Ajax DataGrid Manual

About Ajax DataGrid

Create a great looking, interactive Ajax Grid in Dreamweaver within minutes. Choose from predefined designs, add grid paging and inline search. Select one of the many table formatting options to fit the Grid perfectly in your site.

The grid loads extremely fast due to Ajax technology that only loads the displayed data. Connect with any datasource, whether it's MySQL or MS SQL Server, using auto generated ASP or PHP Code. Add interactive components to your DataGrid such as **3D ImageFlow Gallery**, **Advanced Tooltips**, **Advanced Layer Popup** or **Flash MP3 Player**.

Users can customize the grid display with a rich set of options. Columns are sortable, can be reordered and resized. They can even resize the grid by dragging its borders. Presenting your data has never been so intuitive and easy!

Features in Detail

Presenting your data has never been so intuitive and easy!

- **Create a cool Ajax DataGrid in seconds**
  Create a great looking, interactive Ajax Grid in Dreamweaver without coding.

- **Customize the grid any way you want to**
  o Give your DataGrid a sleek and stylish look with one of the predefined styles.
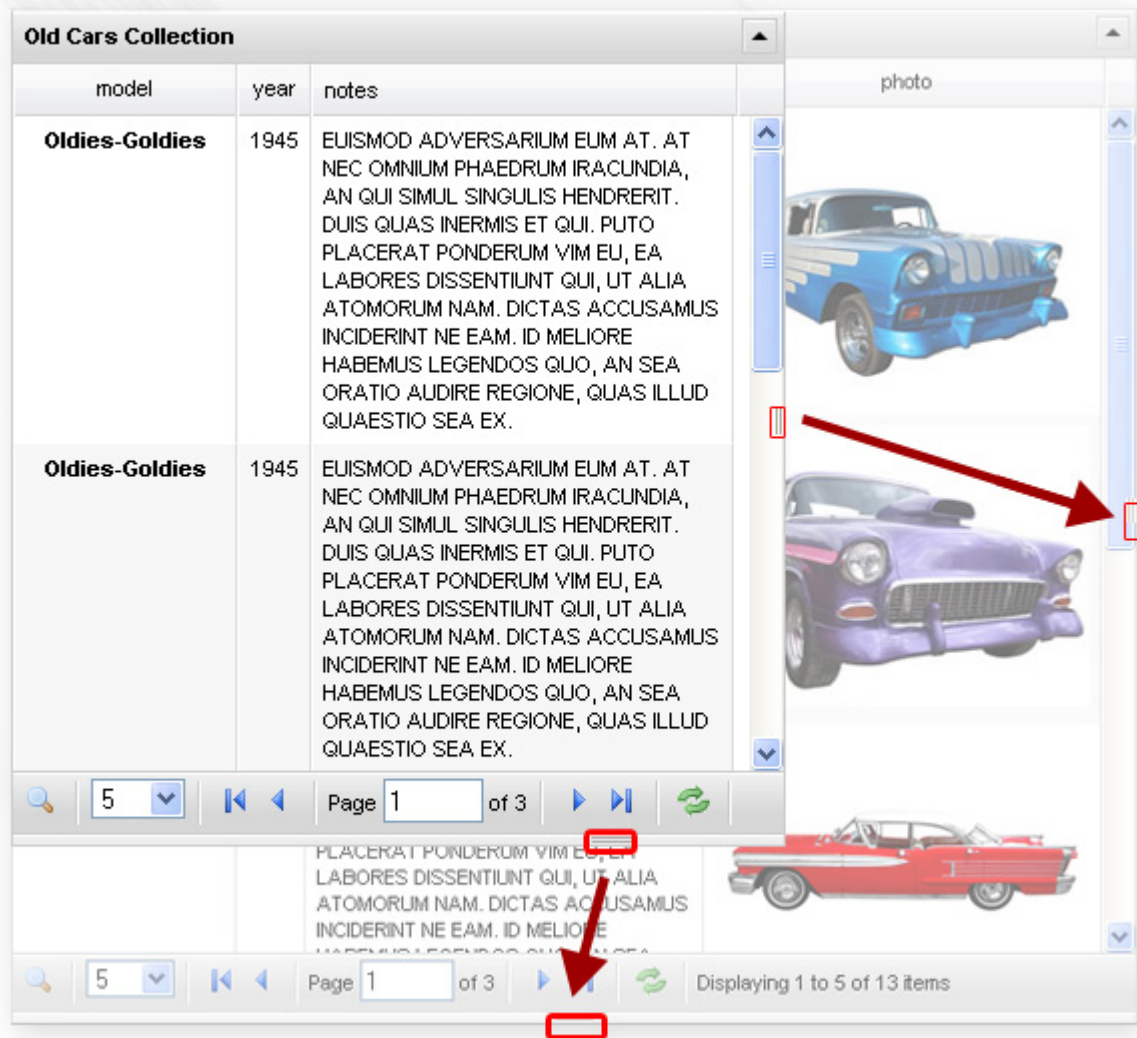 o Columns are sortable, can be reordered and resized.

o   Resize the grid by dragging its borders.

o   Select one of the many table formatting options to fit the Grid perfectly in your site.
o   Choose from a rich set of navigation options that include paging, searching and filtering options.
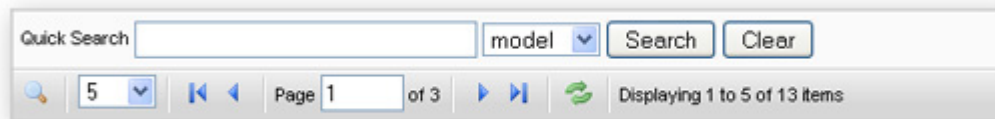
o   You can customize the content/images inside your table with formatting, alignment, width, and more.
o   Choose to highlight rows or columns, wrap cells, and set opacity while loading.
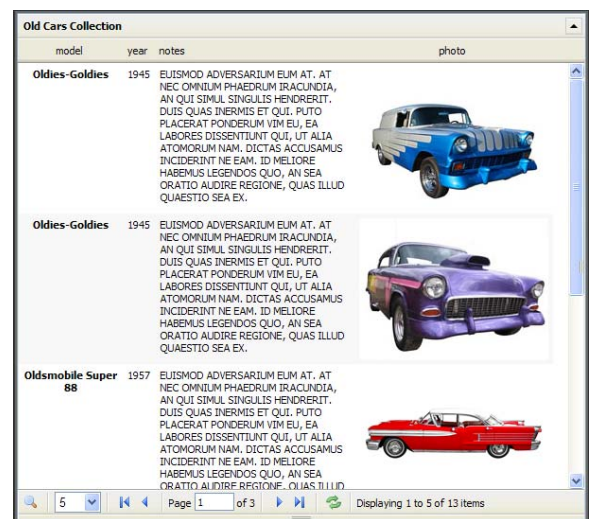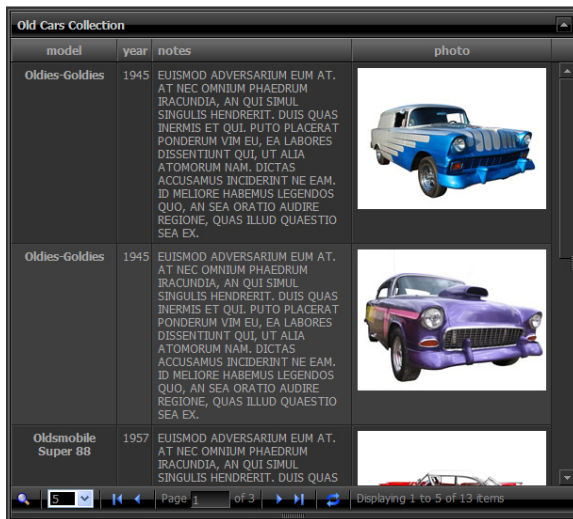o   Enable the users to collapse the grid.

o You can translate your menus and tooltips to any language with the Translation tab. Here are some examples:
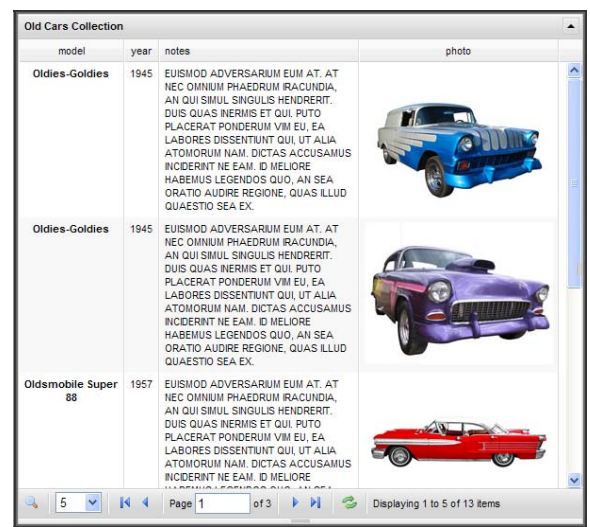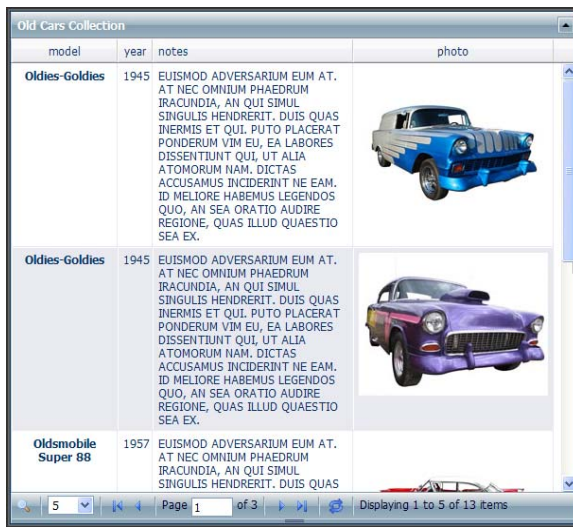Chinese:

Hebrew(supports right to left scripts):

o Great new Skins to chose from for your grid:

- **Ajax Technology**
  The grid loads extremely fast due to Ajax technology that only loads the displayed data.

- **Connectivity**
  Connect with any datasource, whether it's MySQL or MS SQL Server, using auto generated ASP or PHP Code. Use an intuitive interface to browse trough your database and select the table and columns you would like to display.
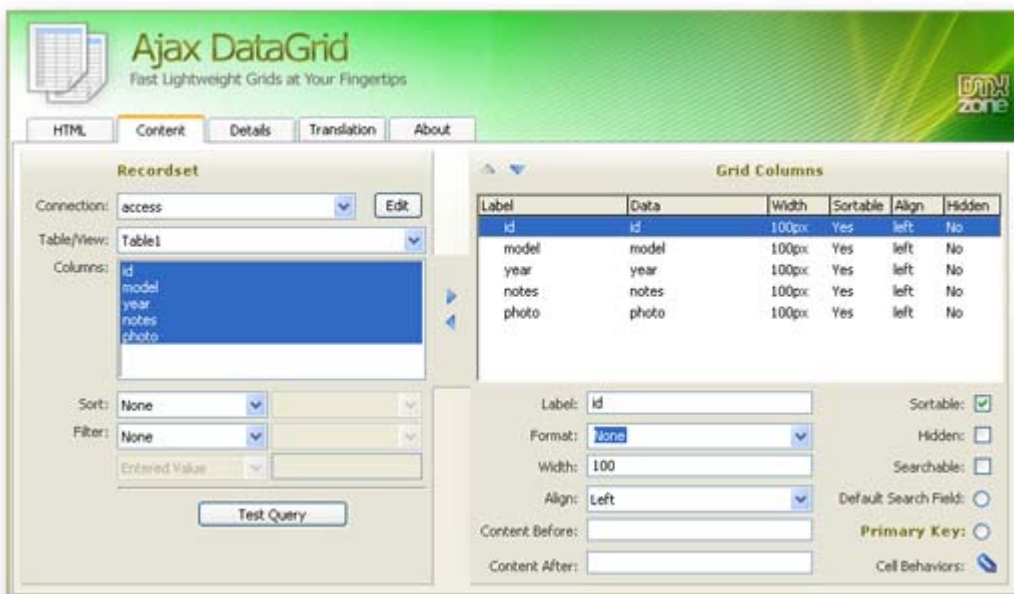
- **Accessibility**
  o   Inline search enables users to quickly find any content they are looking for
  o   Search and sort content
  o   Use Behaviors to control your table from your page.

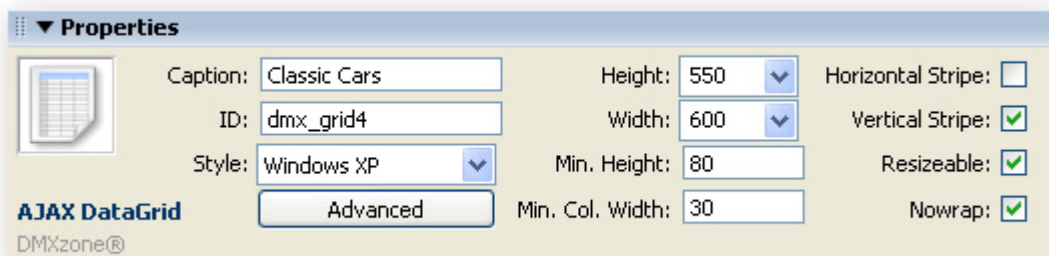- **Fully integrated in Dreamweaver**
  - o Configure the grid with a great looking dialog that visually displays all options. No coding is required.
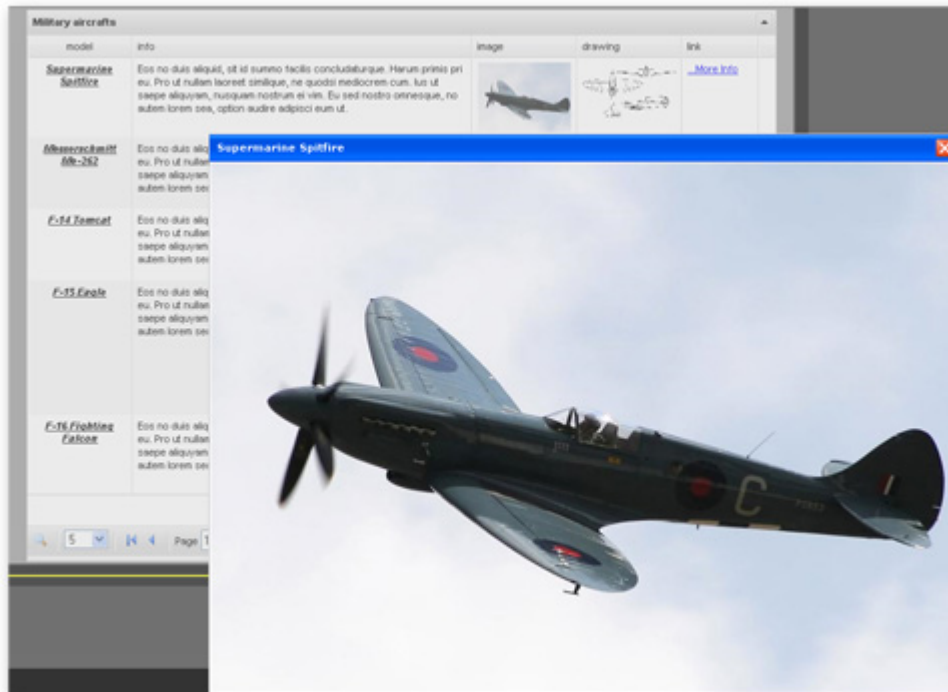

  - o Fully PC and MAC (OSX and Leopard) compatible.

- **Accessible property inspector for easy editing**
  The property inspector enables you to change several options directly.

- **Interoperability**
  - o Add interactive components and behaviors to your DataGrid such as **3D ImageFlow Gallery**, **Advanced Tooltips**, **Advanced Layer Popup** or **Flash MP3 Player**.

  - o Control the grid with your own customized interface, using behaviors.

Before you begin

Make sure to setup your site in Dreamweaver first before creating a page, if you don't know how to do that please read this.

Installing the extension

Read the tutorial at http://www.dmxzone.com/ShowDetail.asp?NewsId=4671

Login, download the product, and lookup your serial at
http://www.dmxzone.com/myZone.asp?Section=purchases

Get the latest extension manager from Adobe at
http://www.adobe.com/exchange/em_download/

Open the .mxp file (the extension manager should start-up) and install the extension.

If you experience problems please read the FAQ at
http://www.dmxzone.com/index.asp?TypeId=7&CatId=687 before contacting support.

The Basics: Creating a Basic Ajax DataGrid

Introduction

In this tutorial we're going to show you how easy it is to insert an Ajax DataGrid into your page. Below is an example of the end result.

The Ajax DataGrid on your page. All the settings in the first tab are also explained in the preview window of the extension.
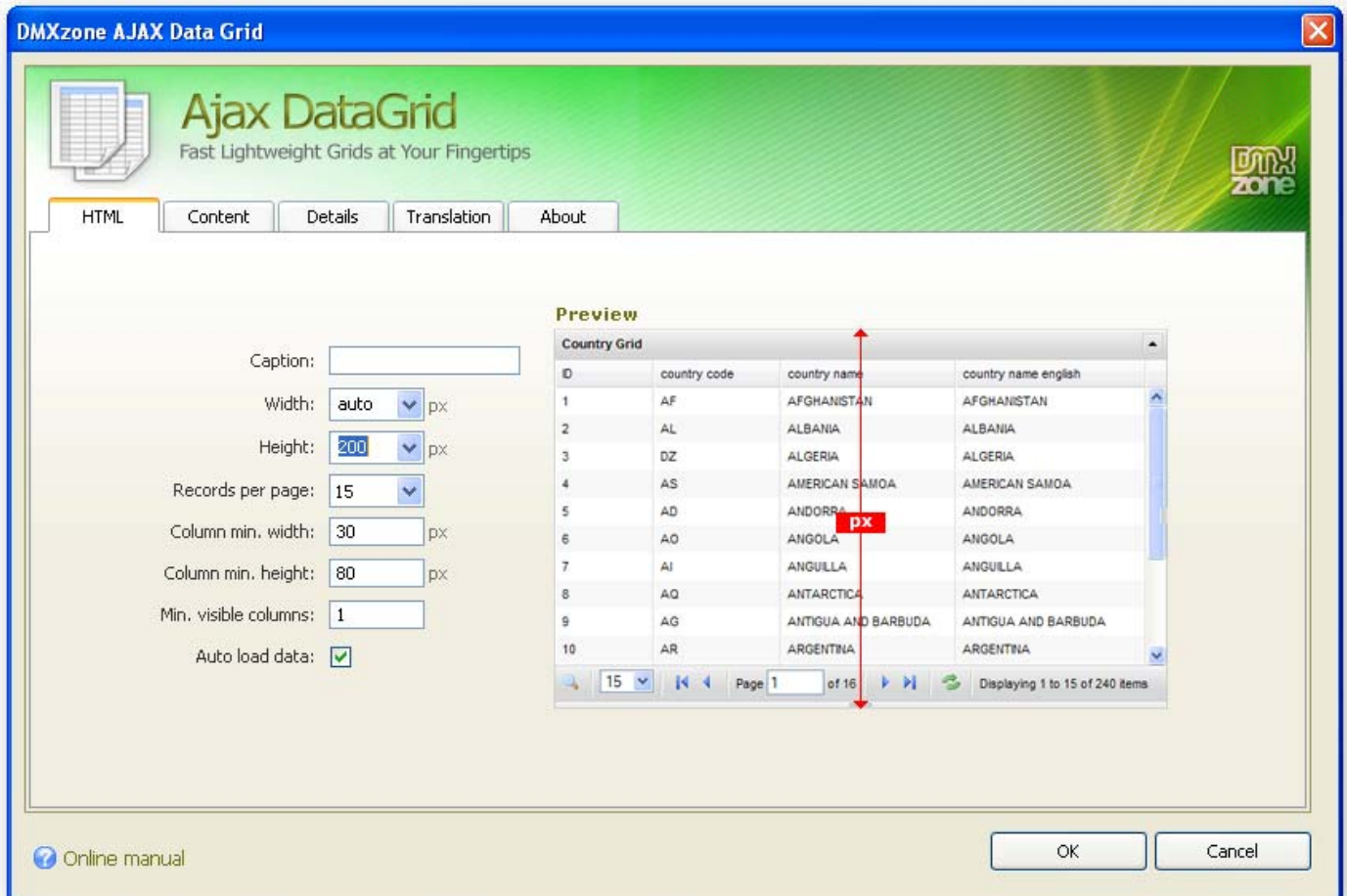
Make sure you have defined a connection to your database before you start this tutorial.
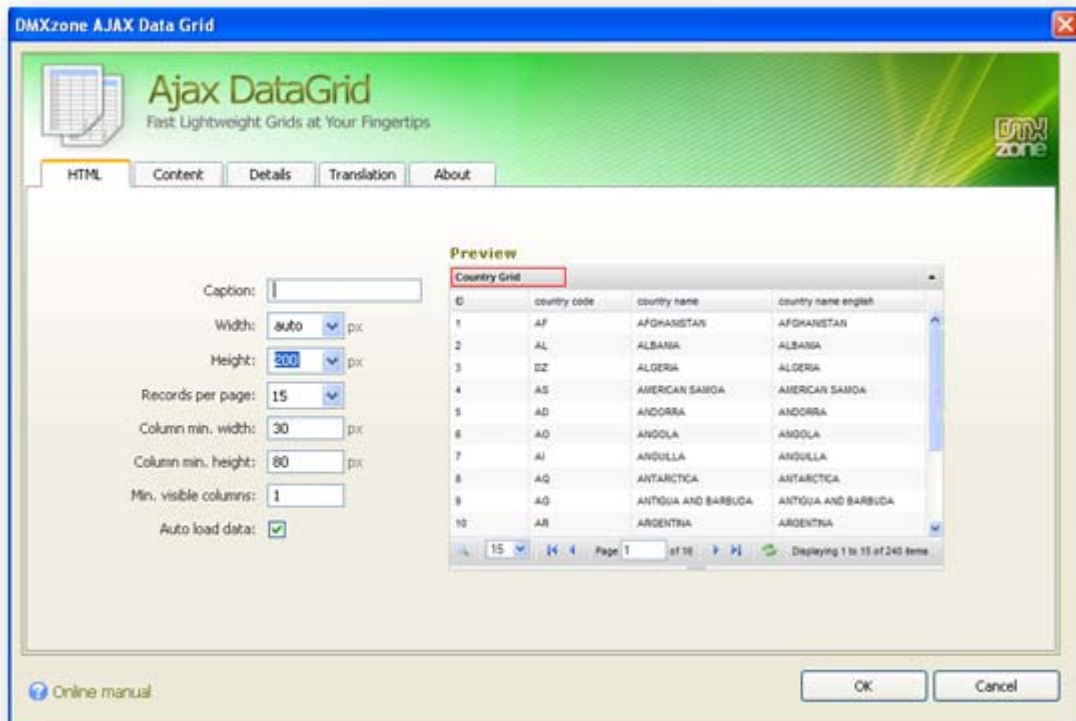
Building the Ajax DataGrid

1. **Create a new page**
   Create a new or open an existing HTML, ASP or PHP page and save the page. We create an ASP page.
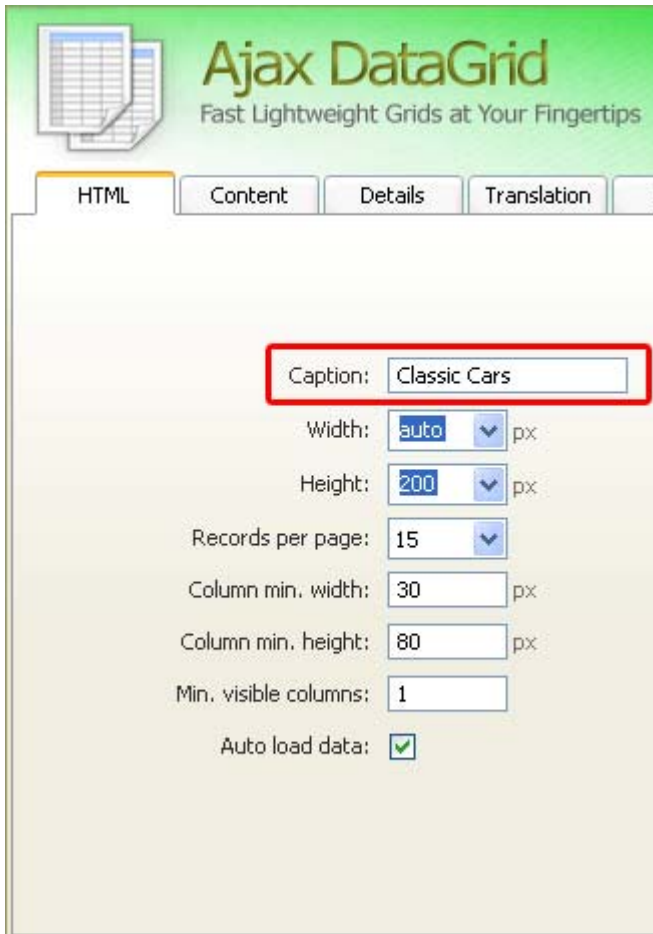
2. **Applying the extension**
   Choose the 🗐 icon from the DMXzone object bar to apply the extension. The following popup appears:

3. **Setting the Ajax DataGrid Options**
   At the HTML tab, the **Caption** sets the title of the Ajax DataGrid. The title is displayed at the top left of the grid. We set the caption to **Classic Cars**. Don't use \ in the Caption.
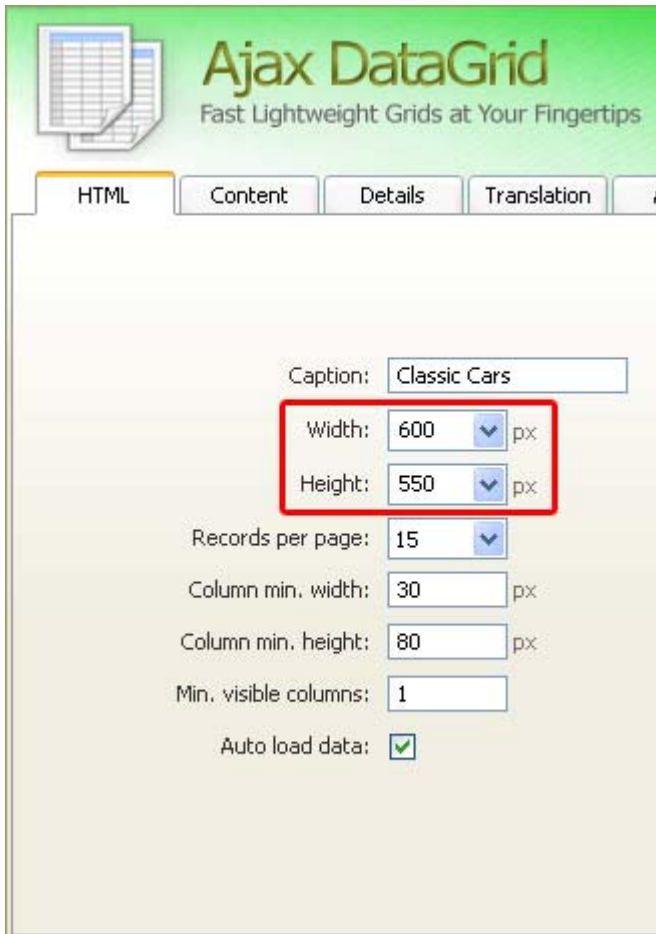
We leave the **Width** to **600** so the grid is large enough and can be resized. When the Width is set to **auto** the grid cannot be resized. We set the **Height** to **550** to make it high enough for a decent amount of content.
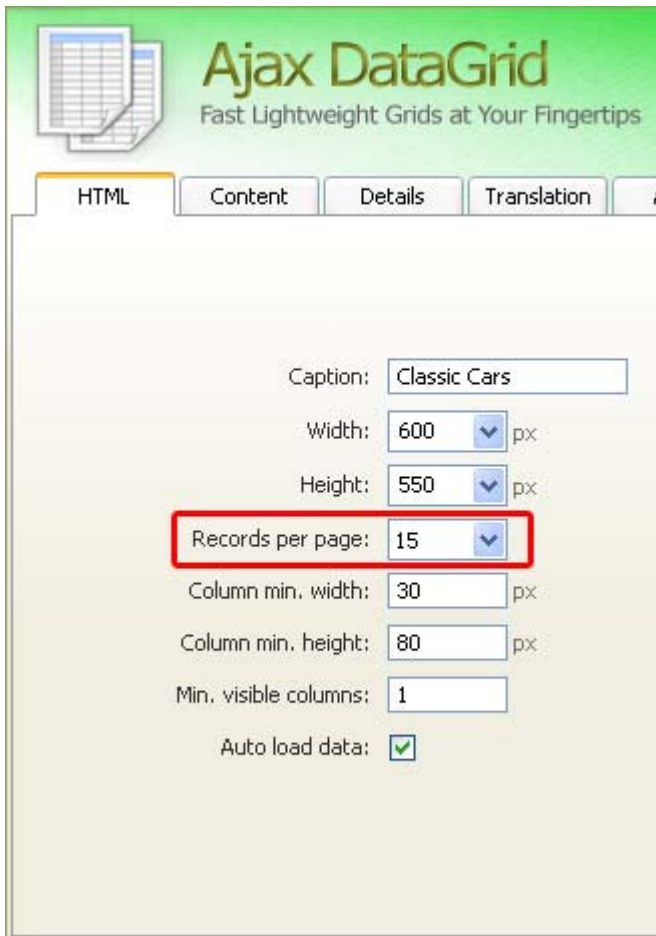
The **Records per page** set the number of rows per page, we'll leave it at **15.**
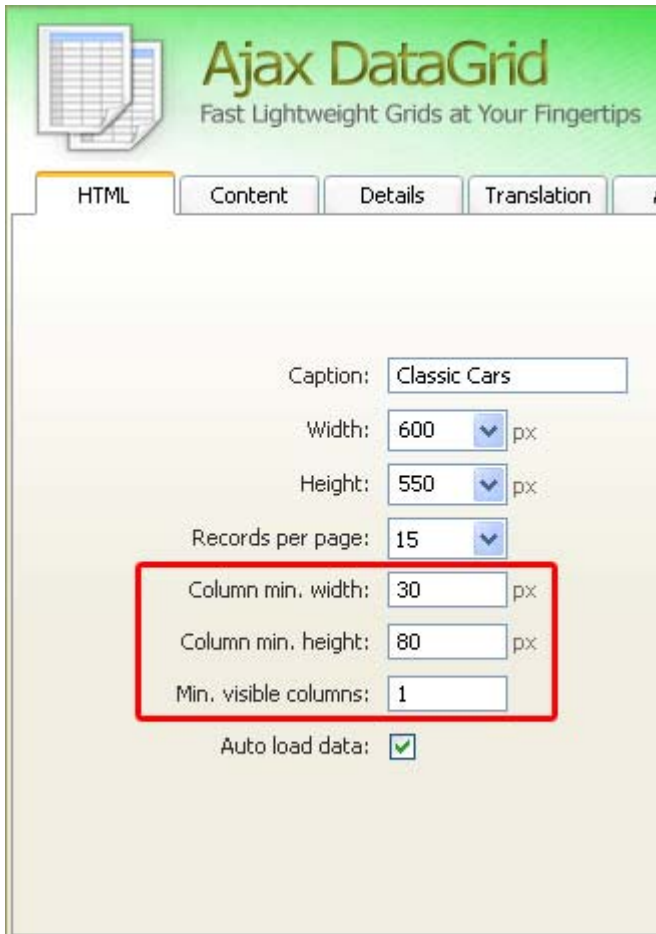
We leave the **Column min. width** at **30px** and the **Column min. height** to **80px**. These are the minimum dimensions of our columns. We leave the **Min. visible columns** at **1** to display at least one column when the user deselects the columns in the front end (see the preview to the right)**.**
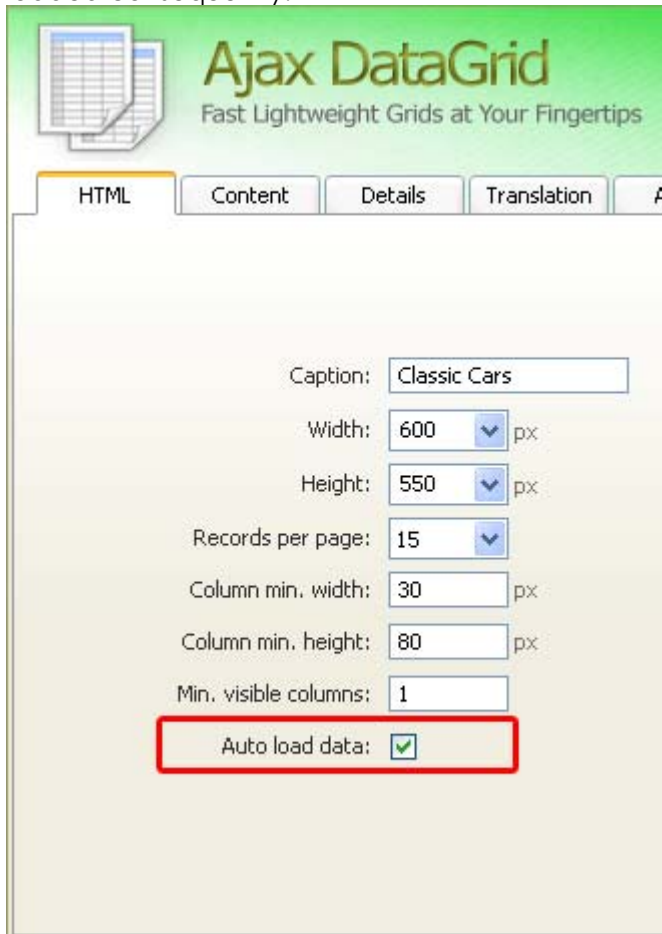
Unchecking the **Auto load data** is convenient when you are working with information that should be loaded consequently.

4. **Adding your Content to the Ajax DataGrid**
   Select the **Content** tab. **Connection** allows you to define what database should be used to populate the grid. Depending on if you have opened an ASP or PHP document, your connection options will differ. We have a database table called bio that we select using the drop down box.
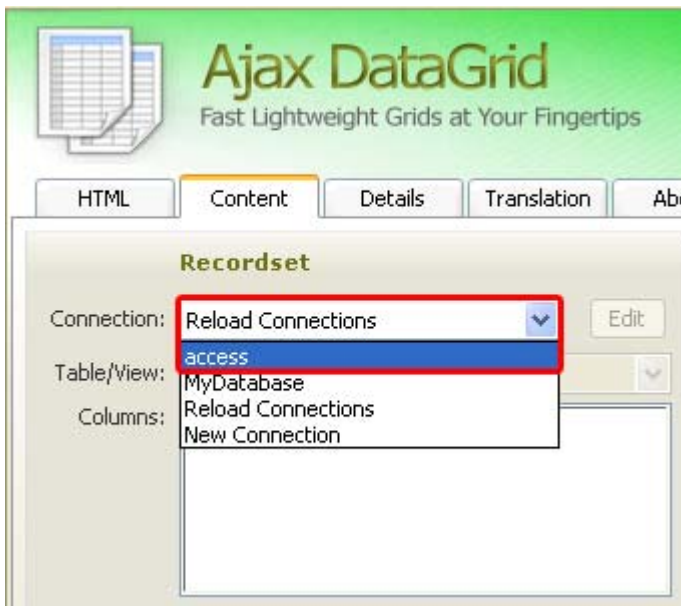
Optionally you can edit your connection string with the **Edit** button, we won't do this now.

Select the **Table** to choose which table to use from the database.

After that we select the **Columns** we want to add to our grid, we choose all Columns in our table.
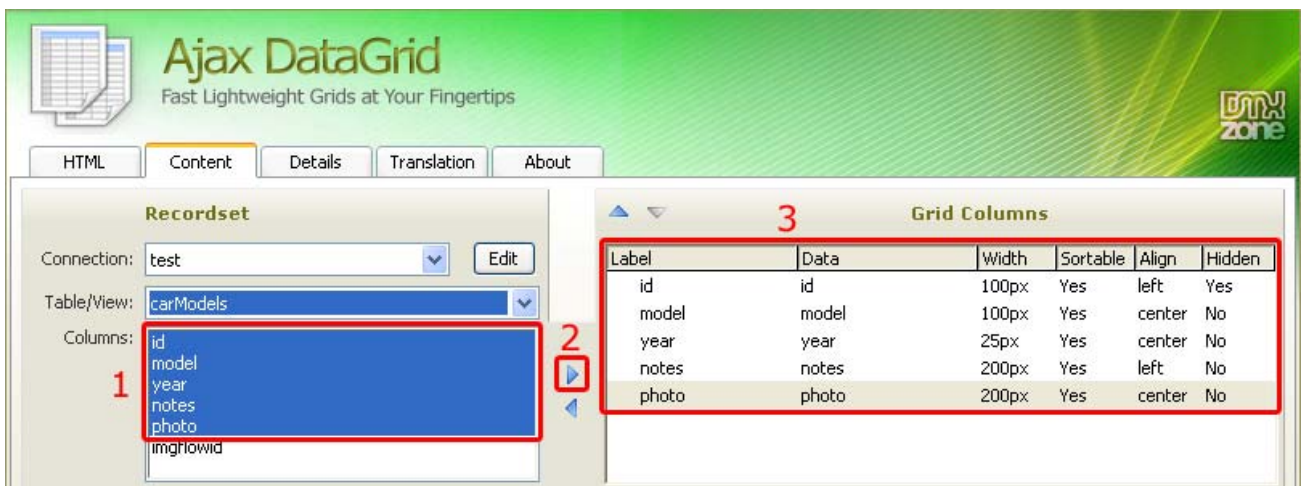
You can choose to **Filter** to filter on a specific value of one of the columns. We filter that we only want cars newer then **1938**.

Use the **Sort** option to sort on a specific field.

Choose whether to sort its content by **Entered Value** which is a static filter or by any of the others which

Page 21 of 68

are dynamic filters and are activated by input after the grid is loaded in the browser. We set it to **year** and **Entered Value.**

We press the **Test Query** button to display a preview of our data.
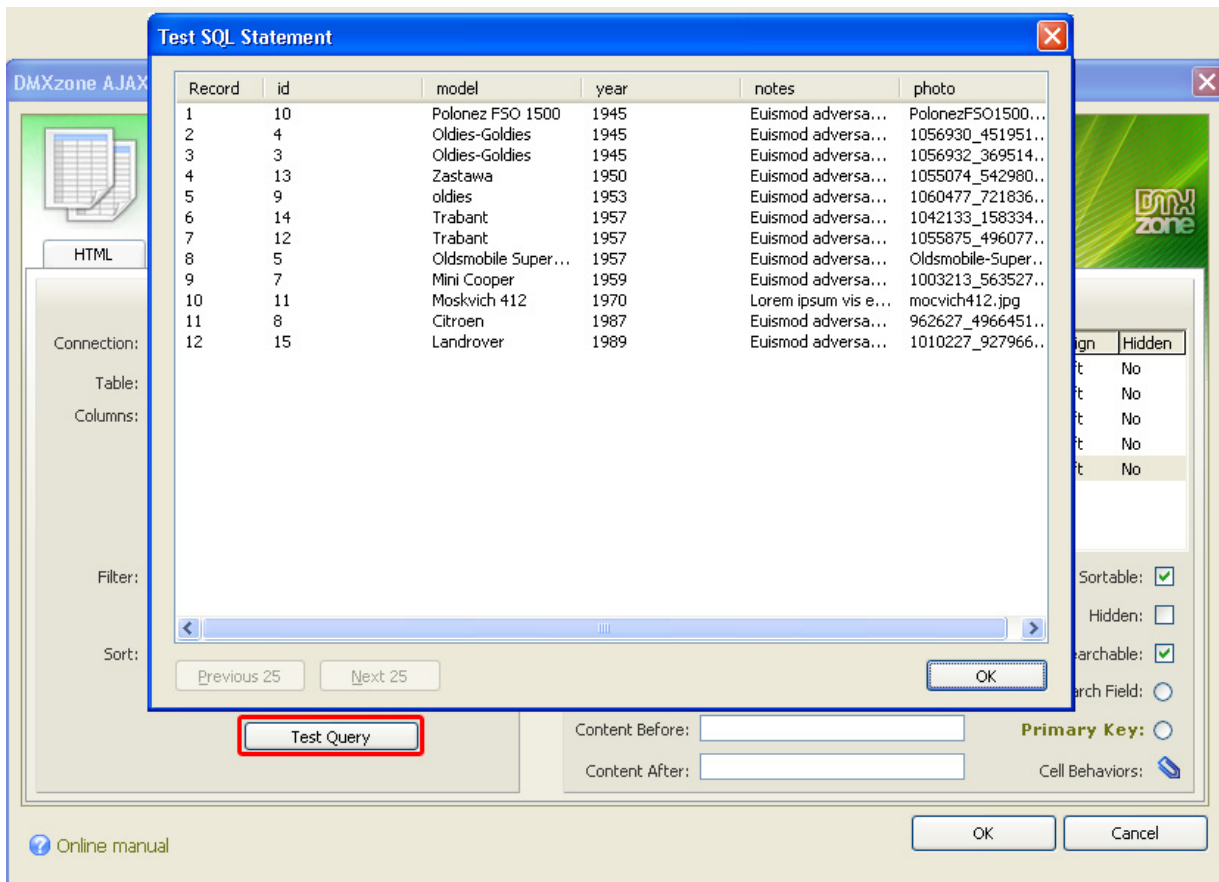
Press **OK** to close the Test SQL Statement.

In the **Grid Columns** display, you can select any particular column and change their names using the **Label** option. We we'll leave the Labels at their default value. You can assign formatting properties to its contents from the **Format** menu. The **AlphaCase** options allow you to convert the string to either *Lower* or *Upper* case; *First Letter Upper* capitalizes only the first letter of all words in the string, and *Capitalize* converts the first letter of the given string to upper case. With the **Trim** option, you can remove any spaces on the *Left*, *Right* or *Both* sides of the string. **Convert – New Lines to BRs** converts newline characters to HTML BR tags
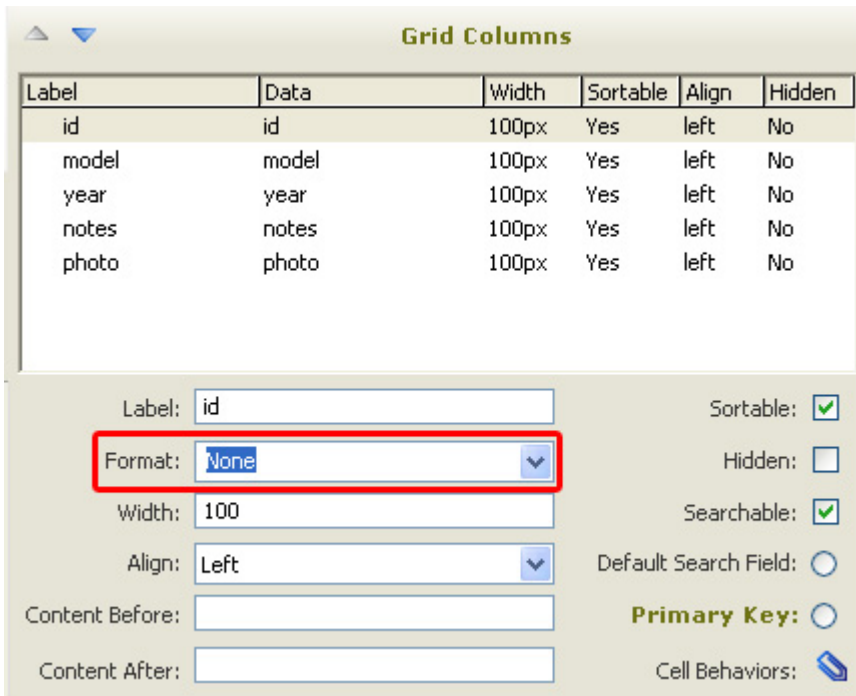
The *HTML Encode* option for **Encode** is needed under the following conditions:
HTML has special handling for characters like < and > symbols, so it doesn't work well with those characters where they shouldn't be. This can all be fixed by 'escaping' those characters. This process replacing them with a special character-code that browsers can interpret as the correct symbol, without actually embedding that symbol in your text. For example, the escaped character code for > is >.

The *URL Encode* option for **Encode** is needed under the following conditions:
URL links can be made up of only certain characters and a "space" for example cannot be used. Instead, *URL Encode* replaces those spaces or unrecognizable characters with ones it recognizes like "%,+, $, etc."

Encode - Raw URL Encode returns a string in which all non-alphanumeric characters except -_. have been replaced with a percent (%) sign followed by two hex digits. This is the encoding described in » RFC 1738 for protecting literal characters from being interpreted as special URL delimiters, and for protecting URLs from being mangled by transmission media with character conversions (like some

email systems). **Decode - URL Decode** decodes an URL-encoded string. **Decode - Raw URL Decode** returns a string in which the sequences with percent (%) signs followed by two hex digits have been replaced with literal characters. **HTML - Strip HTML** removes all HTML tags and leaves plain text only.

We set the **Format** to **None**.

For a PHP document with MySQL connection, the formatting options are significantly more extended than in an ASP document although it is possible to write your own code and by inserting it into the Site/ajax_servers/custom_lib.php file get additional options.

Width sets the **Width** of the column, we'll leave it at **100.** This width excludes the padding and margins that are used in your table, so the full cell width will be a little bit bigger (for example 112 pixels).

Align allows you to set how the data in the column is outlined, we set it to **Left.**

We want to display the photos in our database as images rather then a text that list the path, to do this we select the photo column and enter **** in the **Content After**. The images tag together with the path to the images will make sure the images will be displayed while the width specifies the width of the images.

The Images are now displayed in our grid:

id	model	year	notes	photo	
6	VW Beetle	1938	Euismod adversarium		
10	Polonez FSO 1500	1945	Euismod adversarium		
4	Oldies-Goldies	1945	Euismod adversarium		
3	Oldies-Goldies	1945	Euismod adversarium		
13	Zastawa	1950	Euismod adversarium		
9	oldies	1953	Euismod adversarium		
14	Trabant	1957	Euismod adversarium		
12	Trabant	1957	Euismod adversarium		

Classic Cars

15 Page 1 of 1 Displaying 1 to 13 of 13 items

You can enable or disable the sorting of columns by checking or unchecking the **Sortable** checkbox. We leave it checked.

Optionally you can hide columns by default using the **Hidden** option. We leave it unchecked.

Define whether a column should be searchable or not by checking or unchecking the **Searchable** option. We leave it checked. Note that you need to check the **Use pager** option in the **Details tab** to use this option. We make all our fields searchable.

If you have more then one Searchable column, you can use the **Default Search Field** option to specify if the default column that will be searched on. We do this for the year column.

Set the **Primary Key** to the column that holds a unique identifier for its record. We set it to ID.

Attaching a behavior to a cell allows for additional actions, effects, and links, to name a few, to be executed when a cell is clicked or double clicked for example. We won't specify this now but we'll explain this in another tutorial further on in the manual.
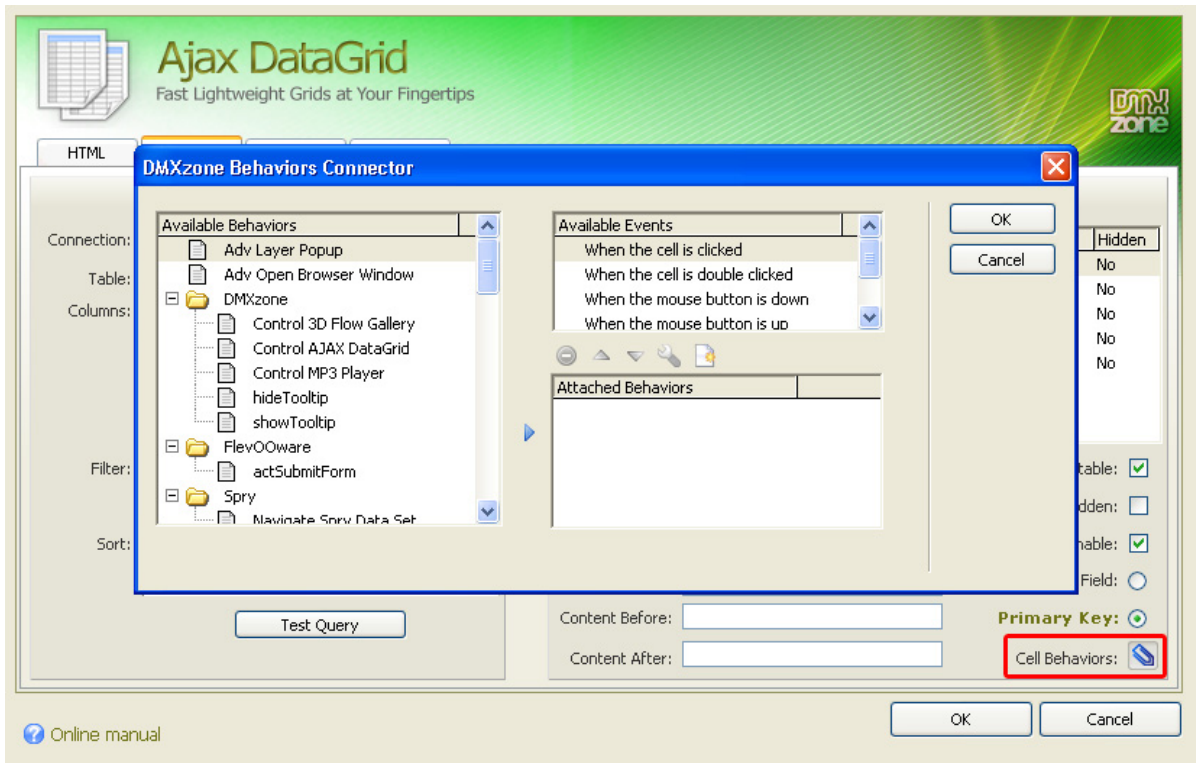
You can use the ▲ ▼ to sort the order of the columns. We'll leave the default order.

Setting the Details options
The **Details** tab allows for DataGrid fine-tuning as it relates to the visual output. As you click each option, the **Preview** on the right shows what it pertains to with a red outline.

We check the **Apply odd even stripes** field to highlight our alternate rows.

Check the **Vertical stripe** field to highlight the sorted column.

We check the **Resizable** field to enable the resizing of our columns. You need to set an exact with and height for this option to work.

We select **Nowrap** to make sure our text is not cut off.

We enable the **Use pager** option to include a navigation bar to browse to our content. It's also necessary to turn it on to enable the search trough columns we enabled earlier.

We check the **Show Toggle Grid Button** to display the button on the right top side of the table.

We also check the **Use results per page selector** field to enable the user how many records should be displayed per page.

We enable the **Show "Choose columns" Popup** checkbox to allow the user to select which columns he wants to display.

Display Errors lets you choose whether you want errors to be displayed
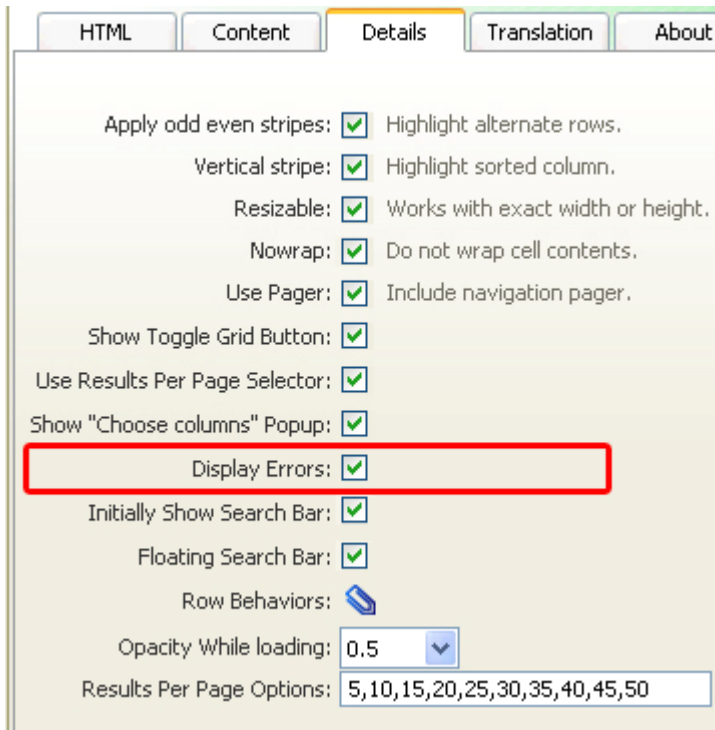
The option, **Initially Show Search Bar** displays the search bar when the grid loads, without additional clicking.
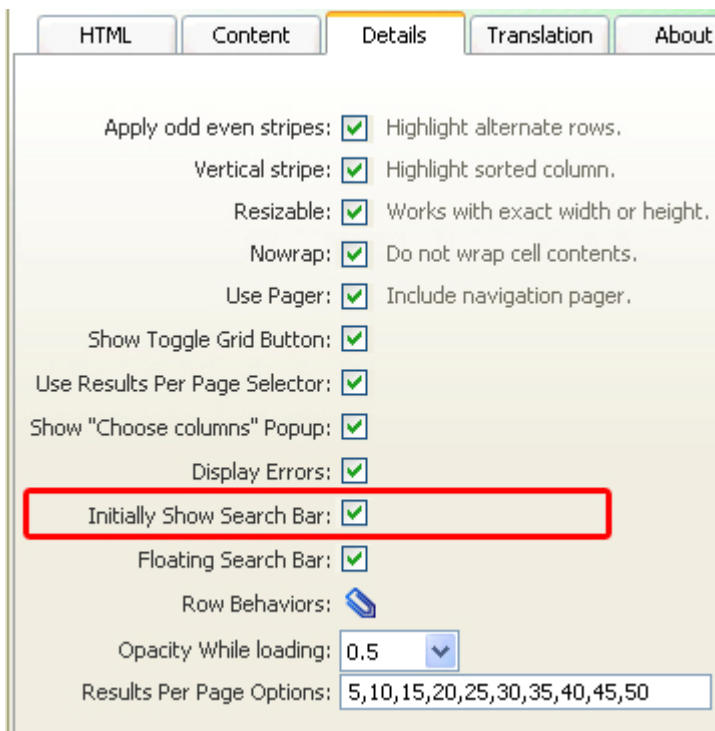
Floating Search Bar keeps the grid from resizing when the search bar is displayed. It just shows up on top of the contents.

Results look like this:

Checked Floating Search Bar Option
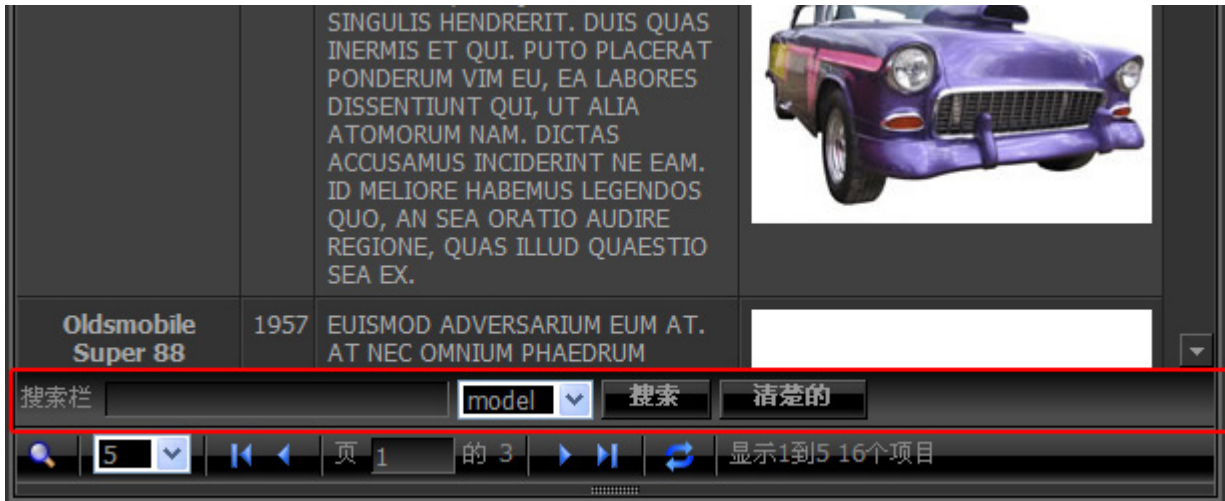
The Search Bar is Floating Above The Grid

Unchecked Floating Search Bar Option
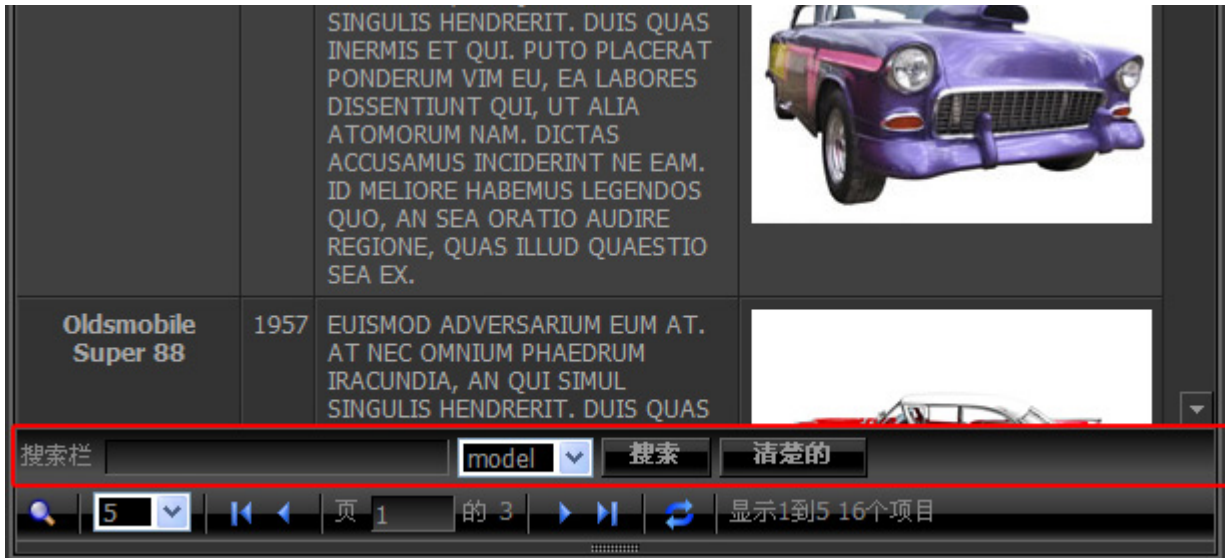
The Search Bar is Displayed Below The Grid

Row Behaviors allow you to add Behaviors to the rows in addition to Cell Behaviors which we won't use now but are explained in a separate chapter.
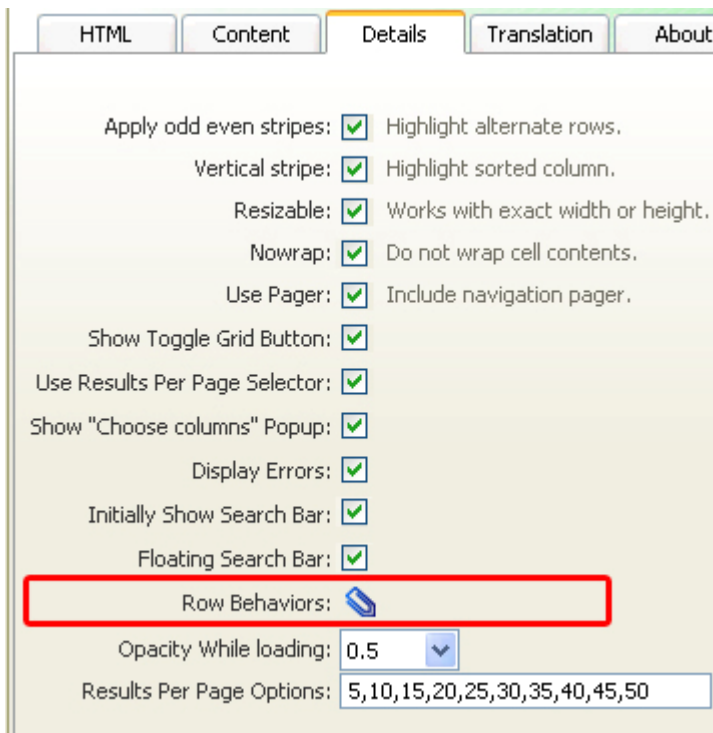
We set leave the **Opacity while loading** of our grid at 0.5 to make the grid half transparent while it's loading.
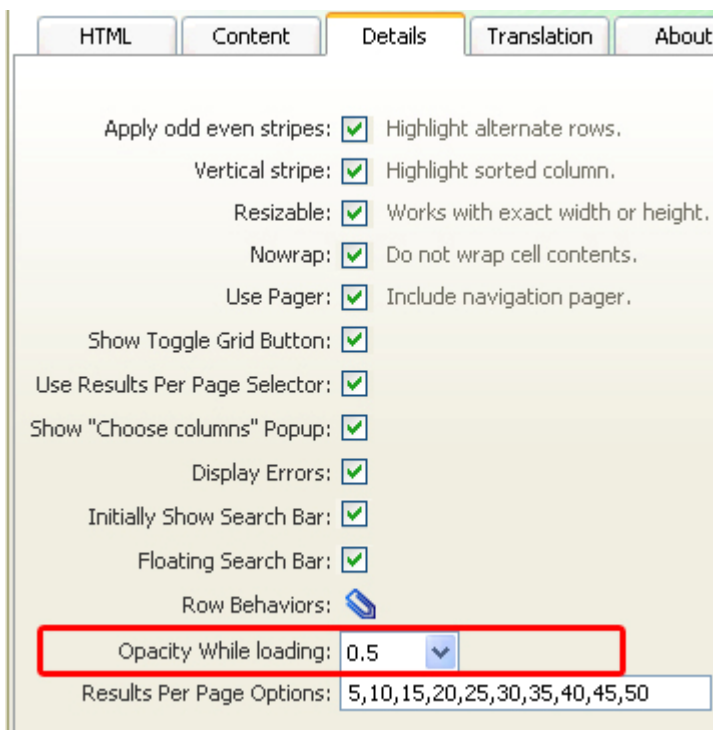
The **Results Per Page Options** allow you to control the allowed choices for the results per page menu

Setting the Translation options
The **Translation** tab allows translation of the menus in other languages (supports UTF8 scripts) as well as customization of the wording of the text. As you click each option, the **Preview** on the bottom shows what it pertains to by highlighting it.

The **Status test (on Connection Error)** is the text that shows up only if a connection error occurs. You can rewrite the message in your own words or language if you choose.

The **Status text (default)** is the descriptive text that shows up on the bottom right as default.

The **Status text (while processing)** option lets you change the message the user sees while the data is loading.

You can edit the text displayed if there are no search results with the **Status text (if no results)** option.

The **Text before the pager** option lets you change the word before the page numbering.

The **Text after the pager** option lets you change the word after the page numbering.

The **Search Field Label** option lets you change the text that appears before the search field.

The **Search Button Label** allows you to customize the text that appears on the Search button.

The **Clear Button Label** allows you to customize the text that appears on the Clear button.

If you want to change the tooltip that appears at the Toggle Grid Button you can enter your own text in the **Toggle Grid Button Tooltip** field.

If you want to change the tooltip that appears at the Hide/Show Columns button you can enter your own text in the **Hide/Show Columns Tooltip** field.

Press **OK** to generate the grid. The result should be similar to the one displayed below:

Classic Cars					▲
id	model	year	notes	photo	
6	VW Beetle	1938	Euismod adversarium		
10	Polonez FSO 1500	1945	Euismod adversarium		
4	Oldies-Goldies	1945	Euismod adversarium		
3	Oldies-Goldies	1945	Euismod adversarium		
13	Zastawa	1950	Euismod adversarium		
9	oldies	1953	Euismod adversarium		
14	Trabant	1957	Euismod adversarium		
12	Trabant	1957	Euismod adversarium		

🔍 | 15 ⌄ | ◀◀ ◀ Page 1 of 1 ▶ ▶▶ | 🔄 | Displaying 1 to 13 of 13 items

The Basics: Editing the Ajax DataGrid with the Property Inspector

Introduction

In this chapter we're going to edit the properties of the Ajax DataGrid using the property inspector of Dreamweaver. Click on the preview of the Ajax DataGrid in the design view to display its properties. Below is a screenshot of the property inspector and a screenshot of the final result.

Editing the Basic options
Editing the Ajax DataGrid Properties

The **Caption** displays at the very top of your table and is the first option that can be set in the Properties Inspector. We'll leave it at **Classic Cars**.

ID sets the ID of the Ajax DataGrid. This entry field is useful when you use more than one Ajax DataGrid, that way the different galleries can be identified. We leave it to its default value.

The **Style** option lets you choose between the **Flexigrid** (default) style and the **Windows XP** style. We set it to the **Windows XP** style.
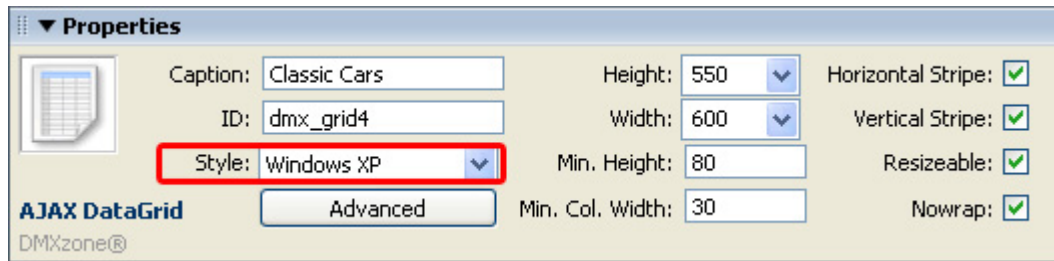
The following popup appears:

Press **OK**.

The **Advanced** button, right under the **Style** field opens up the Ajax DataGrid popup where you can edit all settings that you have defined in the previous tutorial.
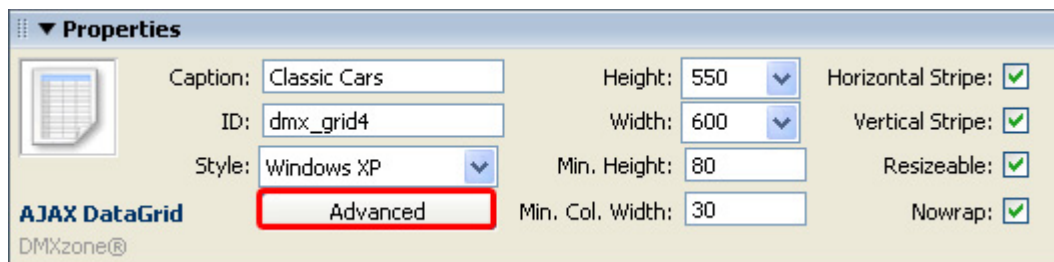
Height is related to the height of the Ajax DataGrid when it is first loaded. We'll leave it at **550** (pixels). **Width** sets the width of the Ajax DataGrid. Take the padding and margins in the standard CSS into account when calculating how many columns will fit in the grid. We'll leave the **Height** at **600** (pixels). We don't set the Width to auto as we want the user to be able to resize the grid manually.
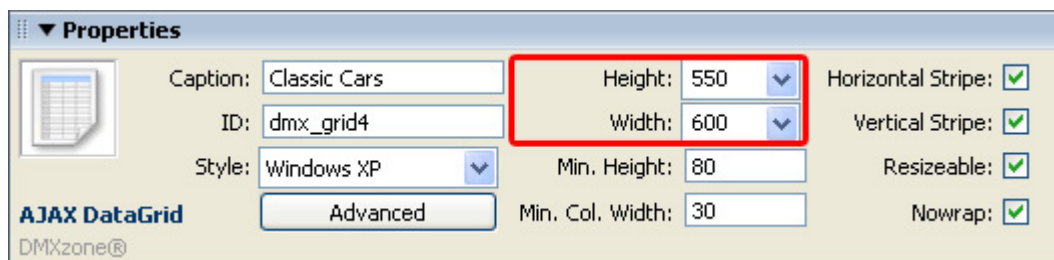
Min. Height sets the minimal possible height of the table when resized manually. **Min. Col. Width** sets the minimal possible width of each column when resized manually. We'll leave Min. Height to **80** (pixels) and Min. Col. Width to **30** (pixels).

We uncheck the **horizontal Stripe** option to prevent our even (2nd, 4th etc) columns from having a darker gray colouring. We do leave the **Vertical Stripe** checked to highlight the sorted column.

We leave the resizable field checked the **Resizable** field, to enable the resizing of the grid.

We select **Nowrap** to make sure our text is not cut off.

Save your files, upload them to your server and the end result would look similar to this:

Advanced: Using the Cell Behaviors

Introduction

In this chapter we're going to add a popup when the user clicks an image with a larger version. We'll use the grid that we've created in the earlier tutorials. Below is a screenshot of the end result.

Creating the Popup

1. **Open the behaviors connector**
   We open the Ajax Datagrid dialog with the **Advanced** button, we select the column where we want to add our cell behaviour to and we press the **Cell Behaviors** button.
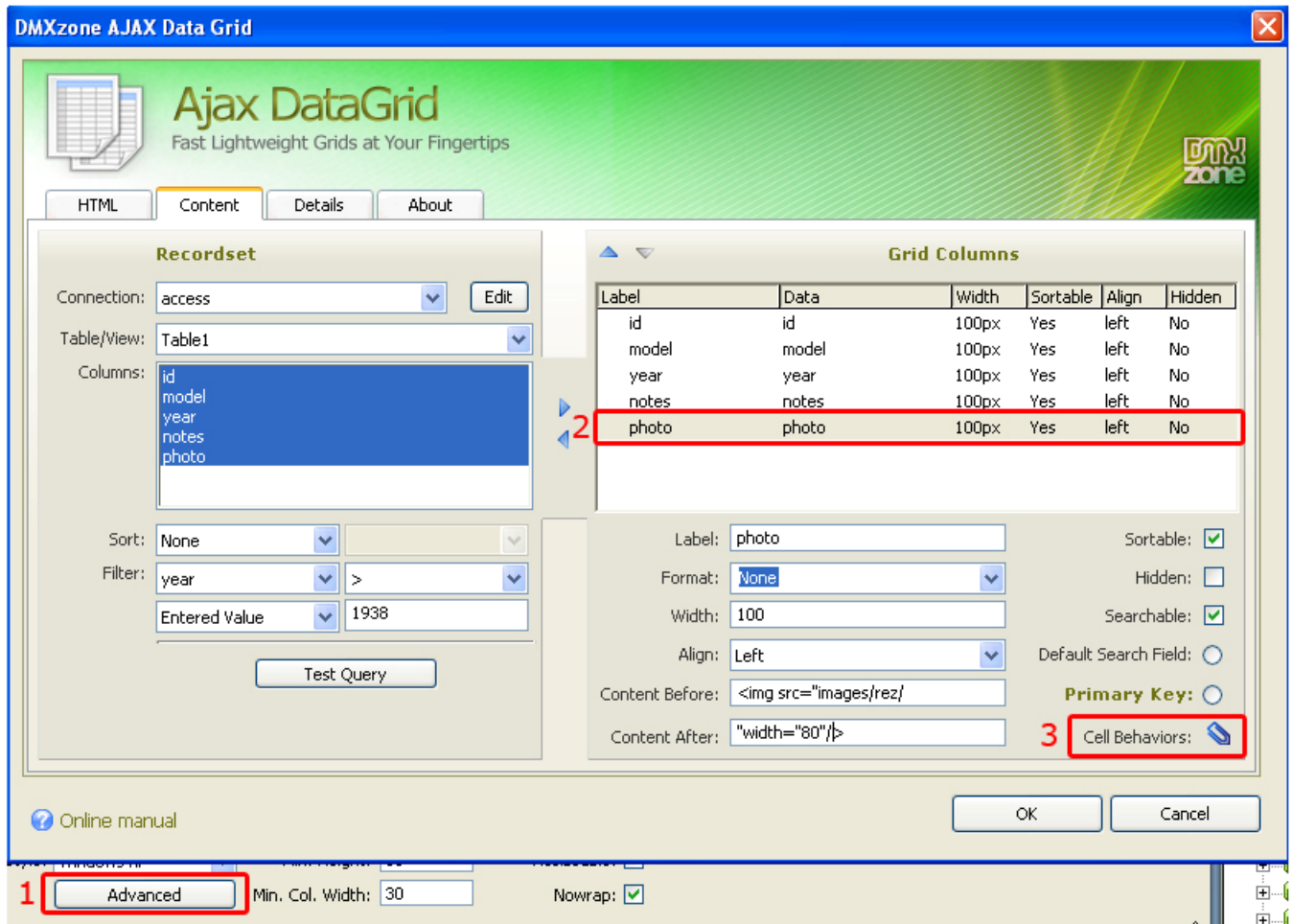
2. **Adding the popup**
    We select the **Adv Layer Popup** behaviour (you need [Advanced Layer Popup](#) to select a similar behavior), if you don't have it you can select another behavior.

    We enter the path to our images in the **URL to Display** box and enter the number of the column where you want to display the popup between {}. The counting starts at 0, the photo column is listed on the 5th place so it gets number 4. So we enter **images/rez/{4}** in our URL to Display box.
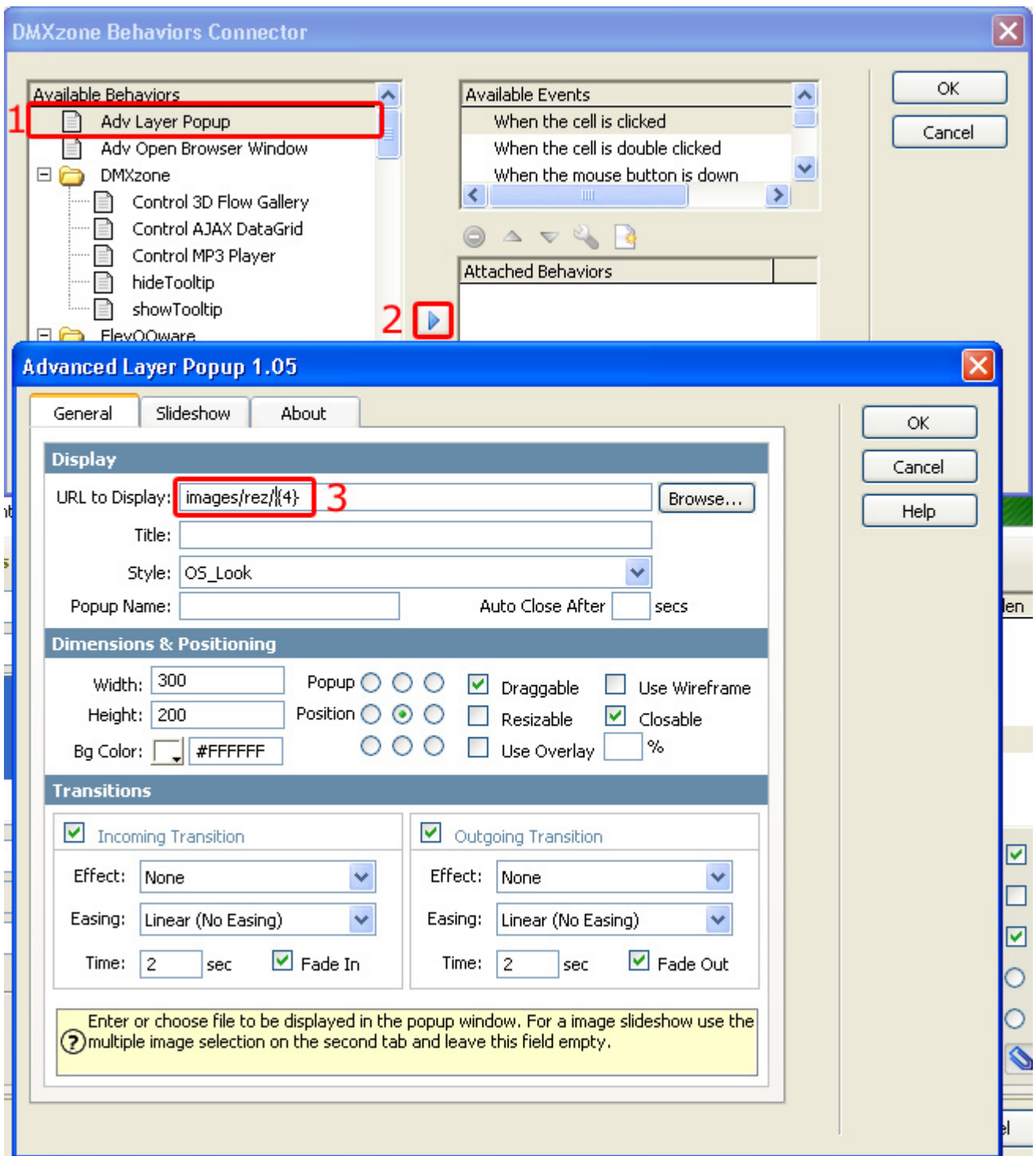
Press **OK**. The Behaviors are listed in the attached behaviors box.
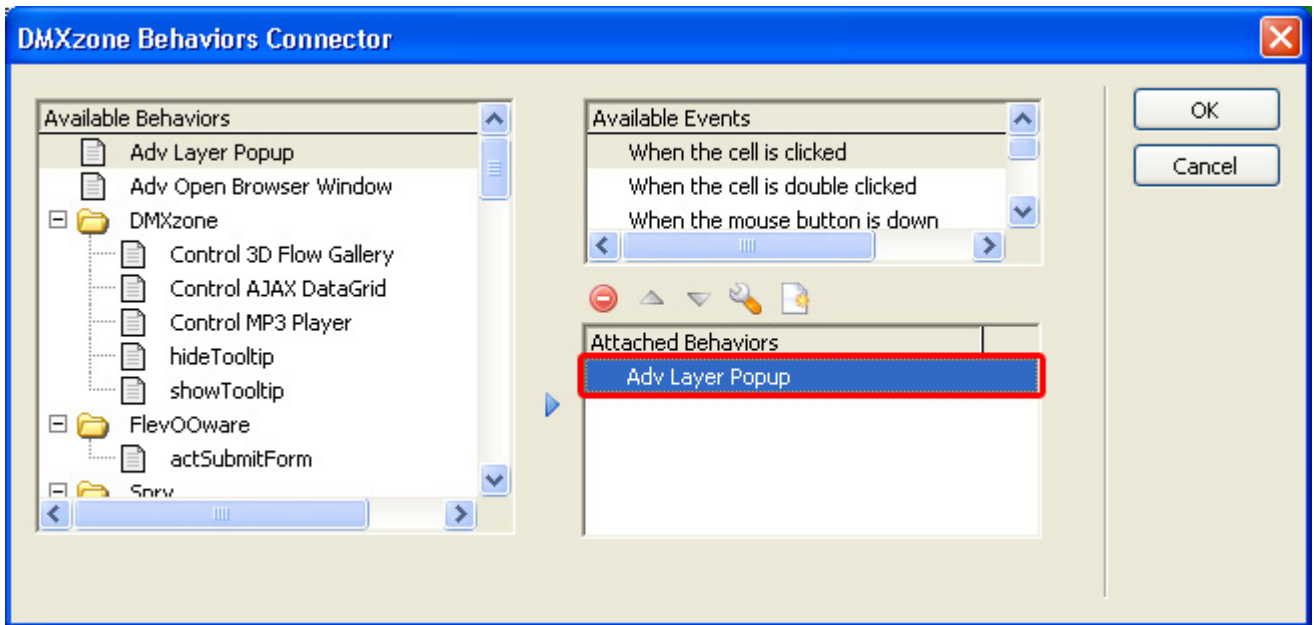
Press **OK**. To apply the behavior.

The **Cell Behaviors** button turns red to indicate we added a behavior.

Press **OK**. To apply the settings.

Save the files and preview your grid. Click on the photos to open them in a popup.

Advanced: Using the Row Behaviors

Introduction

In this chapter we're going to explain Row Behaviors by creating a <u>Master Grid</u>. When the user clicks on the row of the top grid (an e-Magazine in this tutorial's case), the bottom grid will become active and display the articles of the selected e-Magazine. We'll use two different grids for this tutorial. Below is a screenshot of the end result.

Creating the Popup

3. **Open Dreamweaver**
   We open two grids that we have already created and connected to a database and place them on one page. We select the first grid.
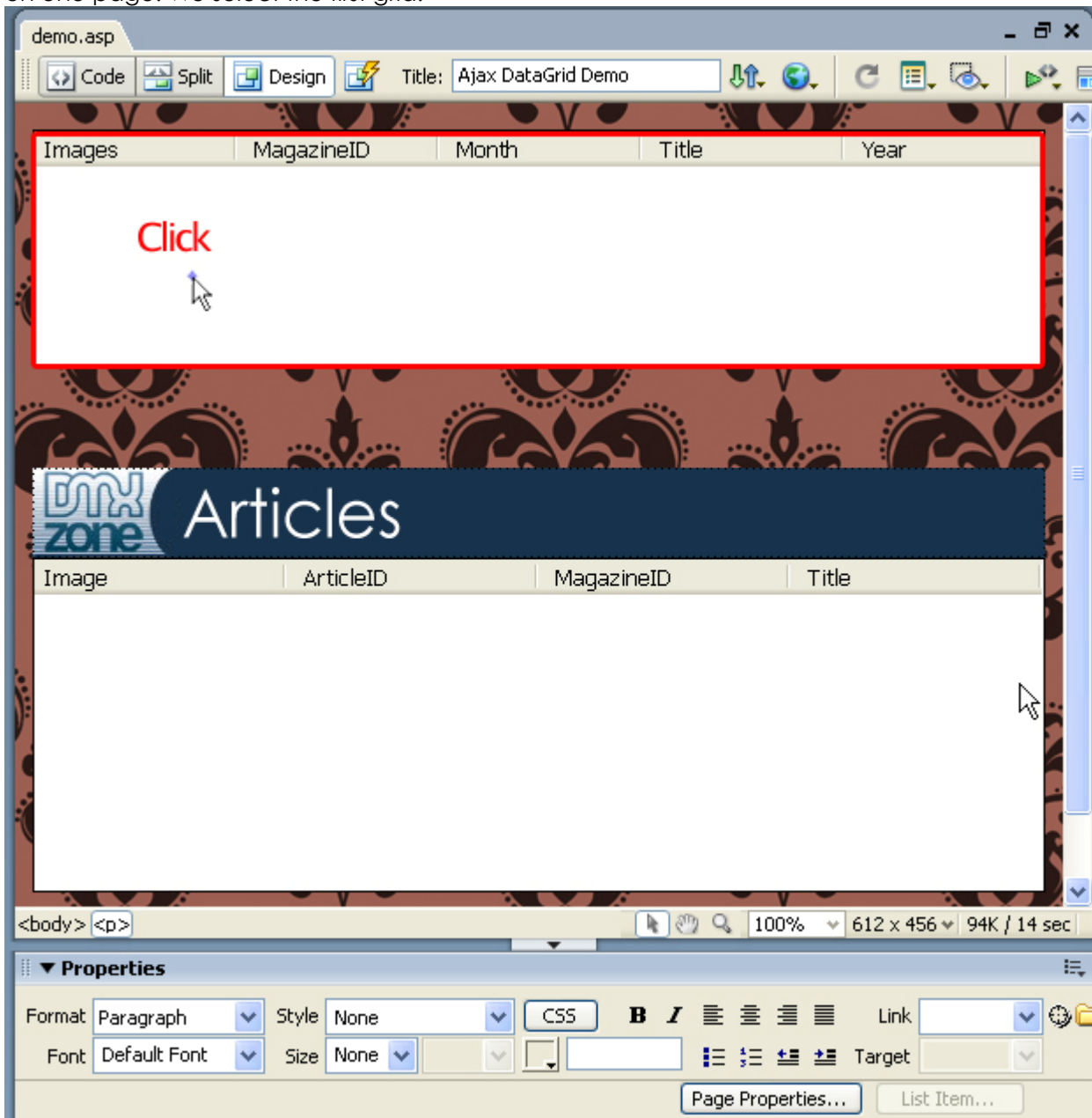
4. **Open the behaviors connector**
   We open the Ajax Datagrid dialog with the **Advanced** button and we go to the **Details** tab. We select **Row** Behaviors. The **DMXzone Behaviors Connector** appears. column where we want to add our cell behaviour to and we press the **Cell Behaviors** button.
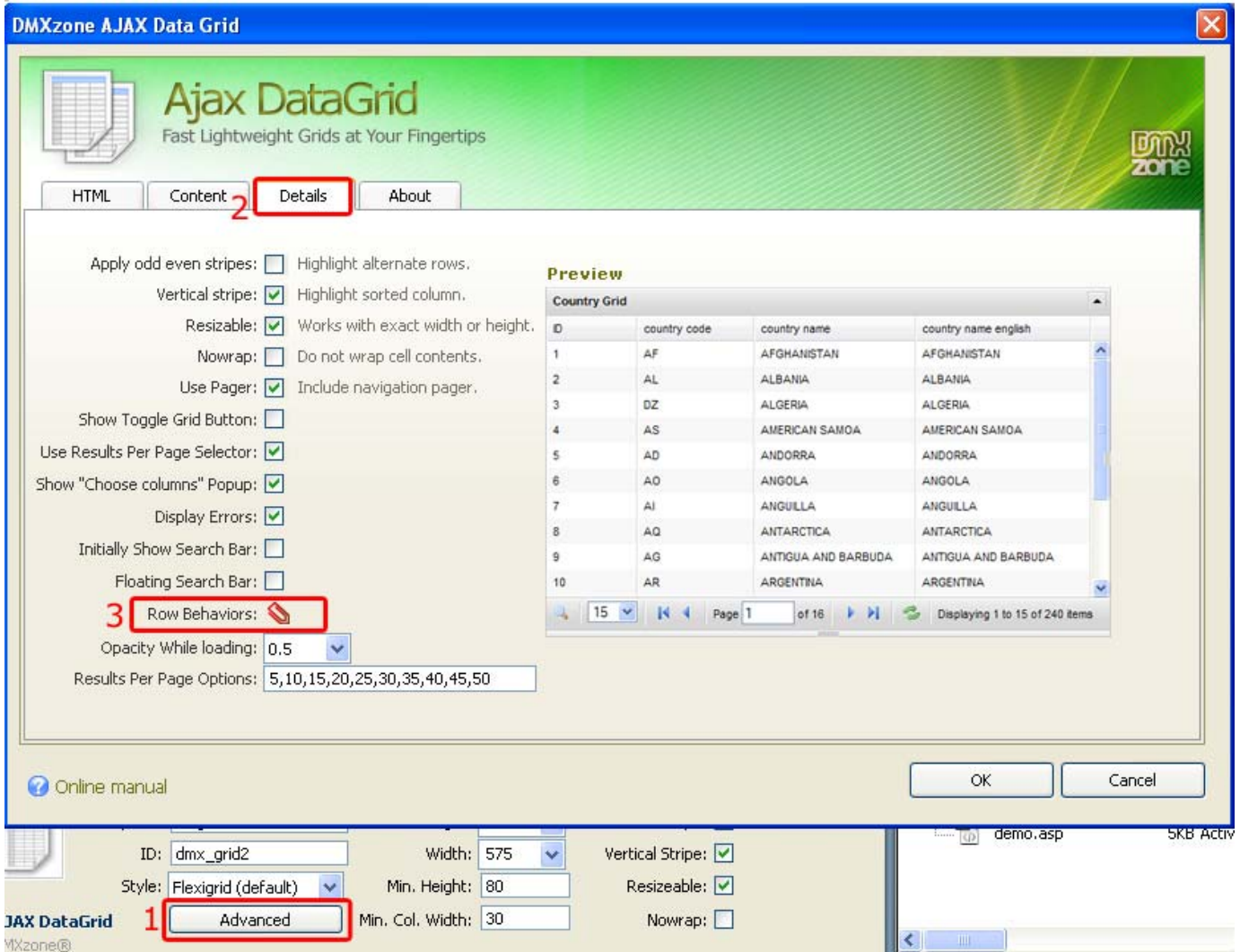
5. **Adding the Behavior**
   We select the **Control Ajax Datagrid** behavior and we apply it to the event "when the row is clicked".
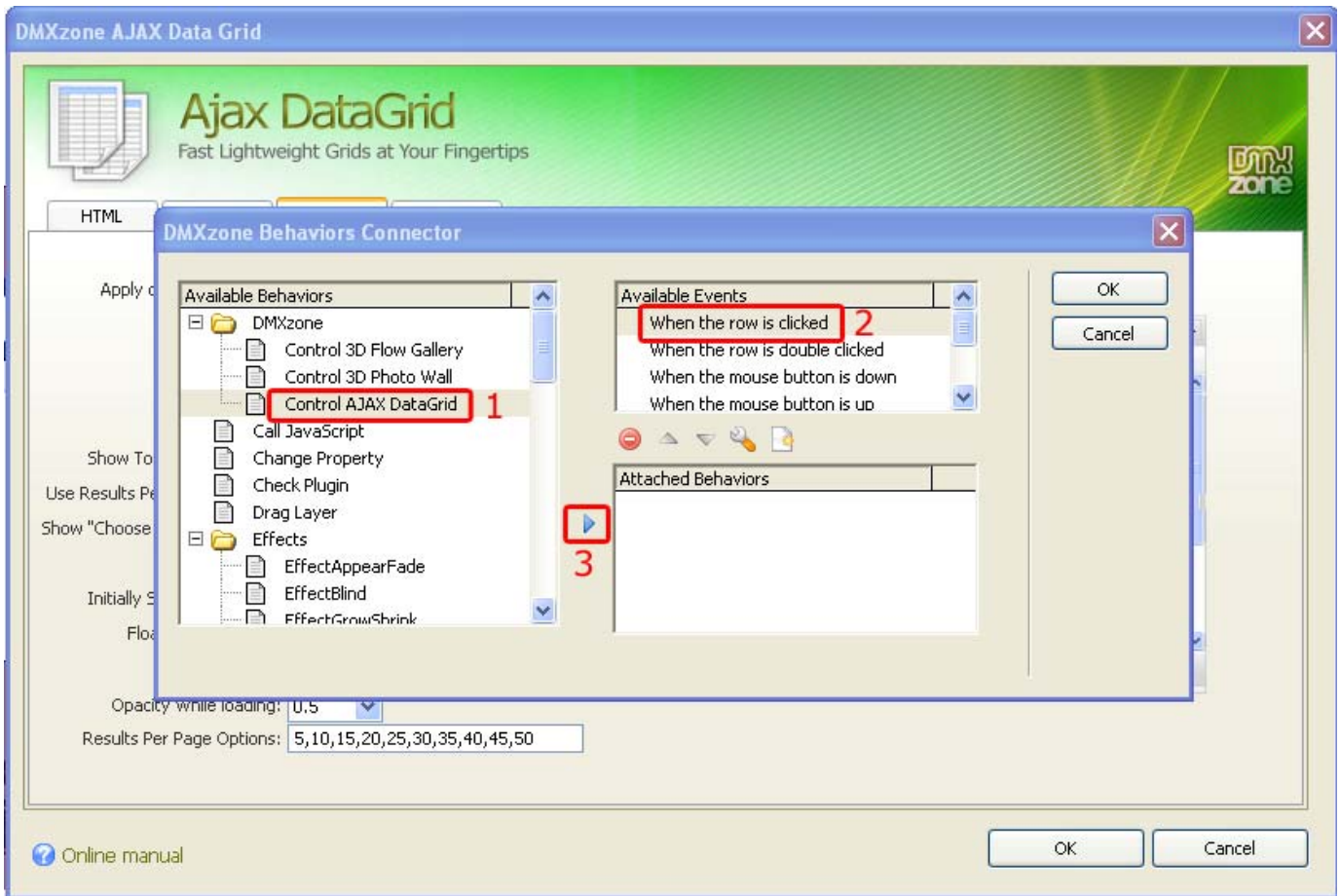
A window of the behavior pops up. We choose the ID name of the second grid and set the **Set Grid Filter** item. In our database we have two tables: one with the e-Magazines that are identified by the column called MagazineID. This table is used by the first grid. The second table lists all articles. We also have a column here that specifies in which e-Magazine the article is published. This column is also named MagazineID. We link the two columns together.
We enter the name of the column of the first grid in the *Column:* field and we enter the value of the second grid between curly brackets in the *Value:* field.

Press **OK**. The Behaviors are listed in the attached behaviors box.
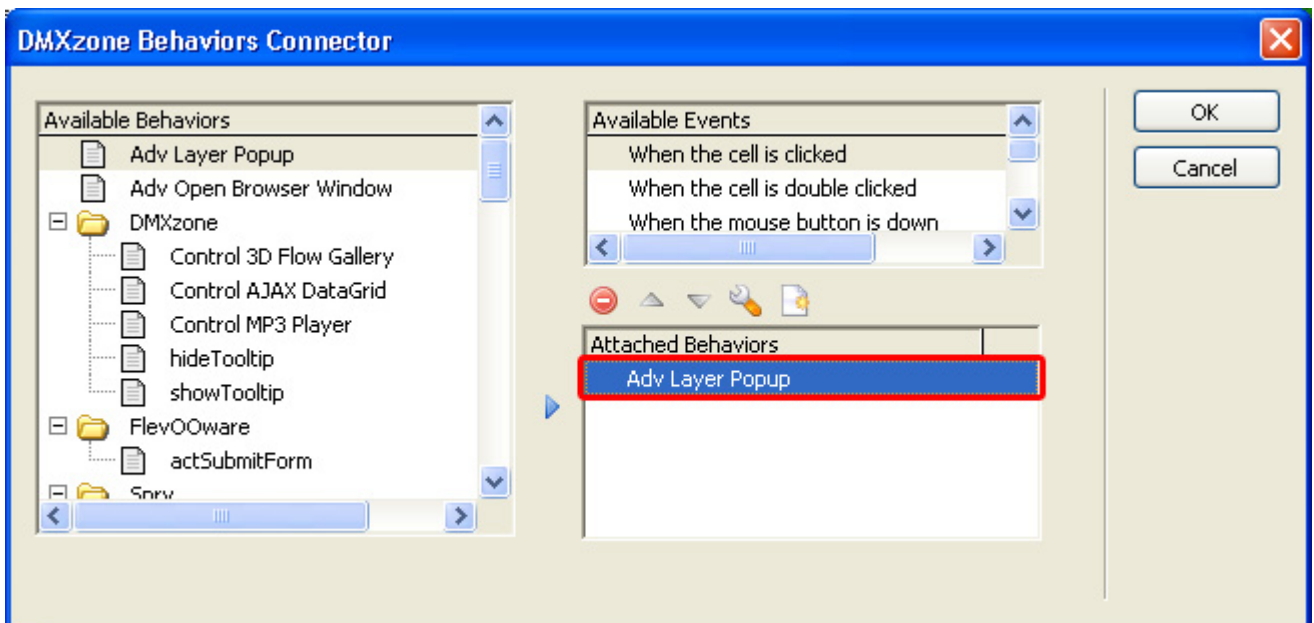
Press **OK**. To apply the behavior.
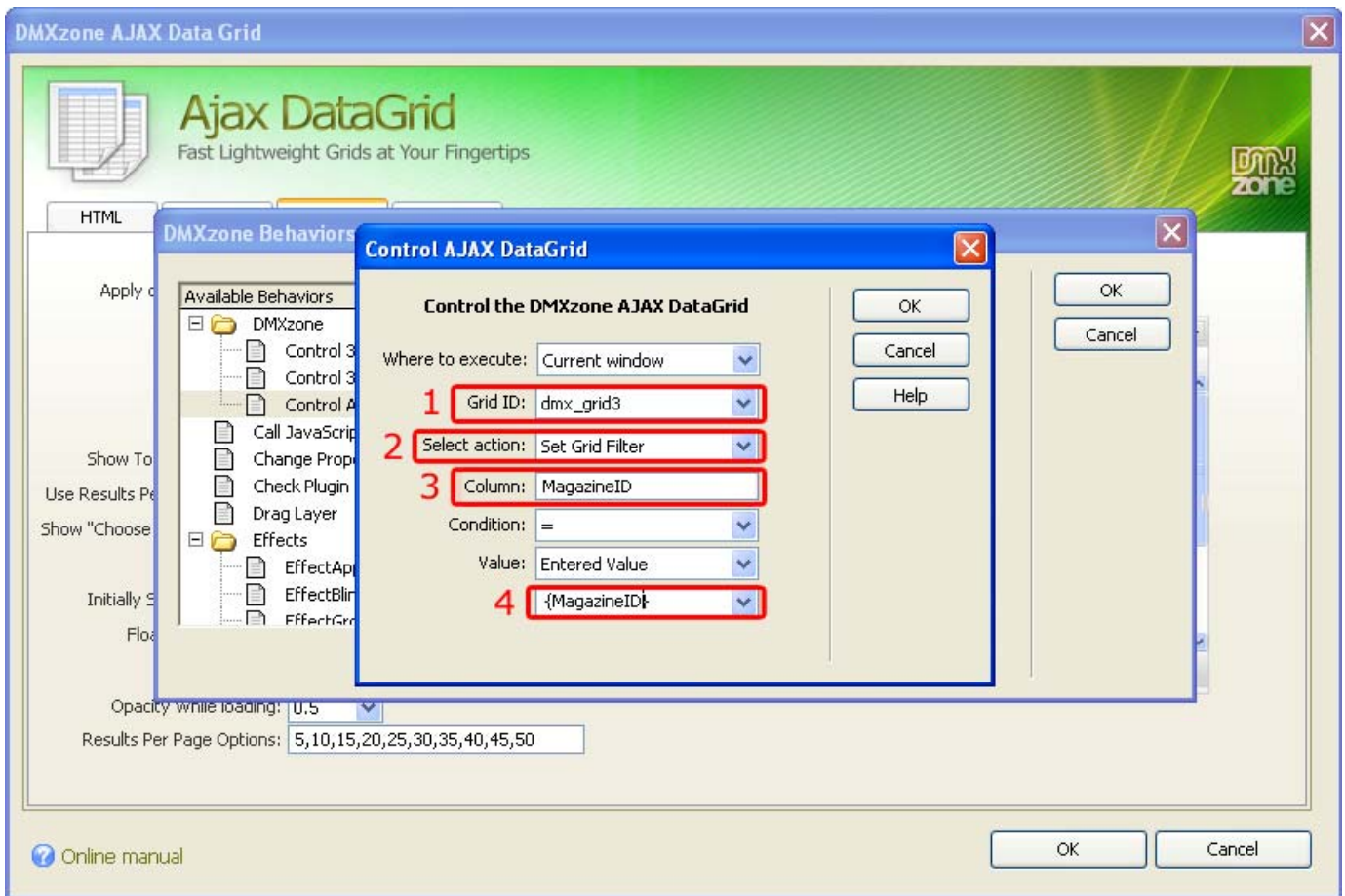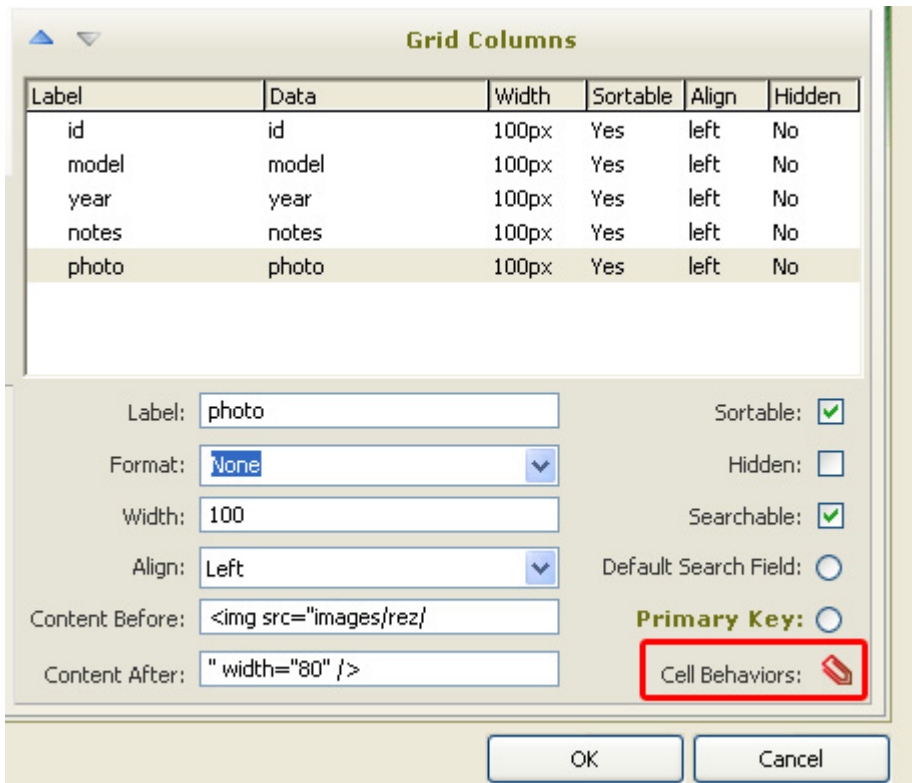
The **Row Behaviors** button turns red to indicate we added a behavior.

Press **OK**. To apply the settings.

Save the files and preview your grid. Click on the photos to open them in a popup.

6. **Modifying the second Grid**
   Now select the second grid to change its settings. We want to prevent it from loading all articles from all e-Magazines when there is no magazine selected.

   Uncheck the **Auto load data** checkbox

   Press **OK** and you are set.

Advanced: Creating a Custom Navigation Using Behaviors
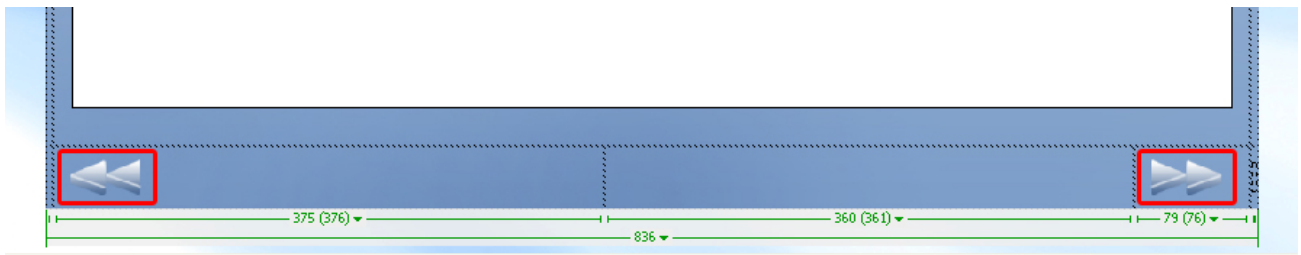
Introduction

In this chapter we're going to show how you can control the grid with behaviors.
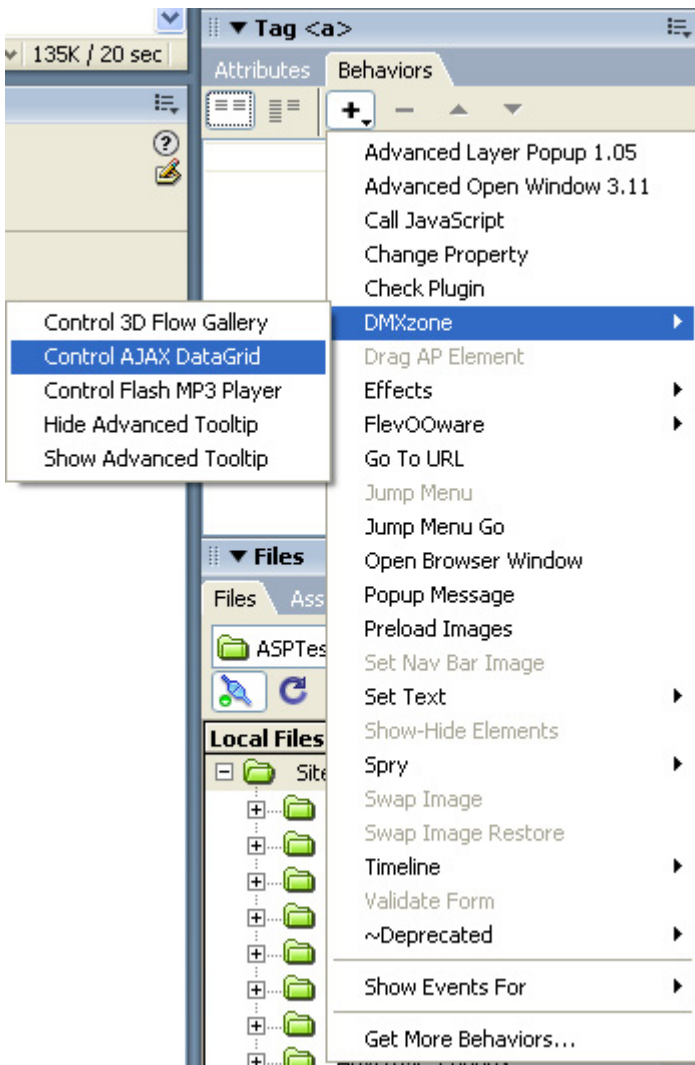
Adding the behavior

1. **Add a navigation object onto your page**
   We'll add buttons (.jpg images) to which we will attach the behavior.

2. **Specifying the behavior**
    Select your button, go to the **Behaviors** tab and select **DMXzone** and choose **Control AJAX DataGrid**.

3. **Setting the Behavior Options**
   We set the **Where to execute** to **Current window** as we don't use any frames. We select the **Grid ID** and specify that the grid needs to navigate to the next page.
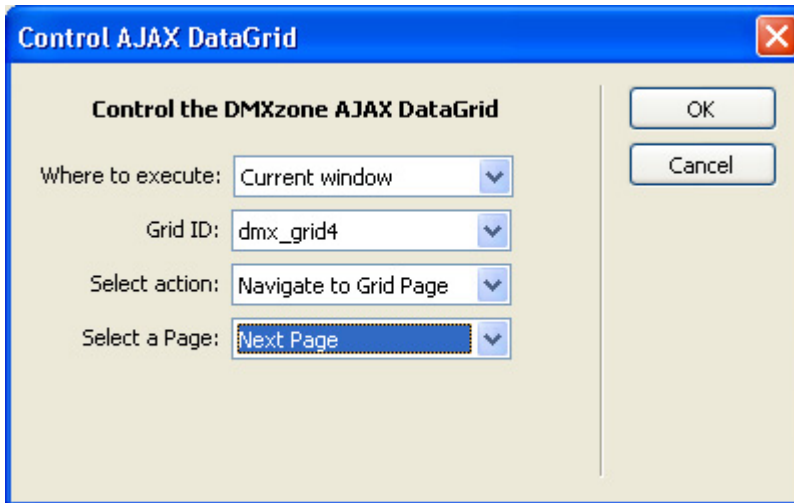
Press ok and save and upload your files. When we press our button we'll navigate to the next page in our grid.