Cold Fusion Upload Script

This article provides a sample script, written using ColdFusion, to upload images via the HTTP POST method. Instructions on how it can be tailored for use in your Web applications are also included. You will need to set this facility if you would like to be able to upload local images to the server.

> ⓘ Download the Cold Fusion multimedia upload example code: coldfusion_postacceptor.cfm.
> The source for EditLive multimedia upload scripts can also be found in the SDK_INSTALL/webfolder/uploadscripts/ directory where SDK_INSTALL represents the directory to where the EditLive! SDK is installed.

Defining the Location of the Image Upload Handler Script

The location of the image upload handler script must be defined within the EditLive! configuration file. This setting is configured via the **href** attribute of the <httpUpload> element of the configuration file. To use this example script, the **href** attribute should point to the location of this script on the server.

Defining the Location of the Image Upload Directory

This example script uploads images to the directory specified by the *imageFolder* variable. In order for images to function correctly within EditLive!, the base attribute of the <httpUpload> element must reflect the location of the directory where images on the Web server.

Creating an Image Upload Handler Script

1. Use the <CFSETTING> tag to prevent extra whitespace from being written to the output. EditLive! uses all output returned for the saved filename; extra whitespace will cause image URLs to be incorrect. For this reason the <CFSETTING> tag must be the first thing in the document.

```
<CFSETTING EnableCFOutputOnly="Yes">
<!--- The above ensures only the <cfoutput> tag causes any output;
     EditLive! includes extra whitespace as part of the file name.
--->
```

2. Create a variable to store the location of the images on the server. In this example, we are using the *images* subfolder of the script location.

```
<!---*********************************************
     * Change this line to set the upload folder *
     *********************************************--->
<cfset uploadFolder="images">
```

3. Use the *expandPath* function to get the full path of the folder on the server.

```
<cfset imageFolder=expandPath("#uploadFolder#")>
```

4. Save the uploaded file to the *images* folder.

```
<!--- Save the uploaded file --->
<cffile action = "upload"
    fileField = "image"
    destination = "#imageFolder#"
    nameConflict = "MakeUnique">
```

5. Return the name of the file to ELJ in case the MakeUnique script directive caused the file to be renamed.

```
<!--- Notify ELJ of the filename --->
<cfoutput>#serverFile#</cfoutput>
```

Example Image Upload Handler Script

Tiny has written a sample image upload handler script using ColdFusion. This script can be found at *SDK_INSTALL\webfolder\uploadscripts\coldfusion\coldfusion_postacceptor.chm*, where *SDK_INSTALL* represents the location where the EditLive!SDK is installed.

For image upload, one line of code in the *coldfusion_postacceptor.chm* file must be changed. This line of code specifies the location where you wish image files to be uploaded to. If the location of the upload acceptor script was *http://www.yourserver.com/scripts/coldfusion_postacceptor.chm*, then setting the *imageDir* variable to *../images* would upload the images to a directory with the URL *http://www.yourserver.com/images/*.

```
$imageFolder = "images/";
```

> ⓘ Relative paths specified within the image upload acceptor script are relative to the Web accessible location of the image upload acceptor script.

Integrating the Upload Script

1. Open your configuration file in any text editor (e.g. Notepad on Windows).
2. Locate the <mediaSettings> element and add a <httpUpload> element to it. Only a portion of the full <mediaSettings> element is shown here.

```
<mediaSettings>
    <httpUpload
        base="images/"
        href="coldfusion_postacceptor.chm">
    </httpUpload>
    ...
</mediaSettings>
```

See Also

- HTTP Upload Support for Images and Objects