Web Data Repeater

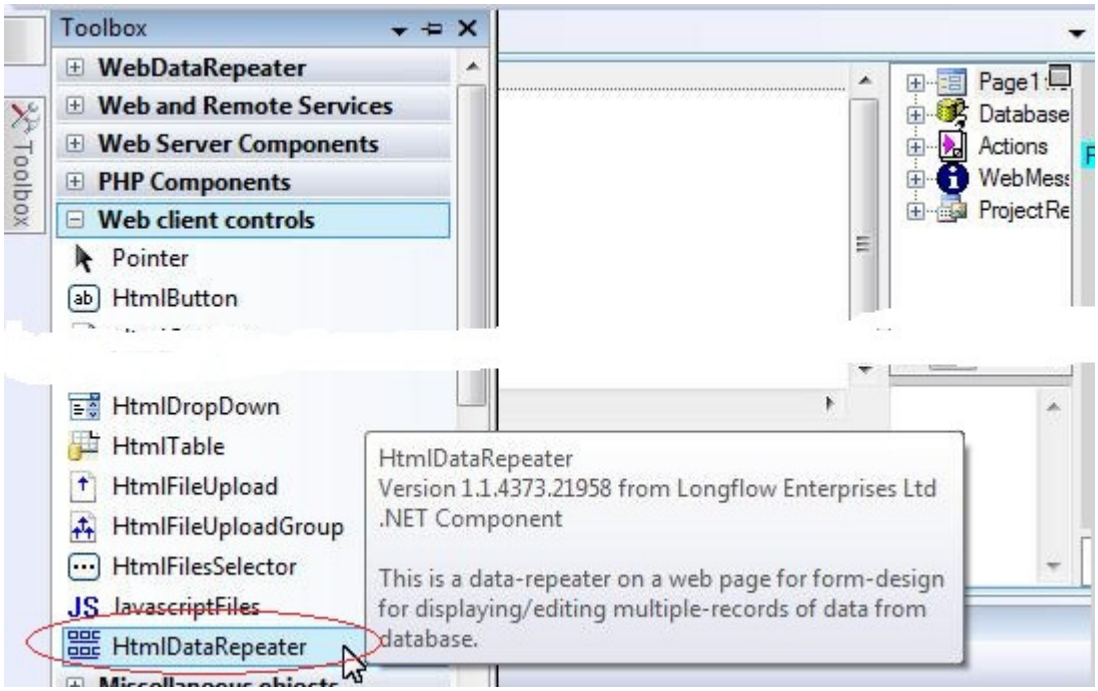
Contents

Introduction

In chapter "Bind-data to single field" of "Web Database Application" (http://www.limnor.com/support/webDatabaseProgramming.pdf), we see that a form can be designed with data-bound controls. Data from database are automatically displayed on the controls. The user may modify data on the controls and the data modifications can be saved back to database.

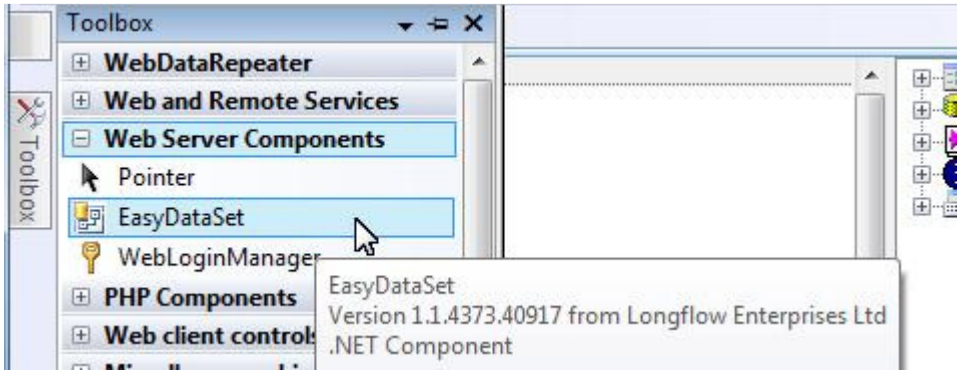
In such arrangement, one record is displayed on one web page.

Data Repeater allows you to design the form with data-bound controls, but the same design can be repeated on one web page. Thus many records can be displayed on one web page.

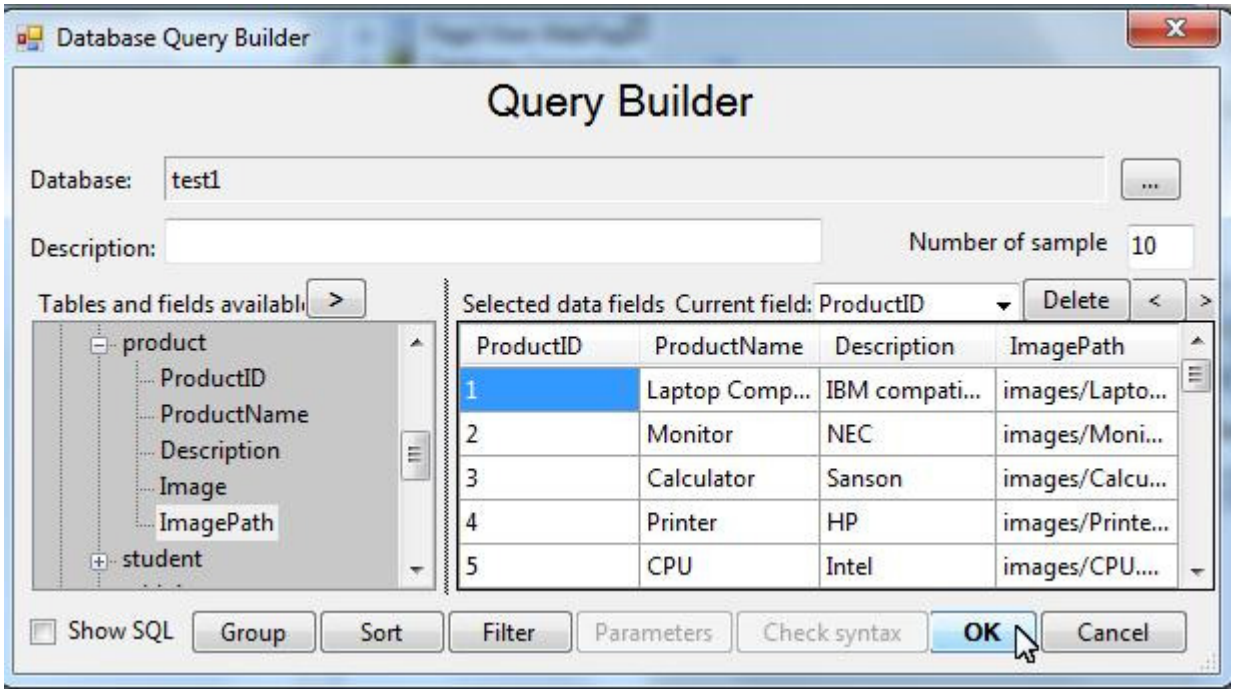


Data Query

Add an EasyDataSet to a web page for getting data from web server:

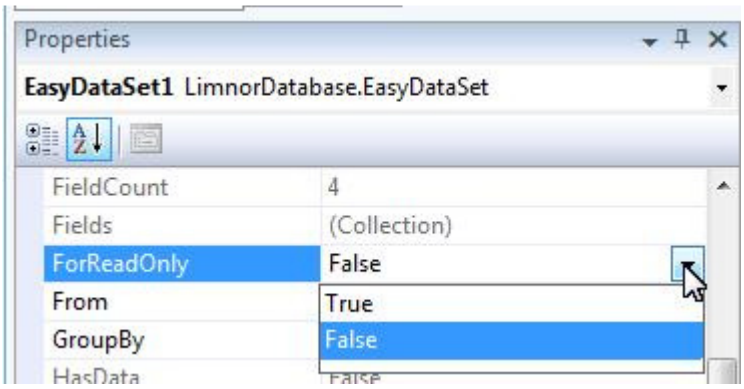


Set its SQL property to build the data query to get the information we want:

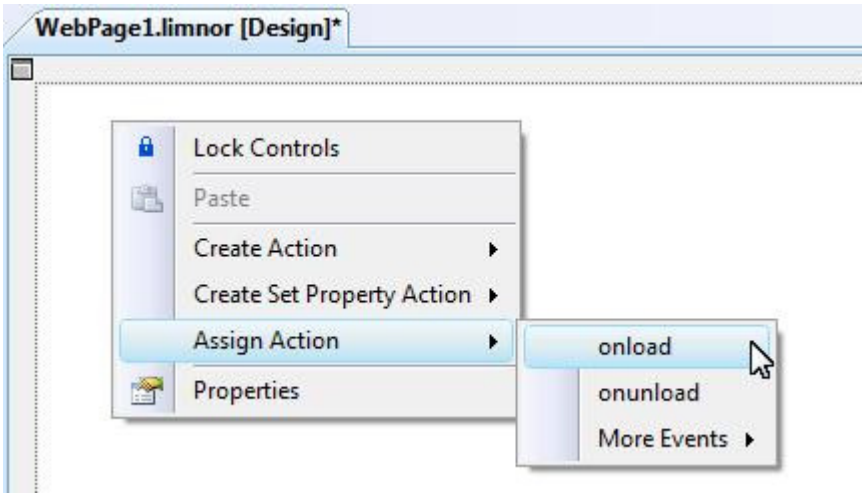


For more information on building data query, see
http://www.limnor.com/support/Limnor%20Studio%20-%20User%20Guide%20-%20Part%20VI.pdf

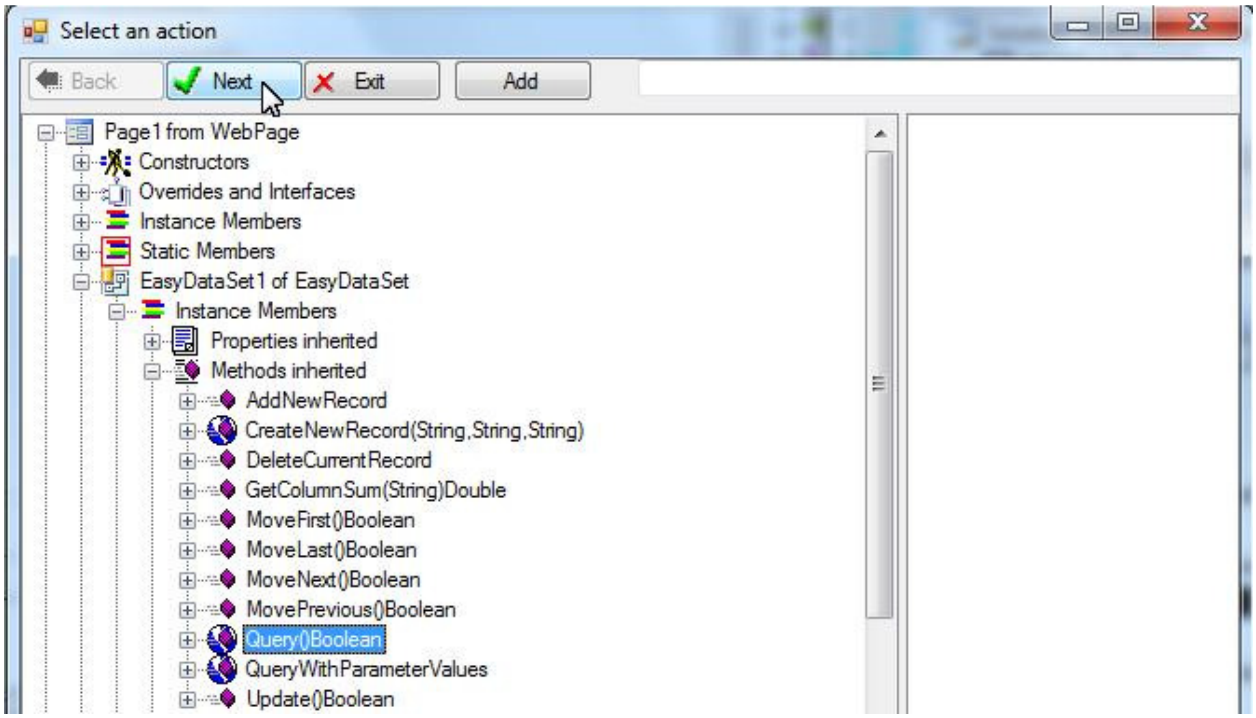
You may set other properties of the EasyDataSet according to your business requirements. For example, set ForReadOnly to True to not allow data editing through the web page; to False to allow data editing:



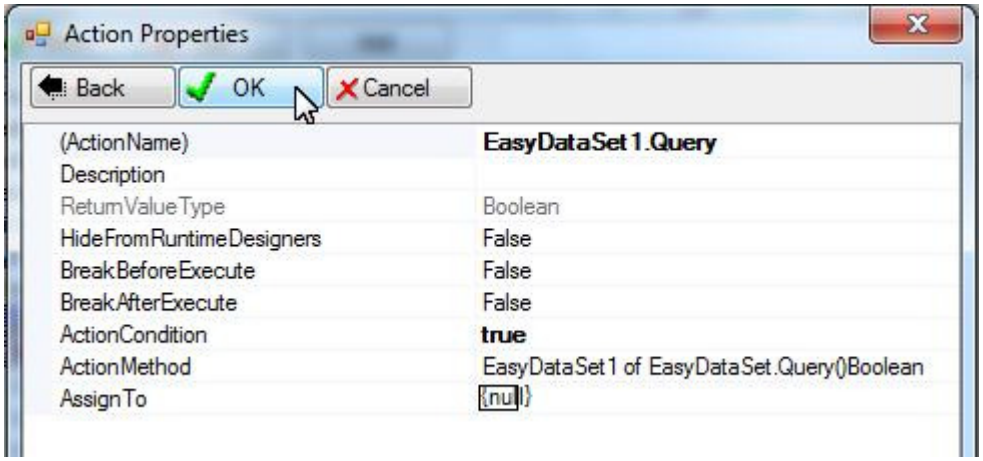
We want to load the data when the web page is loaded. Right-click the web page; choose “Assign Action”; choose “onload” event:



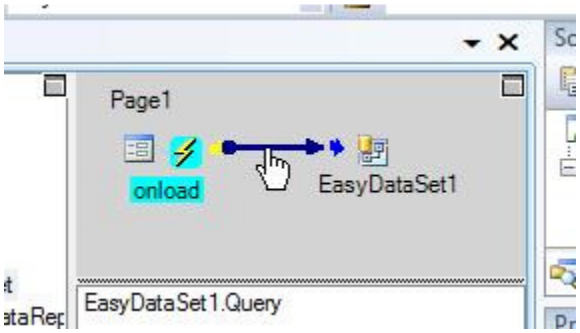
Select the Query method of the EasyDataSet:



Click OK:

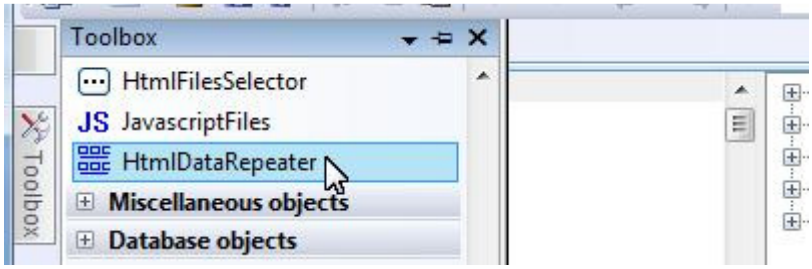


A Query action is created and assigned to the onload event of the web page:

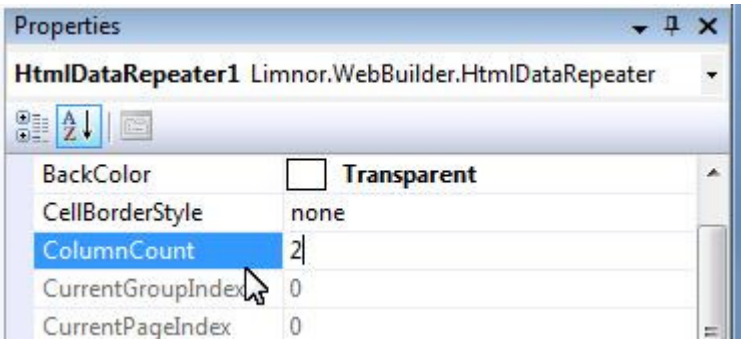


Number of Records in a Page

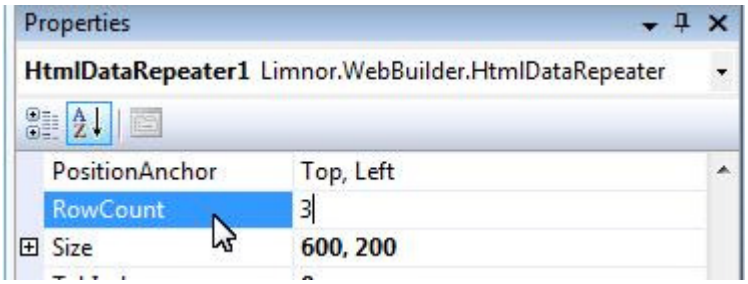
Add a Data Repeater to the web page



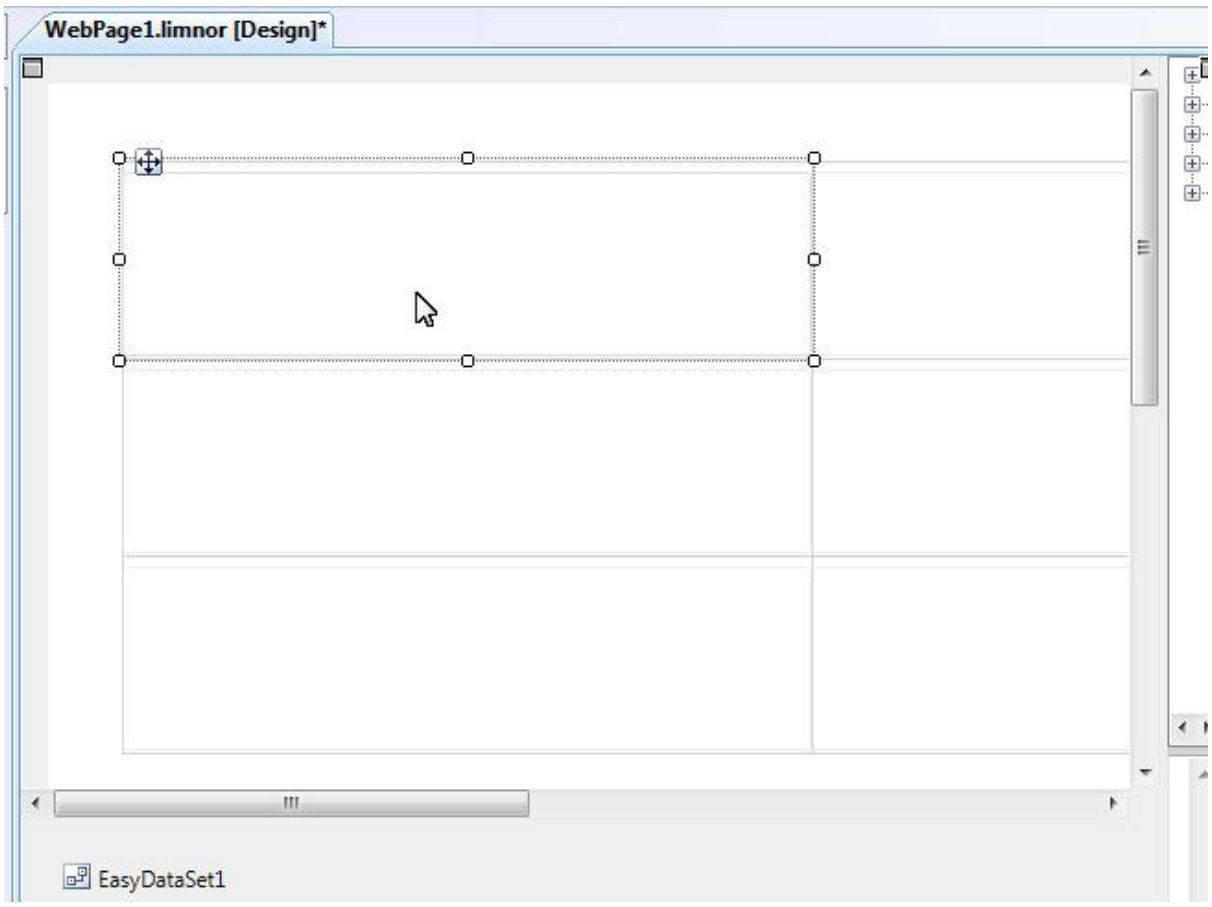
The Data Repeater has a ColumnCount property:



And a RowCount property:

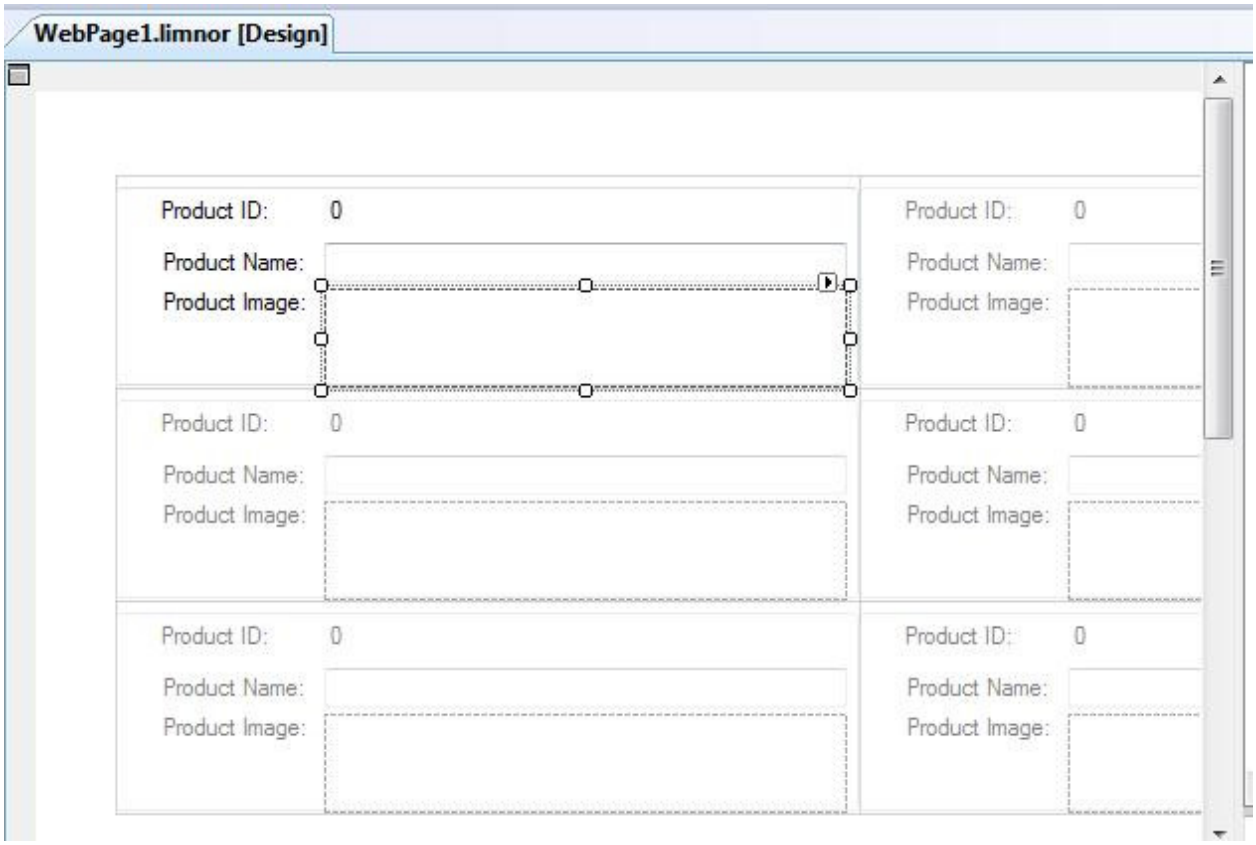


ColumnCount and RowCount determine how many records the data repeater displays:



Form Design

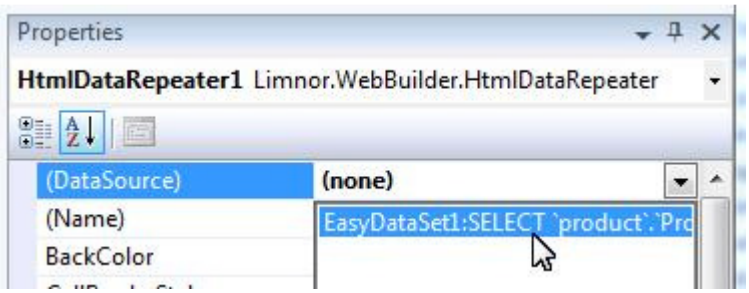
Only the first block of the data repeater can be designed. Drop web client controls to the first block to design the form.



Data Binding

Data-binding to data repeater

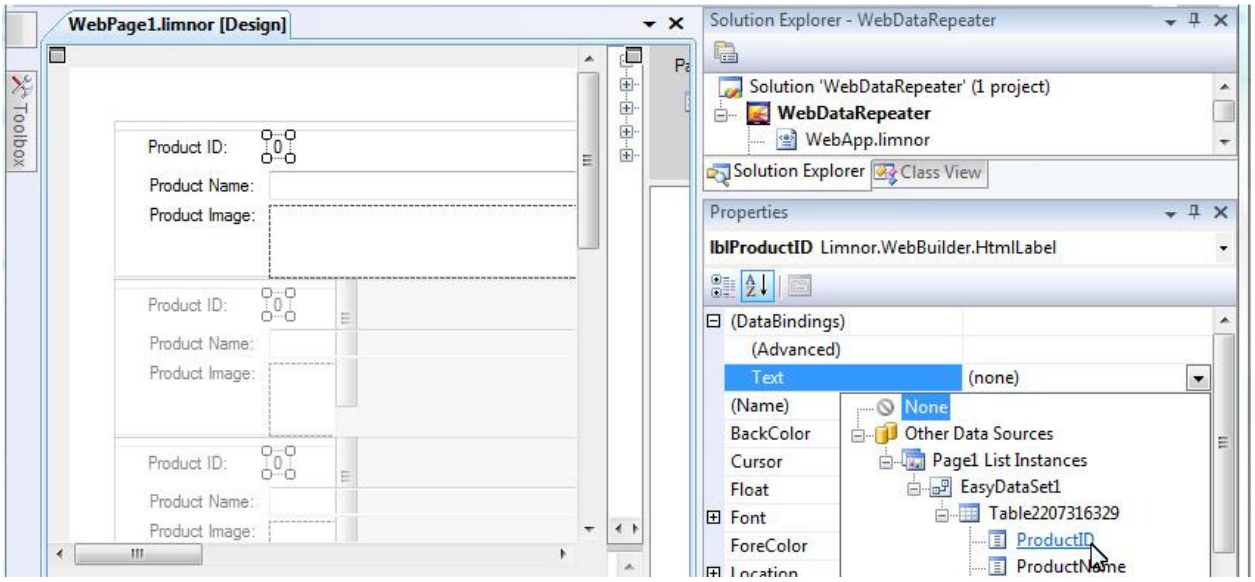
It is very important that the DataSource property of the data repeater is set to the EasyDataSet instance which provides the records to be repeatedly displayed:



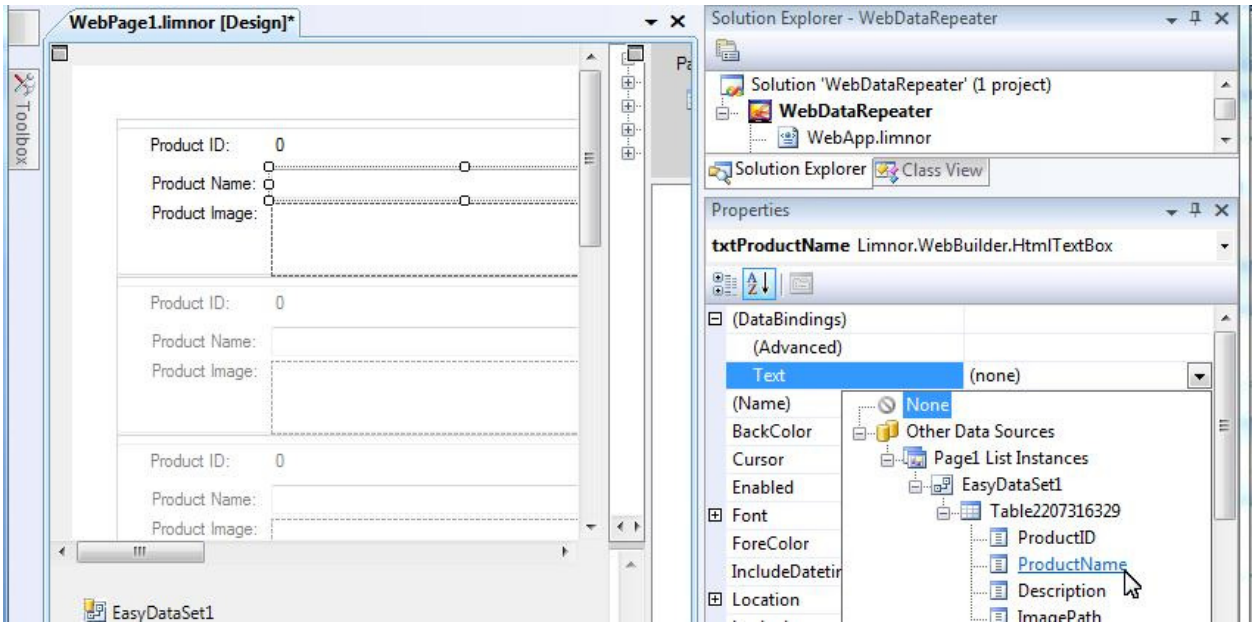
Data-binding to elements

Data-binding to elements are done exactly as it is done without data repeater.

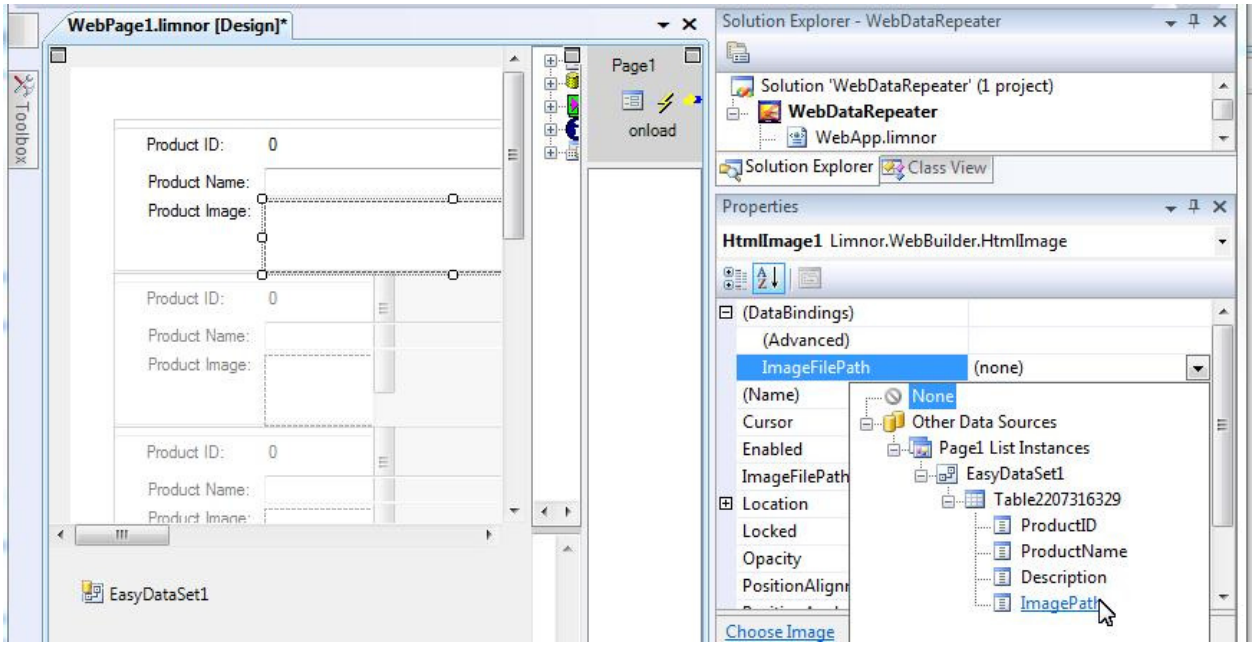
We use a label to show ProductID field because we do not want the user to edit the value of it. Bind its Text property to the ProductID field of the EasyDataSet:



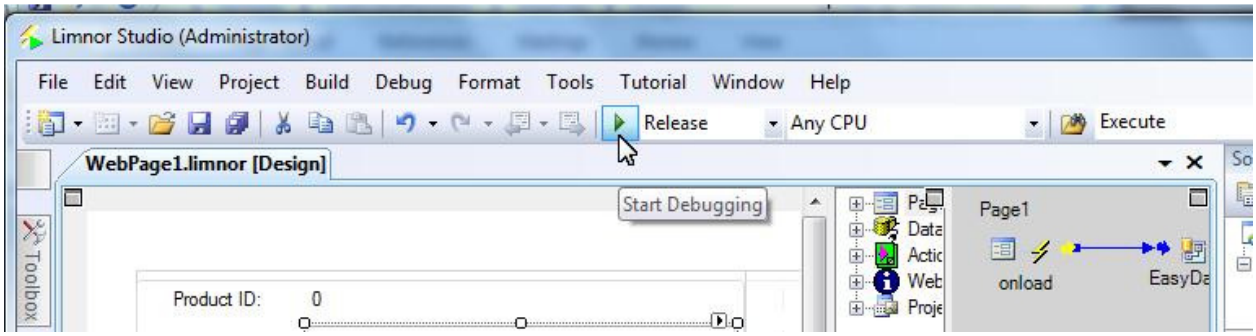
Bind the Text property of the Product Name text box to the ProductName field:



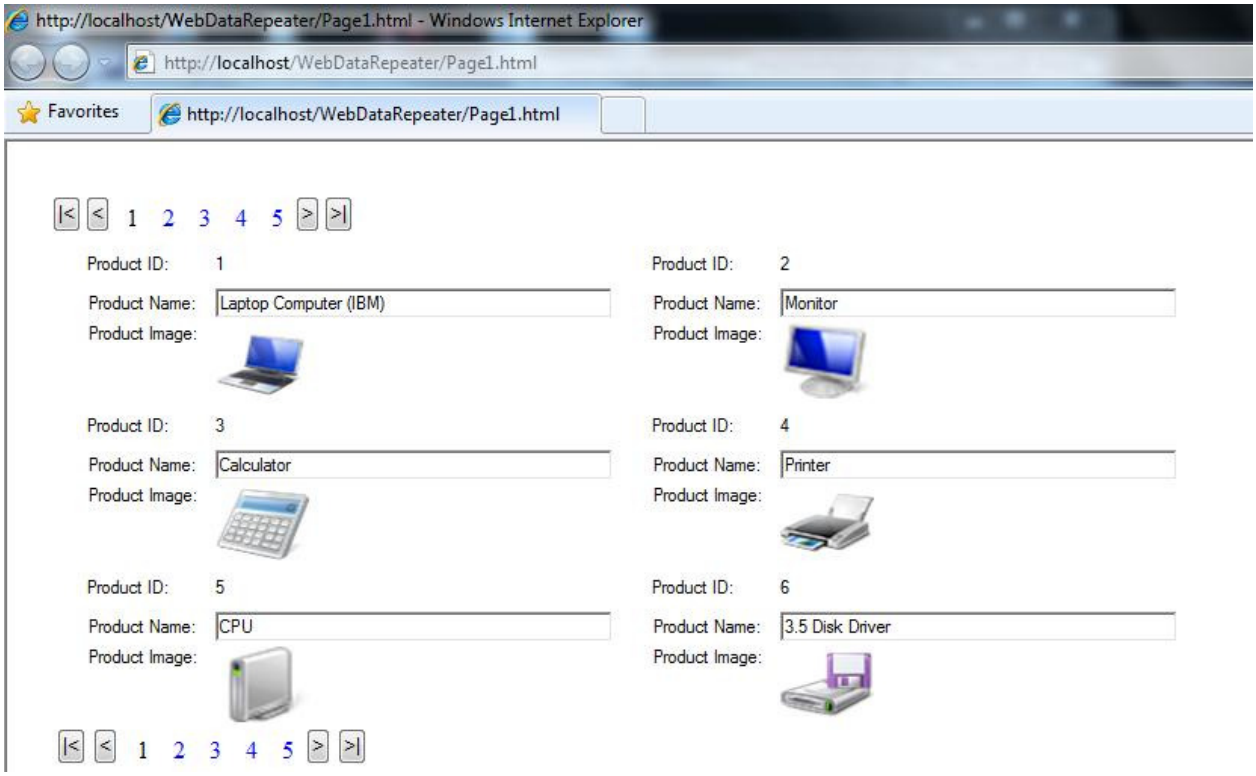
Bind the ImageFilePath property of the image control to the ImagePath field:



We may test this sample now:



The web page appears in the browser. After a while, the data arrive from the web server:



Page Navigation Design

Default page navigators

By default the data repeater provides page navigators at the top and bottom of the data display.

A default page navigator consists of following elements:

|< -- button for going to the first page.

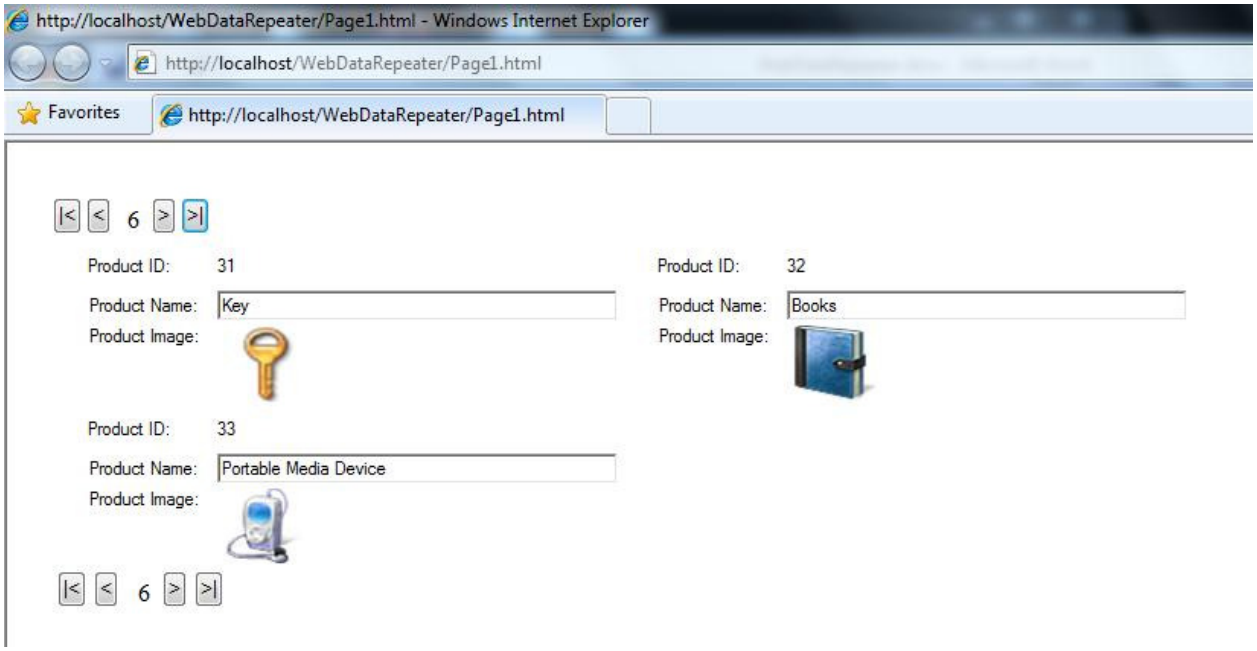
< -- button for going to the previous page.

> -- button for going to the next page.

>| -- button for going to the last page.

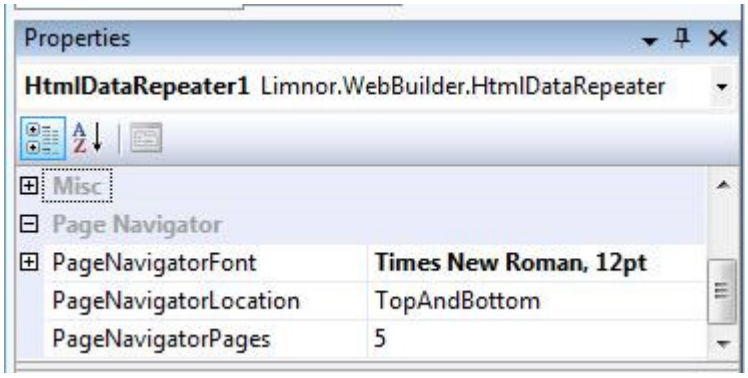
1 2 3 4 5 -- page numbers for going to the specific page.

In this sample, click >|, the last page is displayed:

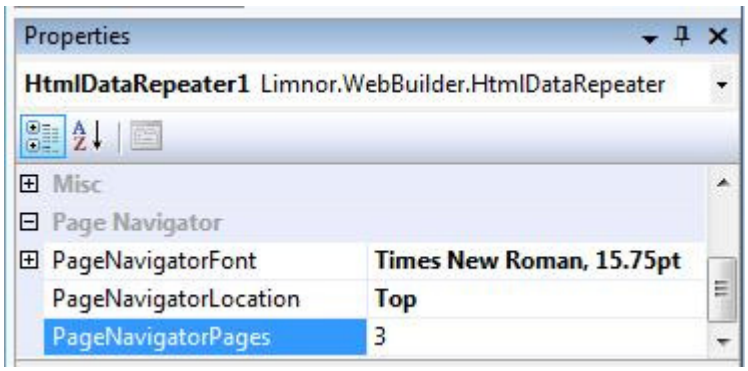


Adjusting default page navigators

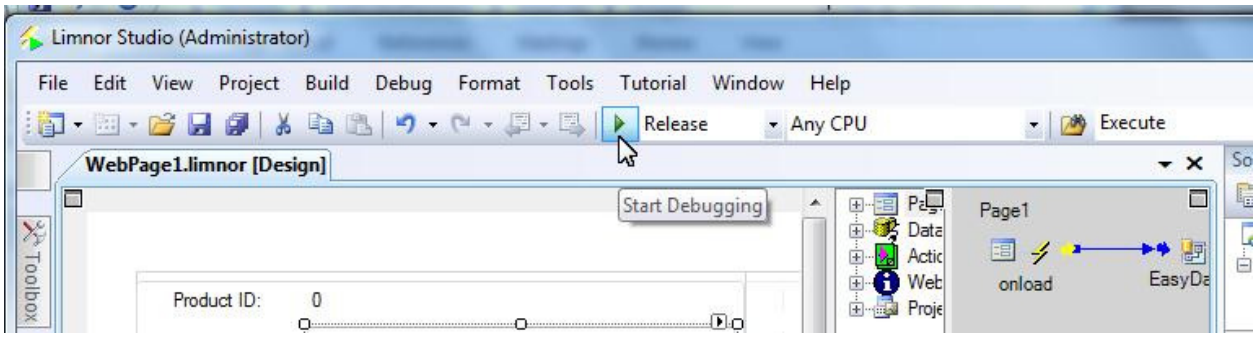
You may adjust the default page navigators through properties PageNavigatorFont, PageNavigatorLocation, and PageNavigatorPages:



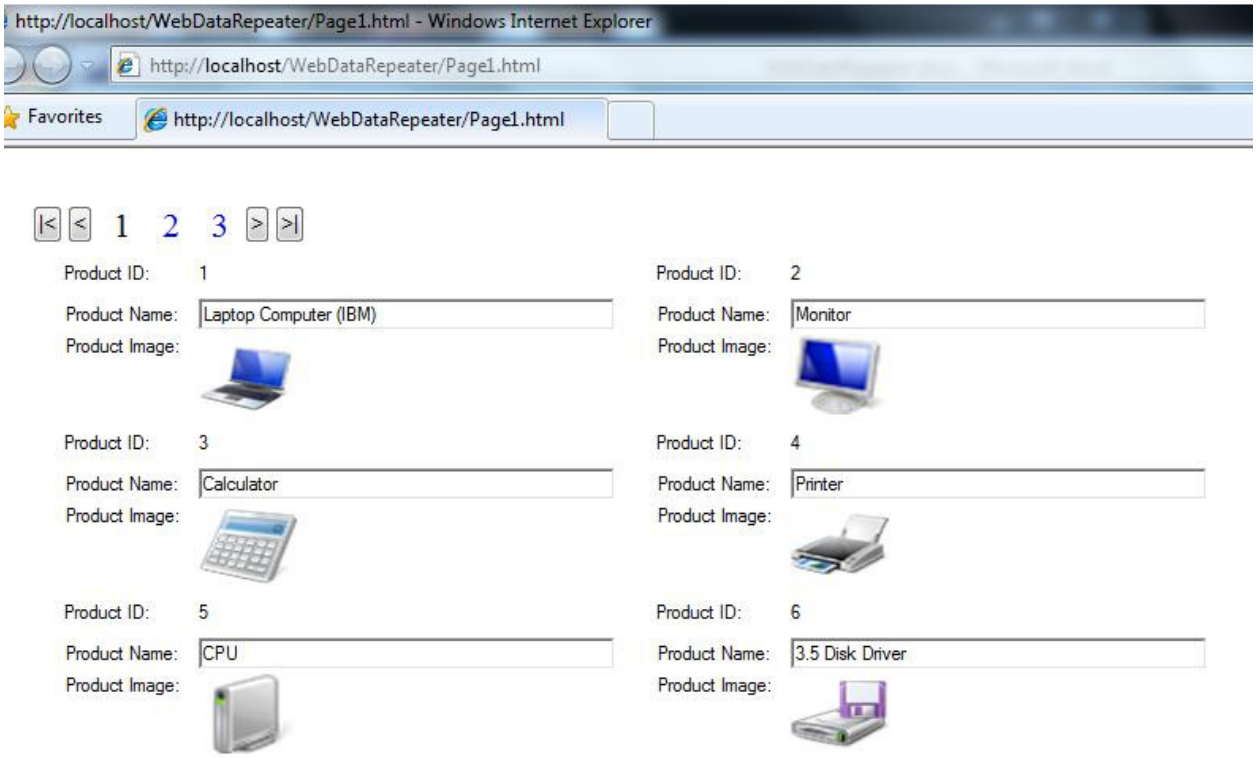
Let's modify these properties to see the effects:

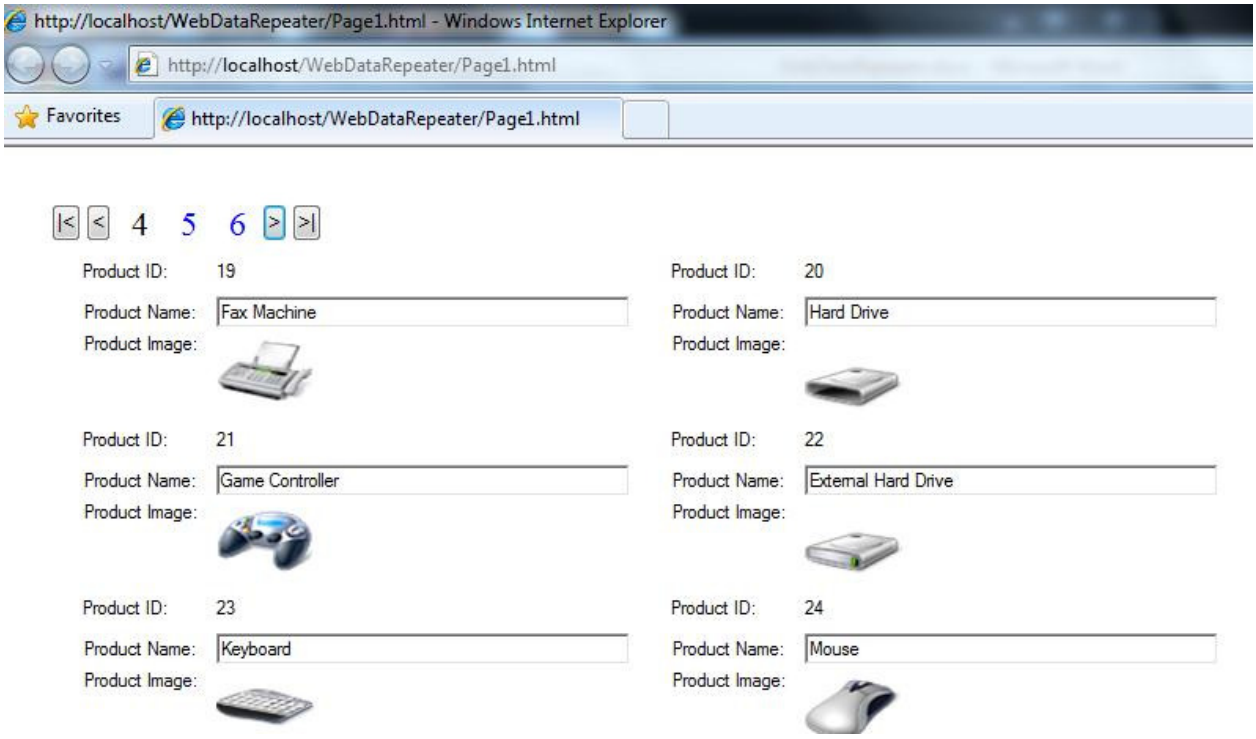


Run the sample project again:



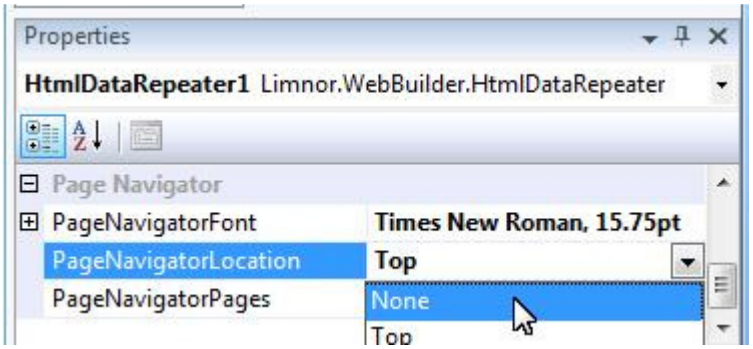
We can see that only 3 page numbers are displayed; the font is bigger; the navigator at the bottom disappears:

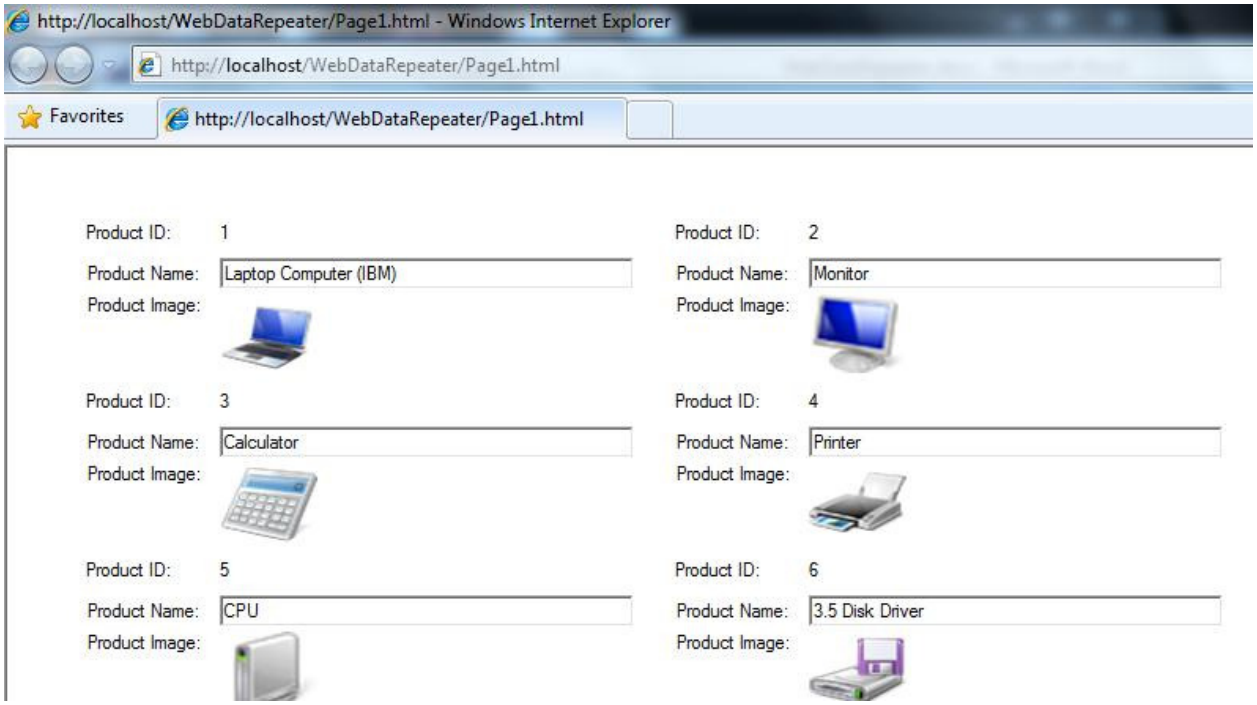




Remove default page navigators

You may completely remove the default page navigators by setting PageNavigatorLocation to None:





Without the default page navigators you may want to add your own page navigators.

Build your own page navigator

Your own page navigators may co-exist with the default page navigators; or you may remove the default page navigators and only display your own page navigators.

The data repeater provides following methods and properties for building a page navigator.

Page navigation methods

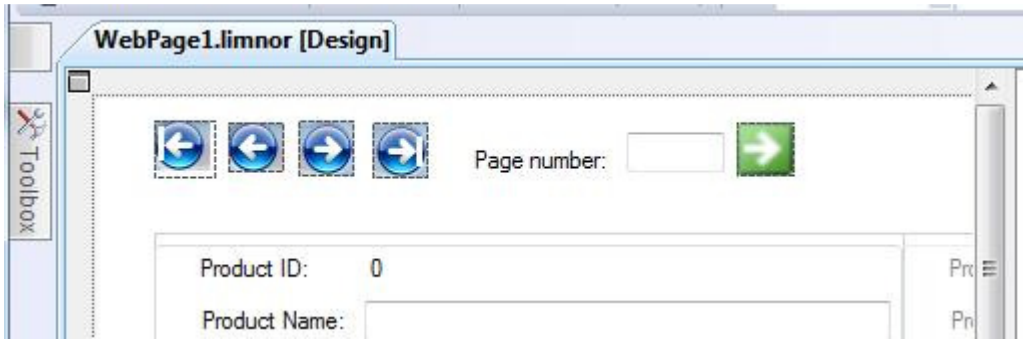
- `MoveToFirstPage`
- `MoveToLastPage`
- `MoveToNextPage`
- `MoveToPreviousPage`
- `MoveToPage(int pageNumber)` – Note that the page number starts from 1, not from 0. The first page is page 1, the second page is page 2, and so on.

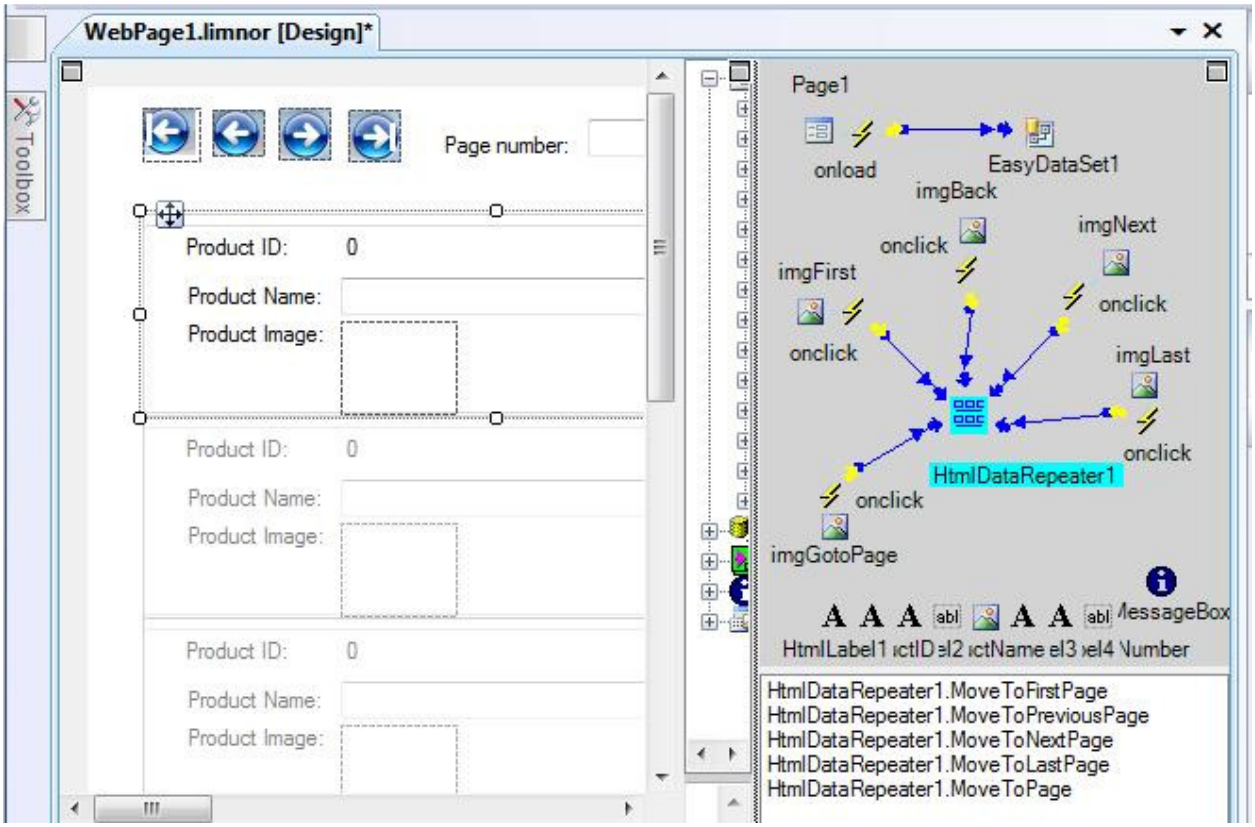
Page navigation properties

- `GroupsPerPage` -- Gets the number of groups. It is the product of the number of columns and the number of rows.
- `TotalPages` -- Gets the number of pages currently available.
- `CurrentPageIndex` -- Gets the page number of the current page. Note that this index starts from 0, not from 1. The first page is page 0, the second page is page 1, and so on.
- `PageNavigatorPages` -- Gets and sets the number of page numbers displayed on the page navigator

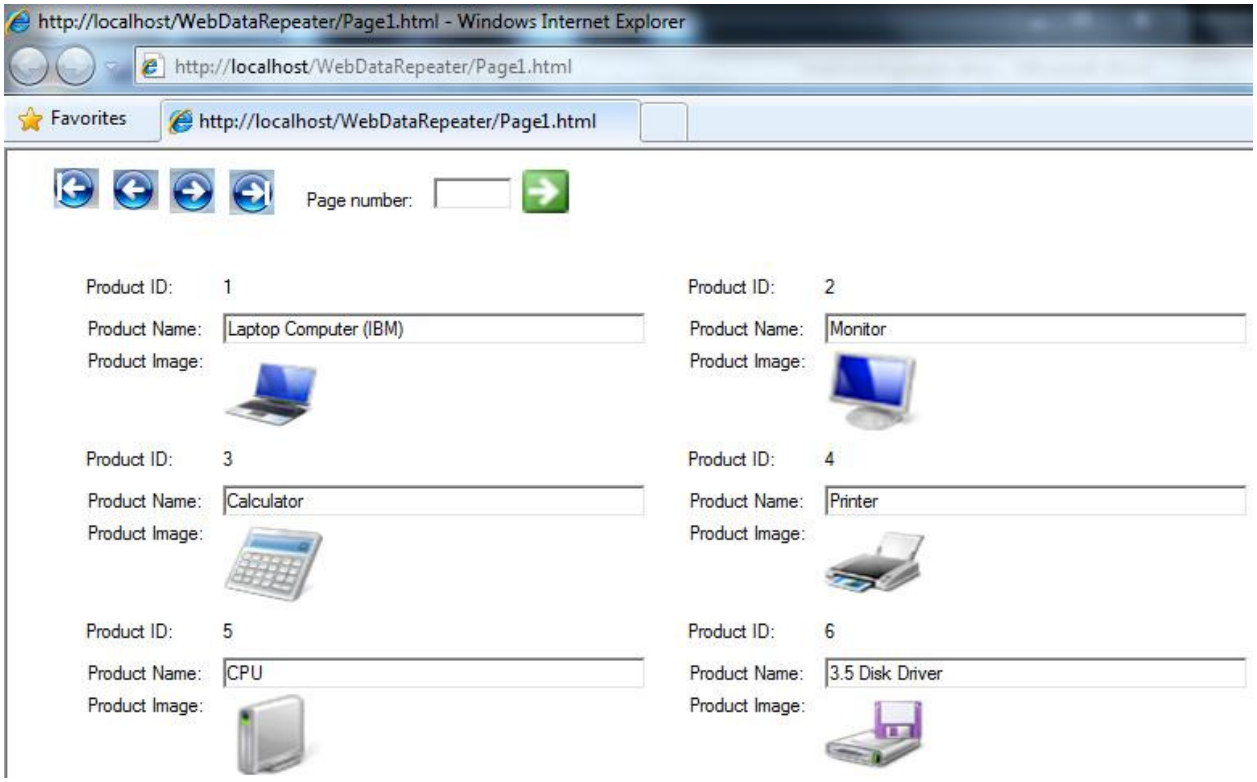
A simple page navigator

We use 4 images to execute MoveToFirstPage, MoveToLastPage, MoveToNextPage, and MoveToPreviousPage respectively; use a text box to allow the user to enter page number and use an image to execute MoveToPage with the page number.

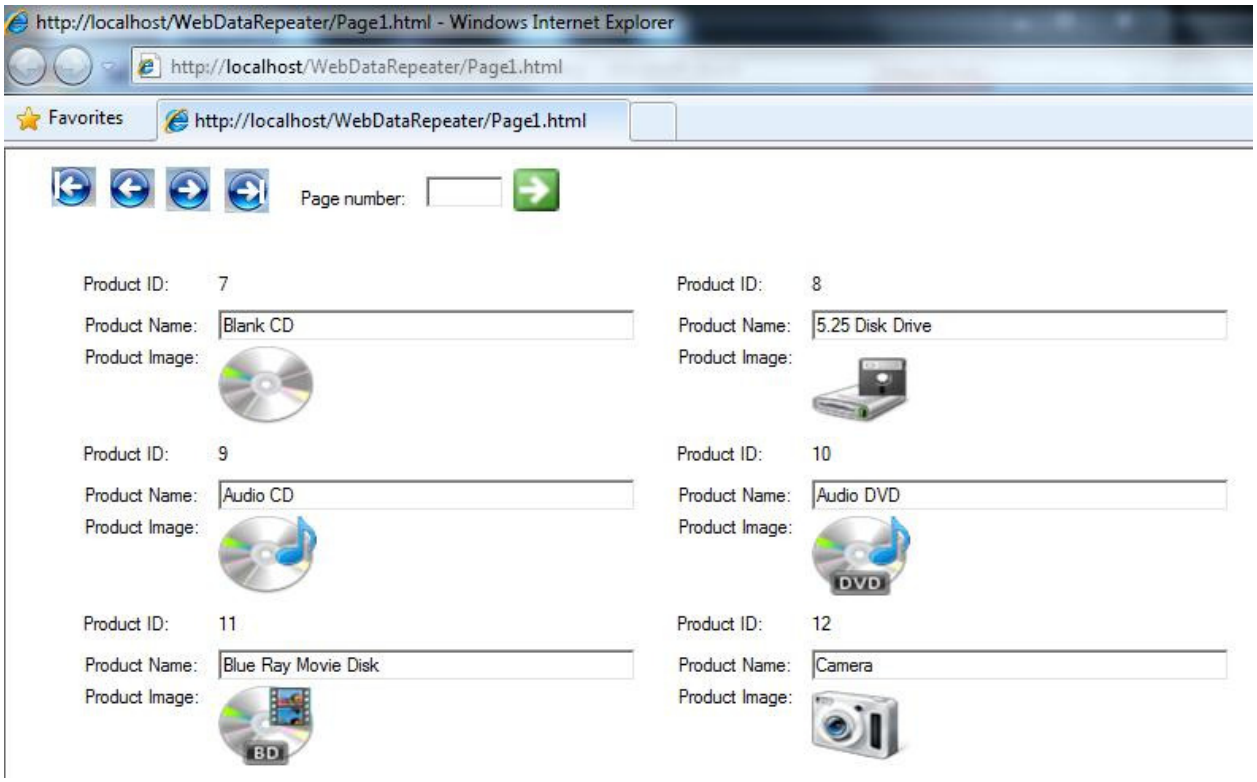




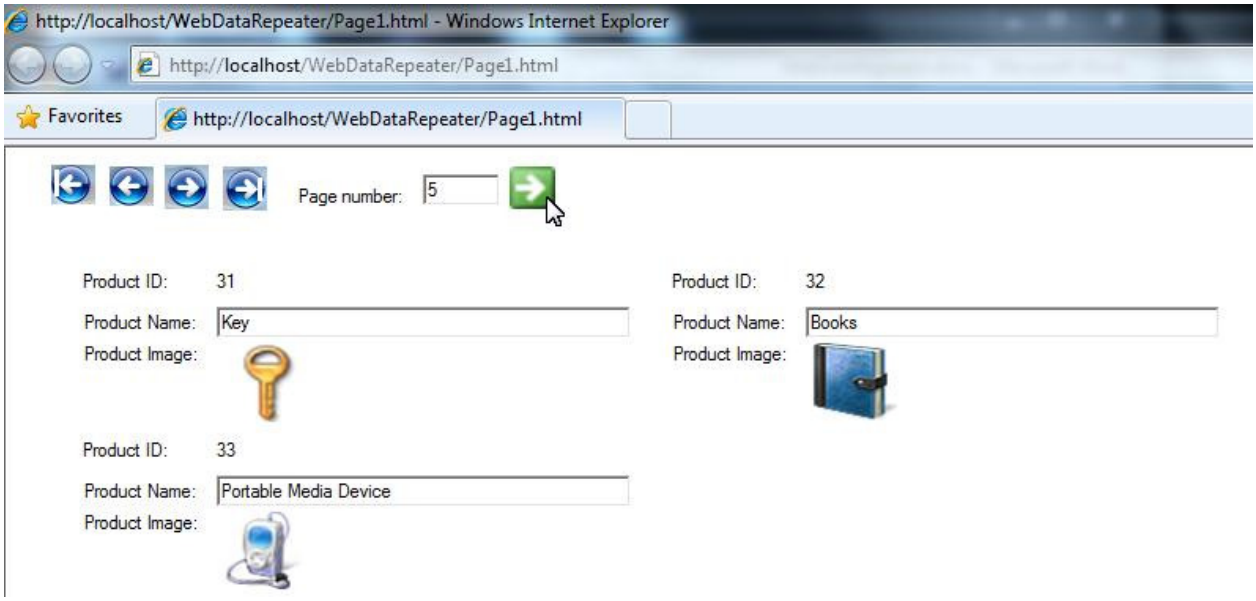
Run the sample, the first page appears:



Click  , and the next page appears:



Enter a page number and click , and the page appears:

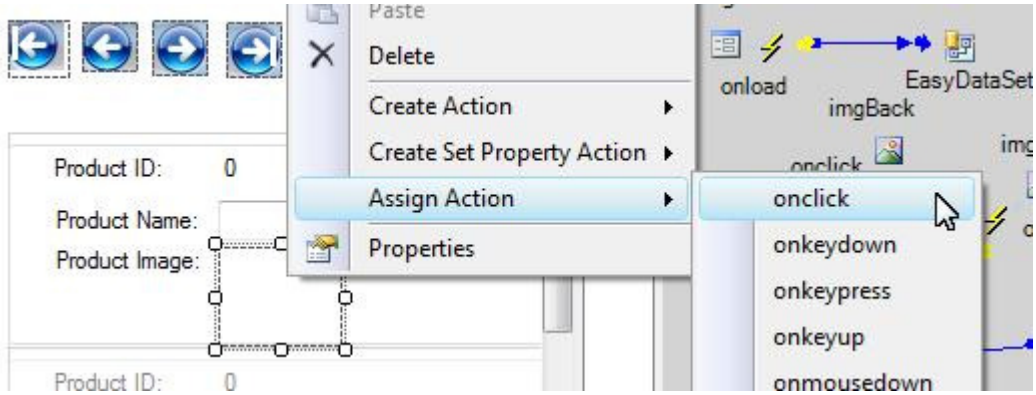


Programming inside data repeater

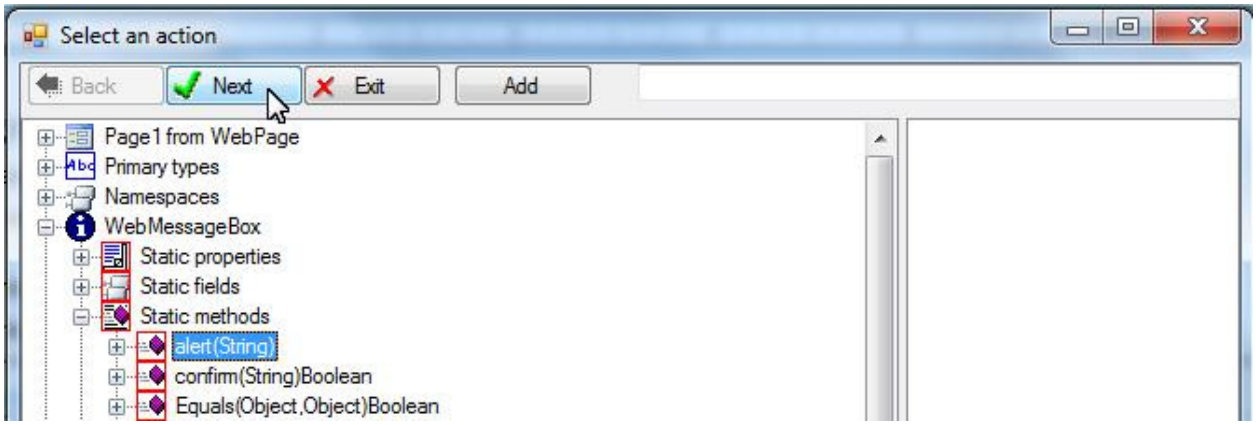
You may create actions and assign actions to events on those elements inside the data repeater. The programming is done on the first block and it automatically applies to all blocks.

Let's show an example. Suppose when the image is clicked we want to show a message box showing the contents of the product name text box.

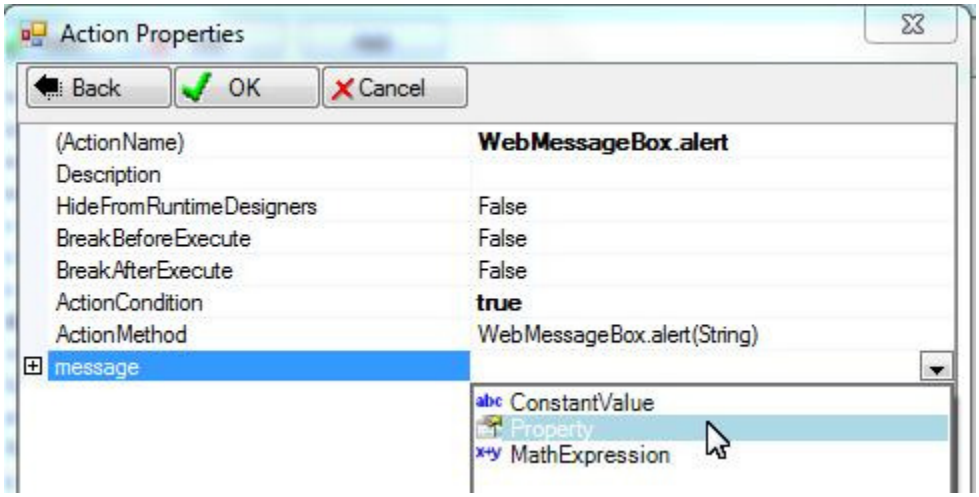
Right-click the image; choose "Assign Action"; choose "onclick" event:



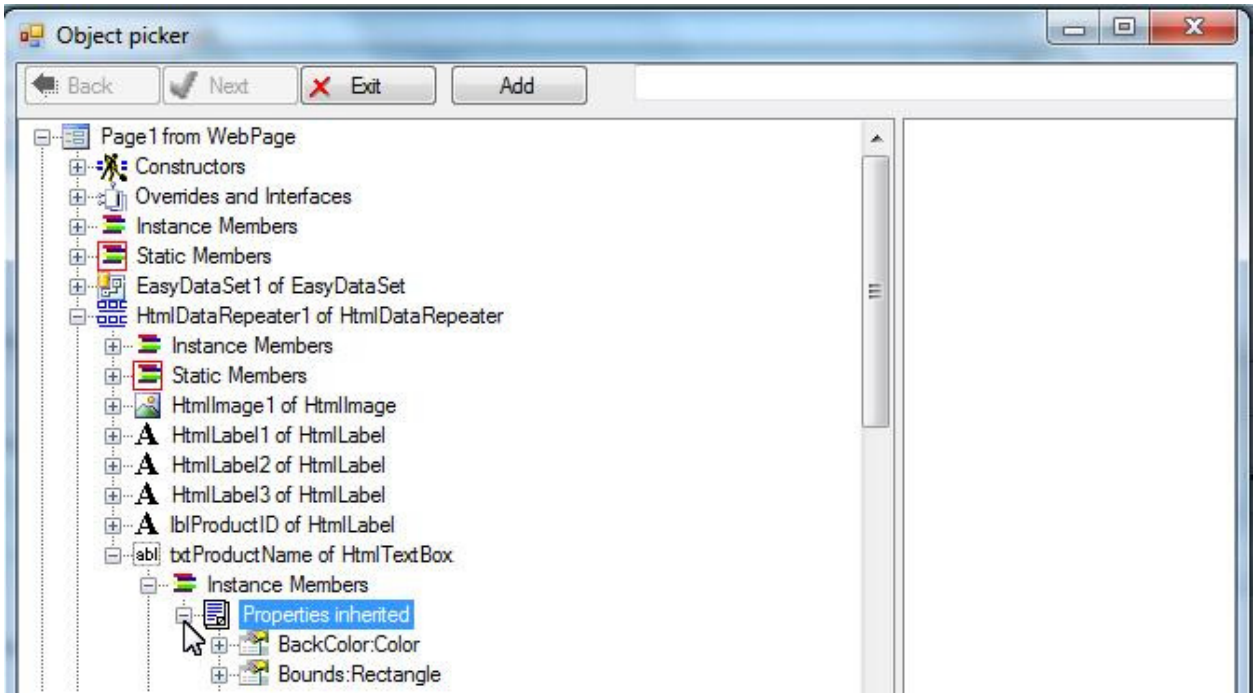
Select the alert method of the message box:

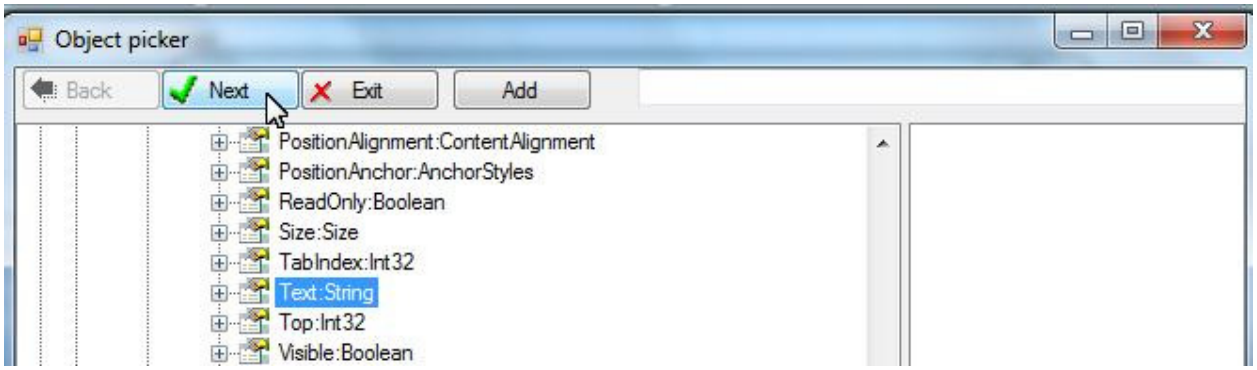


For the “message”, select Property:

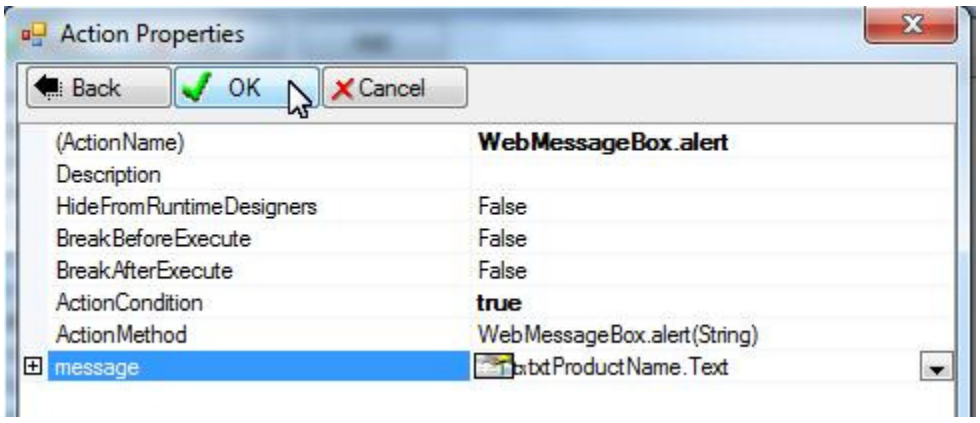


Select the Text property of the product name text box:

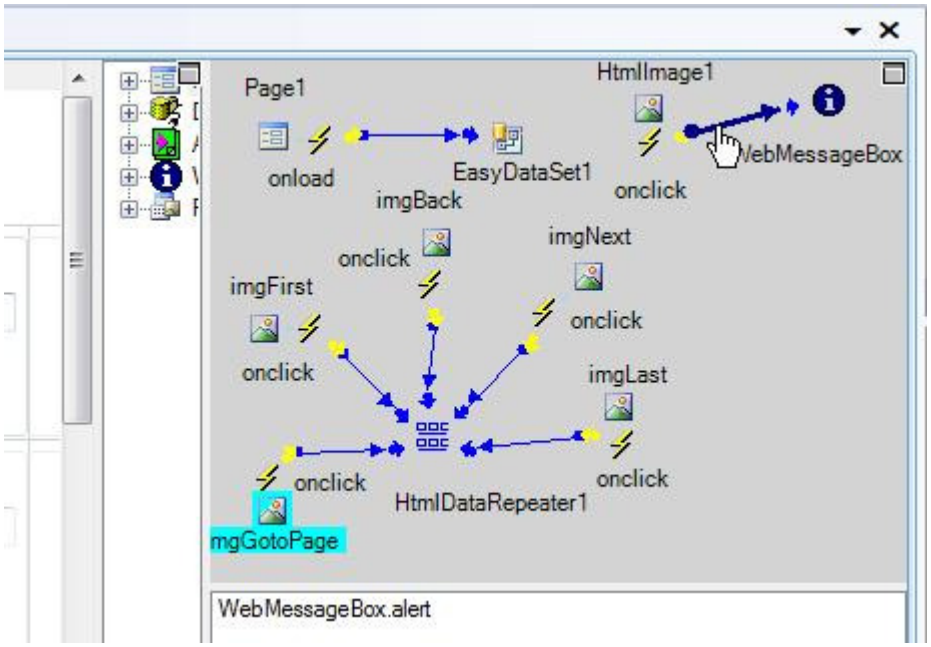




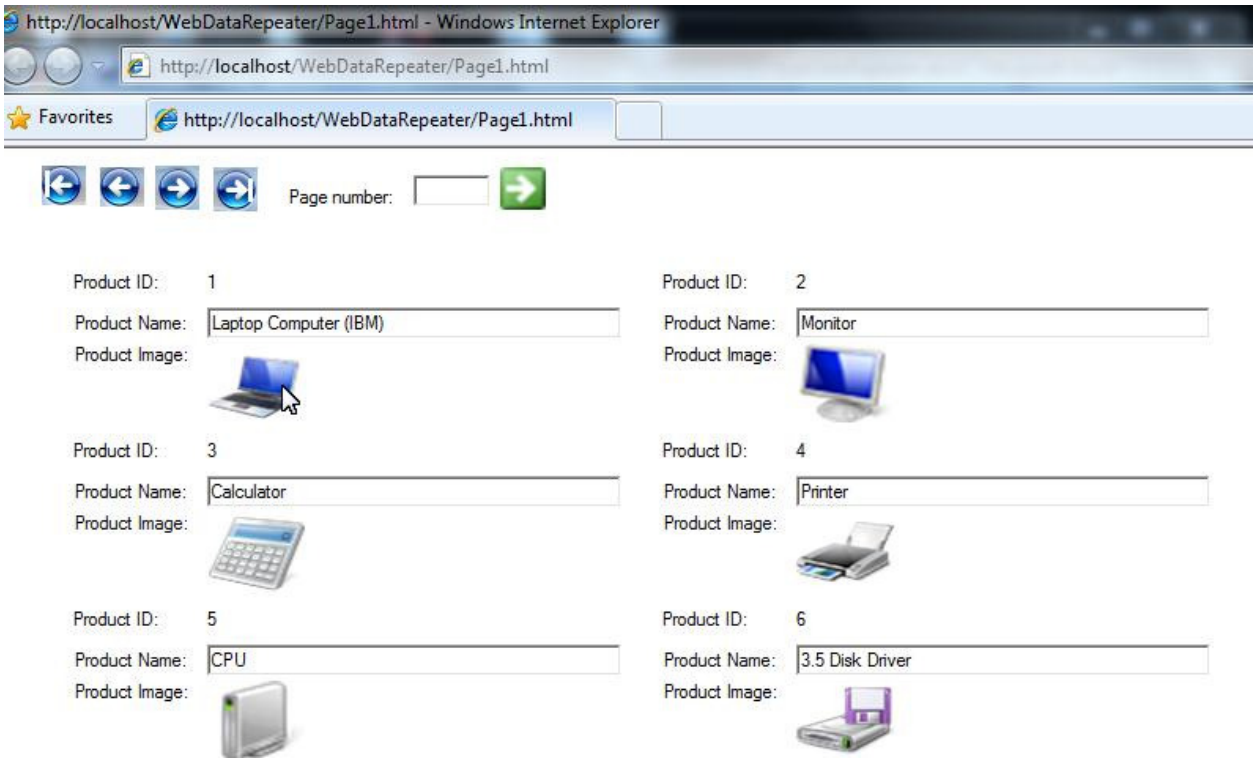
Click OK:



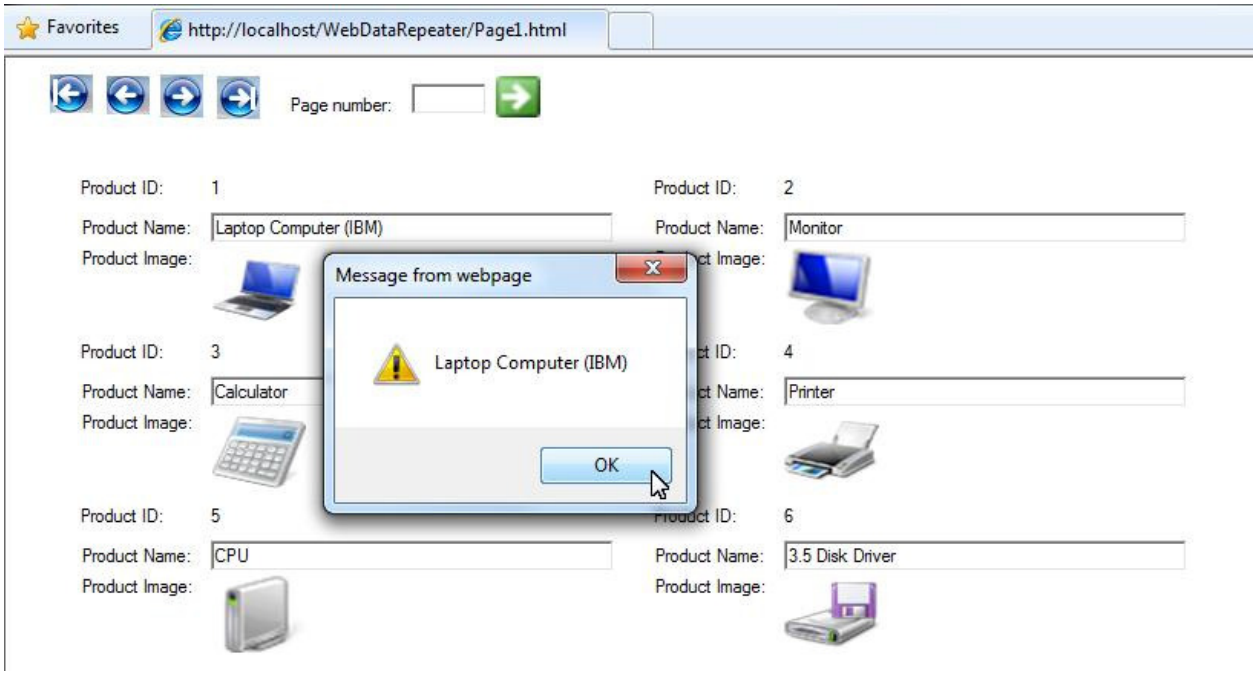
An action is created and assigned to the image:



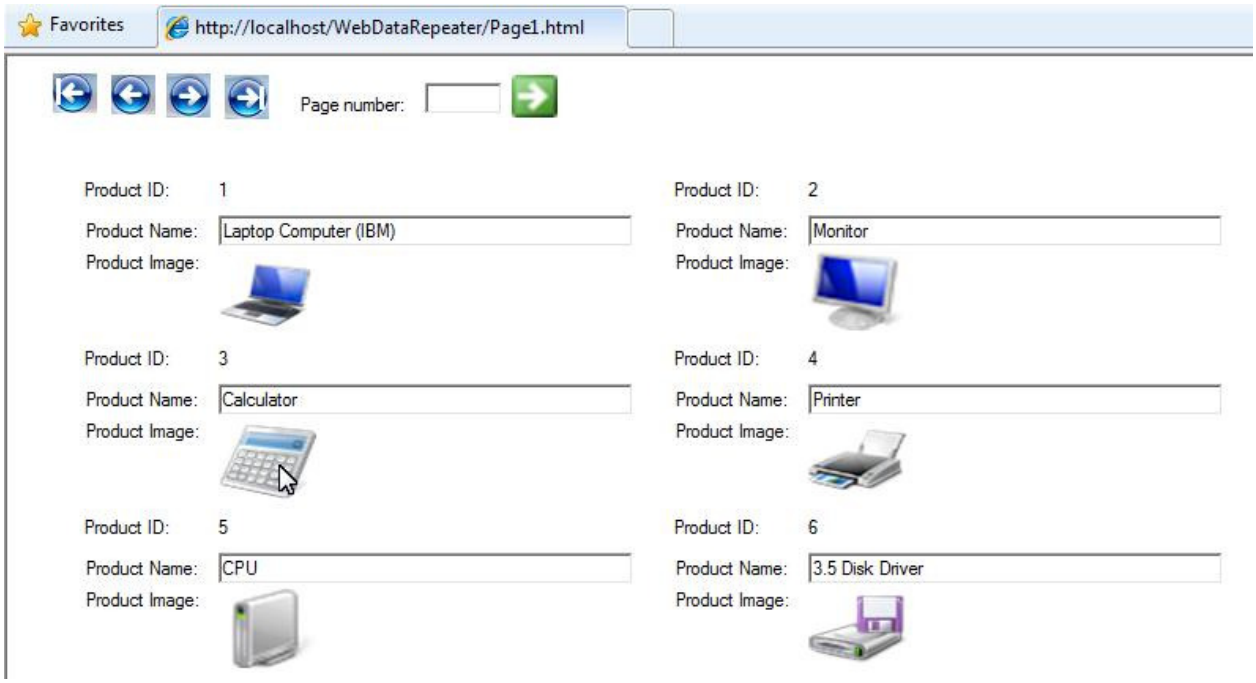
Run the sample. The first page appears. Click the image on the first block:



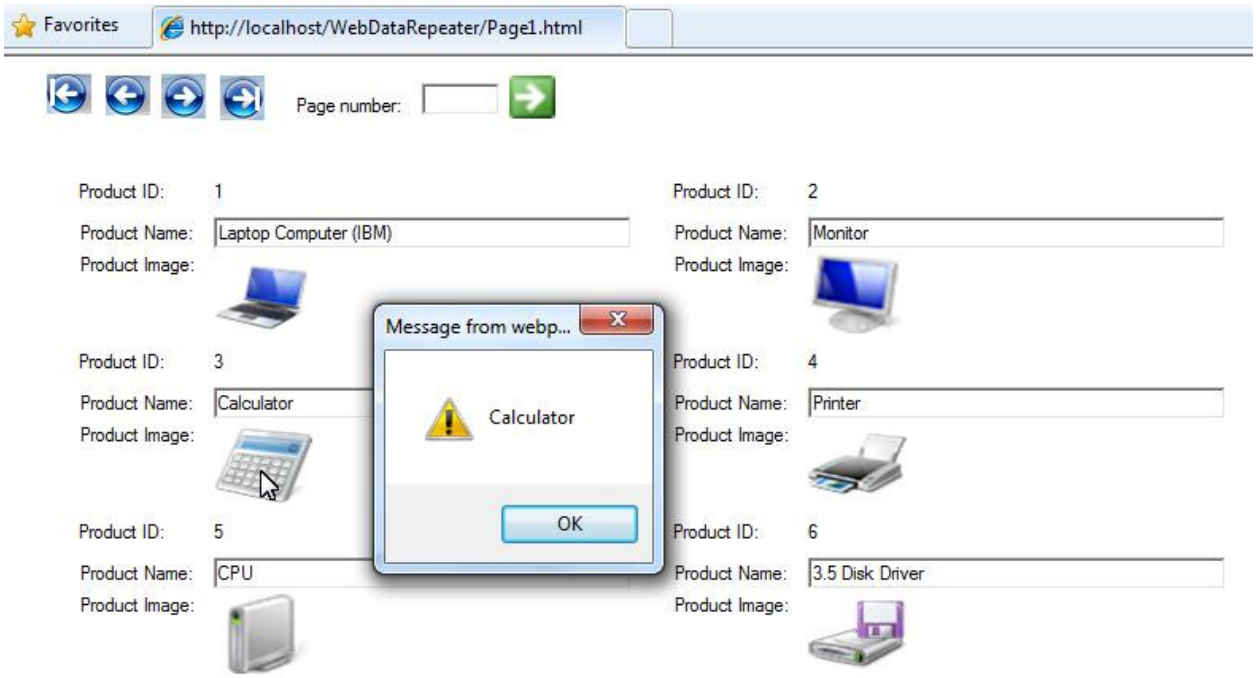
A message box appears showing the contents of the first product name text box:



Click the image of the 3rd block:



A message box appears showing the contents of the product name text box of the 3rd block:



Move to another page and click another image, the corresponding text box contents are displayed: