Chestysoft

Website: www.chestysoft.com Email: info@chestysoft.com

Demo to Load, Resize and Upload an image in an HTML Page Using Javascript - Saving Using ASP.NET

REQUIREMENTS

The trial version of csXImage (csXImageTrial.ocx) must either be installed on the client machine, or copied to the same location on the server as the file ResizeUploadASPNET.htm together with the licence package file (csXImageTrial.lpk). These files are available as a free download from https://www.chestysoft.com/ximage/download.asp

The script filesave.aspx must be placed on an ASP.NET enabled web site.

The client browser must be Internet Explorer and it must not have security in place preventing the use of ActiveX controls.

DESCRIPTION

This demo consists of two pages. There is a web page that runs in the client browser which uses the csXImage control to open and resize an image. It can be uploaded to a remote server as an HTTP upload using the PostImage function. This is equivalent to using the <input type=file> HTML tag except the image is sent directly from the control and it does not need to be saved first and manually selected by the user. The second page is an example script that could be used on the remote server to save the file. It is ASP.NET, using Visual Basic. There is also a button for saving the image back to the client file system.

The buttons in the application are enabled and disabled as appropriate. First the image must be opened and that enables the other buttons. This prevents any problems caused by trying to resize, upload or save an empty image.

MAKING THE HTML PAGE WORK

The HTML page, ResizeUploadASPNET.htm, must be edited before use. The PostImage function is on line 78 inside the Javascript function, UploadClick(). The first parameter must be set to the URL of the remote script that saves the upload. It must begin with "http://" and cannot be a relative path.

The second parameter of PostImage is the file name which is used by the "filesave.aspx" script as the name used when the uploaded file is saved. This file name is extracted from the local path when the file was opened.

The third parameter is not used with "filesave.aspx" because there is only one uploaded file and the index can identify it. This is the name that would be given to the form variable containing the file. In an equivalent web form it would be the name inside the tag <input type=file name=_____>. Some cgi scripts and components use this name to identify the file when it is saved.

The final parameter determines the file format to use. Refer to the csXImage instructions for a full list of available formats and their numeric values. In this example the value is found by extracting the extension of the original file and setting it to a number in the switch statement in the OpenClick function. It defaults to 2 for a jpeg if the extension was not one of the standard extensions used by csXImage.

If different resize options are required, the ResizeClick function must be changed to suit. We have shown three sizes which can be selected by radio buttons and these are used with the ResizeImage method.

THE OBJECT TAGS

The csXImage trial control is called using the following:

```
<object classid="clsid:5220cb21-c88d-11cf-b347-00aa00a28331">
  <param name="LPKPath" value="csximagetrial.lpk" />
</object>
<object id="csxi" classid="CLSID:D7EC6EC3-1CDF-11D7-8344-00C1261173F0"
codebase="csximagetrial.ocx" width="800" height="600"></object>
```

The full version would be called with a different class ID. classid="CLSID:62E57FC5-1CCD-11D7-8344-00C1261173F0"

This example uses the codebase attribute to allow downloading of the OCX file but will also work on a client that has the OCX registered and licensed. For more details on running csXImage in a browser and issues relating to registration and licensing, see:

https://www.chestysoft.com/ximage/clientside.asp

TROUBLESHOOTING

If the control fails to run when ResizeUploadASPNET.htm is loaded into the browser, check that the trial version of csXImage is on the client machine. If it has been installed using the executable installer it should be registered and the licence file should be included. Check the security settings in Internet Explorer to make sure that ActiveX controls are allowed to run.

When the file is uploaded to the server a message will be displayed either saying "Image Uploaded" or "Upload Failed". The "Image Uploaded" message is shown if the remote script returns a 200 OK message, which is not proof that the file was saved although the filesave.aspx script provided with the example will have saved the file if this message is seen. If the "Upload Failed" message appears check that the URL in the PostImage command is valid, as described above. There are several reasons why the remote script may fail and some of these are beyond the scope of this example. Check that the directory containing the script allows the ASPNET machine account permission to write.

When problems are encountered the remote script should be checked using a simple HTML form, because then the results can be seen. If the server restricts uploads above a certain size this could lead to upload failures that do not have an obvious cause. When testing with your simple HTML form, use files of a comparable size.

If a different method is used to save the uploaded file it might be necessary to change the PostImage parameters to include the third parameter, the name of the form variable containing the file, as described earlier.

Chestysoft, August 2019.
Website: www.chestysoft.com