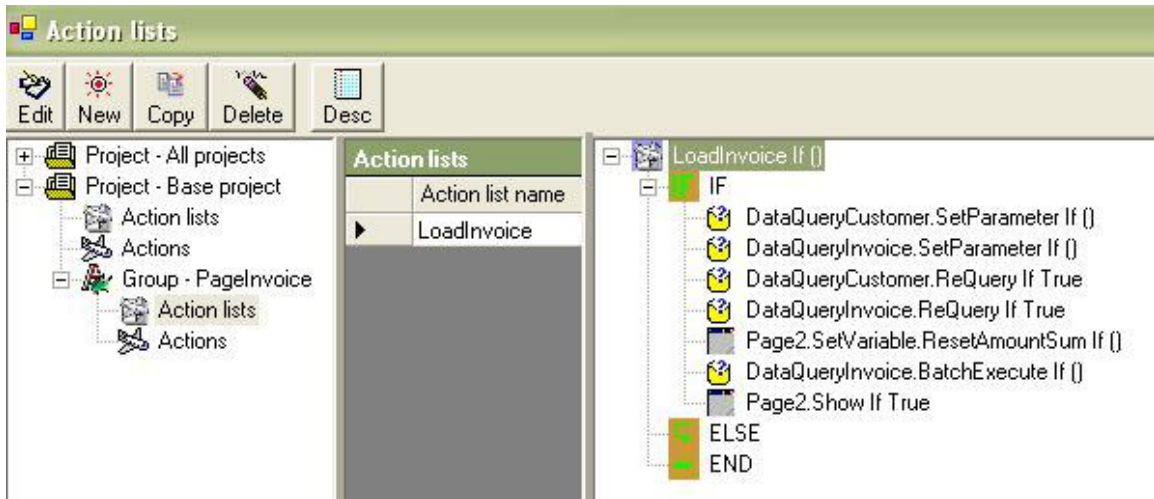
  

Note that for each control on the report page, you may drag blue dots to re-size it and drag the yellow dot to move it and drag the red dot to rotate it. We use a Text to show the total amount by setting its Text property to the totalAmount variable:



## 4 Show Report

On the first page, we use a text box to let the user type in an Order ID and click a button to display the invoice. We assign an action list, LoadInvoice, to the Click event of the button:



The first two actions set the @OrderID parameter of the DataQuery to the Text property of the Text box. The next two actions fetch the invoice data from the database based on the value of the @OrderID. The next two actions sum the order item amounts and save the sum to a variable, totalAmount.

The last action, Page2.Show, display the invoice page.

## 5 Print Report

We added a File menu on the Invoice page to do Print-Preview and Print:



Print-Preview action is made from PrintPreviewReport method. Print action is made from PrintCurrentReportPage method.