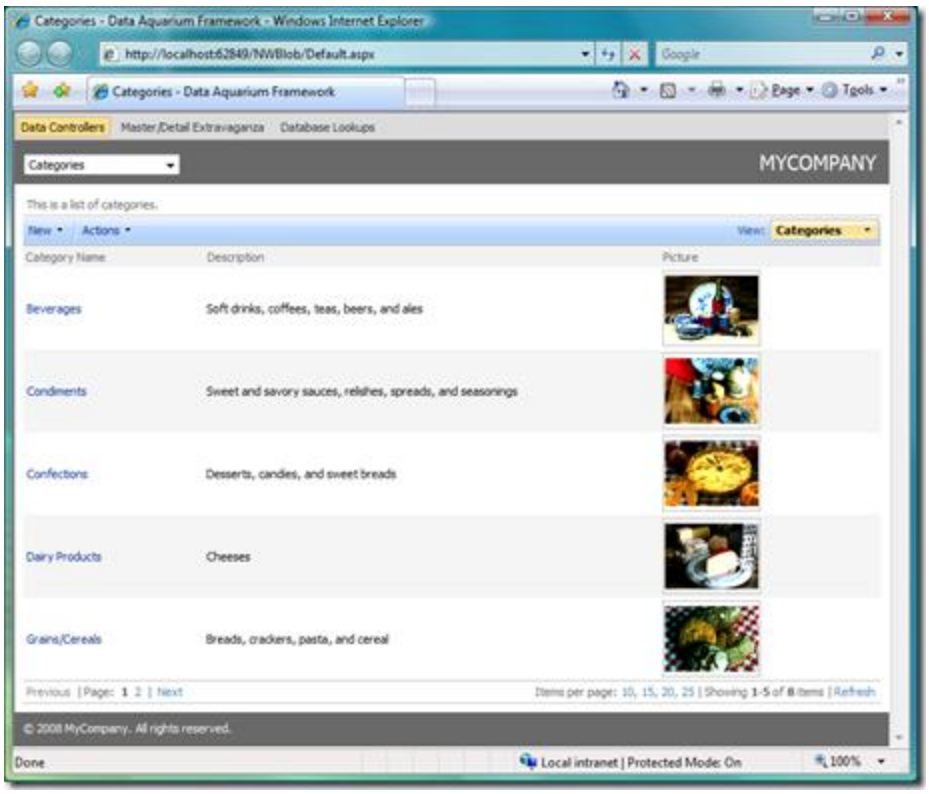
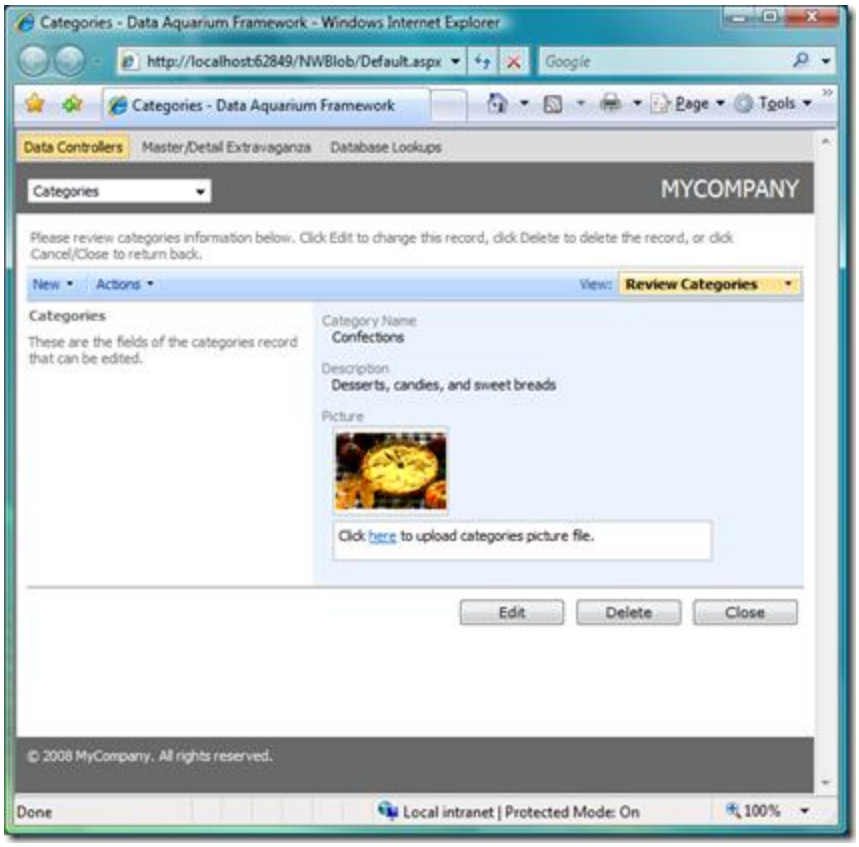
Upload / Download of BLOB in AJAX Applications

Data Aquarium Framework introduces a new feature that allows code-free uploading and downloading of **B**inary **L**arge **O**bjects to/from a database table column in AJAX style. The feature is based on a generic handler *blob.ashx* from *File Upload* premium project.

Here is screen shot from a sample application available at http://dev.codeontime.com/demo/nwblob.

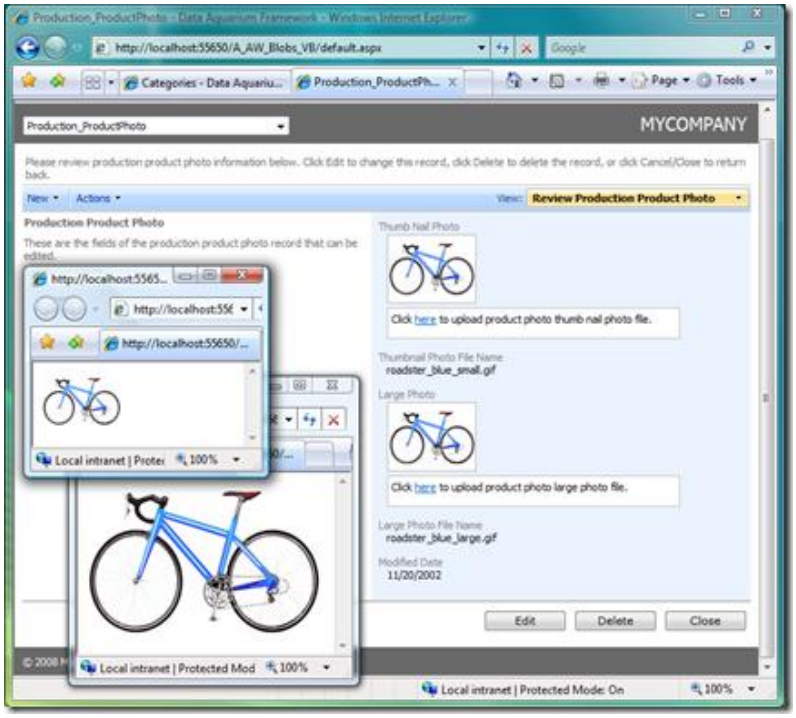


If a BLOB field is storing an image then a thumbnail preview is automatically created. A preview is available in all views. Other types of BLOB are presented by a simple *download* hyperlink. Views are displaying additional controls that allow BLOB uploading when records are presented in edit or detail mode.

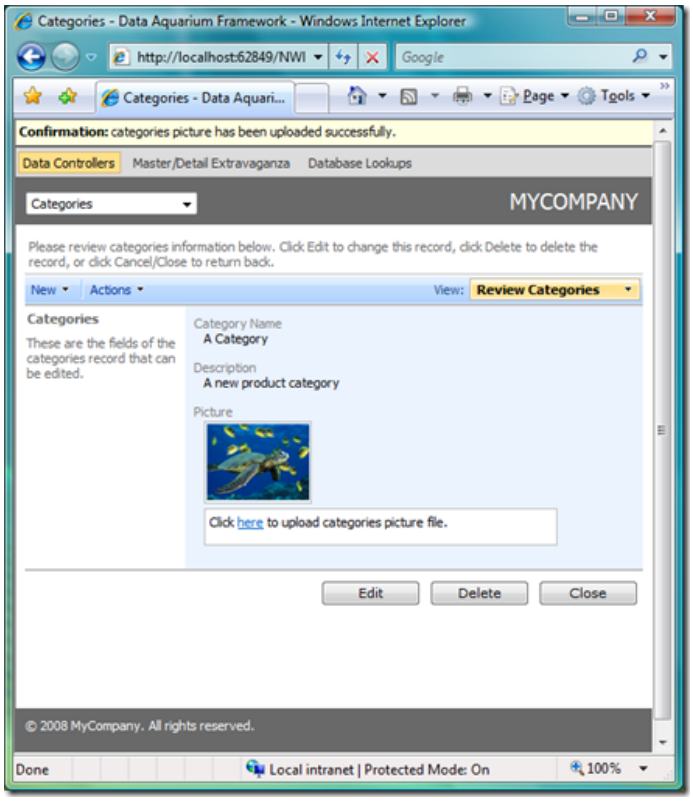


BLOB support is enabled only when you are interacting with existing records. BLOB data cannot be uploaded when creating new records. The internal preview and upload mechanism is based on automatic table row references based on primary keys, which makes uploading of BLOB impossible when records are created.

Original BLOB data can be retrieved from a database with a click of mouse. Here is a screen shot from *Adventure Works* application that shows side-by-side small and large photographs of products. Original BLOB data other then images will be opened by native applications when downloaded completely by your web browser.



A confirmation message is displayed at the top of the page when a BLOB data has been uploaded successfully.



You can find complete implementation details for all of these features in the project source code.

Making It Work

Generate a Data Aquarium Framework application from *Northwind* database and open *~/Contollers/Categories.xml* data controller descriptor. Find definition of *Picture* field. It must look like the one below.

```
<field name="Picture" type="Byte[]" onDemand="true"
  sourceFields="CategoryID"
  onDemandHandler="CategoriesPicture" onDemandStyle="Thumbnail"
  allowQBE="false" allowSorting="false" label="Picture" />
```

Three attributes are turning on the BLOB support. Attribute *sourceFields* must list all fields that allow to find the row with a BLOB. This information is passed on to a generic handler *blob.ashx* located in the root of your web site project. The name of the handler is specified by *onDemandHandler* attribute. Optional style for an on-demand field is controlled by *onDemandStyle* attribute.

Generic handler *blob.ashx* is automatically generated by Code OnTime Generator. All handlers discovered in your database are listed at the top of the file.

Here is how they look in a *Northwind* application written in C#.

```
public partial class BlobFactory
{

    static BlobFactory()
    {
        // register blob handlers
        RegisterHandler("CategoriesPicture", "\"dbo\".\"Categories\"",
"\"Picture\"",
            new string[] {"\"CategoryID\""}, "Categories Picture",
String.Empty);
```

```
        RegisterHandler("EmployeesPhoto", "\"dbo\".\"Employees\"",
"\"Photo\"",
            new string[] {"\"EmployeeID\""}, "Employees Photo",
String.Empty);
    }
}
```

You can quickly change these handlers and add the new ones when needed. The parameters of *RegisterHandler* method are easy to understand.

The first parameter defines a key that is provided as a value for *onDemandHandler* attribute in data controller descriptors. The second parameter is a fully qualified name of the table that stores BLOB values. The third parameter specifies a column name of a BLOB. A list of key columns of this table follows in the fourth parameter. A user-friendly name of the blob in the fifth parameter is for GUI presentation.

The last parameter defines a content type of a BLOB. If the value is an empty string then the code of generic handler will try to automatically find the content type by treating the value as an image. If you are storing a specific type of data other then image then enter a valid content type that matches the BLOB. For example, if you are storing *Microsoft Word* documents then use *application/msword* as content type.