HTTP Upload Support for Images and Objects

Overview

When inserting and uploading local images and objects (multimedia files) to a remote server, Ephox EditLive! uses the HTTP multipart form-data protocol.

The <httpUpload> configuration file element contains the configuration information for the HTTP Upload functionality for images and media embedded via <object> tags.

Why Use HTTP?

Using the HTTP POST method to insert and upload images and objects offers a secure way of allowing end users to interact with the remote server that the images and objects are to be stored on.

Requirements for HTTP Upload

In order to upload local files to the remote server via HTTP, you will need a server-side upload handler script that accepts the images and objects on the server and stores them in the correct directory or database. This script is the same script that would be used for uploading any file to the server via the HTTP POST method.

For example, when you use a file input element (i.e. <INPUT type="file">), the script specified in the action attribute of the parent <form> element is used to upload the file to the server.

```
<html>
   <body>
      ...
      <form method="POST" action="http://www.yourserver.com/scripts/upload.jsp">
         <input type="file" name="imageSelection">
      </form>
      ...
   </body>
</html>
```

This script would be the same script specified through EditLive! to upload images to the remote server.

EditLive! HTTP Upload Configuration

The EditLive! HTTP upload configuration requires a URL corresponding to the image upload handler script as well as a base property that will be used to replace all existing local image and object URLs after the upload. These properties can be set via the **href** and **base** attributes of the <httpUpload> element respectively.

Example
For the following configuration file example, the image upload script is located at *http://www.yourserver.com/scripts/upload.jsp*. The uploaded files can be found in a directory with the Absolute URL *http://www.yourserver.com/userfiles/*. (i.e. all local images and objects URLs will be replaced with the URL *http://www.yourserver.com/userfiles/*.)

```
<editLive>
   ...
   <mediaSettings>
      ...
      <httpUpload
         base="http://www.yourserver.com/userfiles/"
         href="http://www.yourserver.com/scripts/upload.jsp" />
      ...
   </mediaSettings>
   ...
</editLive>
```

Example
For the following example, the URL of the directory containing pages with EditLive! embedded in them is *http://yourserver.com/editlive/pages*. The URL of the directory containing images is *http://yourserver.com/editlive/repository/images*. Hence, the relative base URL for displaying images in EditLive! would be *../repository/images*.

The upload handler script is located at *http://yourserver.com/scripts/uploadhander.asp*

```
<editLive>
   ...
   <mediaSettings>
      ...
      <httpUpload
         base="../repository/images"
         href="http://yourserver.com/scripts/uploadhandler.asp" />
      ...
   </mediaSettings>
   ...
</editLive>
```

Hence, if an image *myImage.jpg* is located at *http://yourserver.com/editlive/repository/images* and a HTML page located at *http://yourserver.com/editlive/pages/myPage.html* wants to reference this image, the URL would be *../repository/images/myImage.jpg*. With the above configuration file settings, this path would be correct.

Ephox recommends using absolute URLs when possible, as it is less likely to lead to confusion.

When are Files Actually Uploaded?

Images and embedded object files are uploaded (via your upload script) in the following circumstances:

- EditLive! is embedded in a HTML <form>, which is then submitted via a submit button.
- The uploadImages Method of EditLive! is invoked.
- The getBody Method of EditLive! is invoked, passing true as the second parameter.
- The getDocument Method of EditLive! is invoked, passing true as the second parameter.

When relying on images being uploaded when the containing <form> is submitted then the setAutoSubmit Method must **not** be set to false if local images are to be uploaded.

Example

```
<form name="form1" method="POST" action="http://www.yourserver.com/scripts/upload.jsp">
   <script src="../../redistributables/editlivejava/editlivejava.js"></script>

   <script language="JavaScript">
      var editlivejava1;
      editlivejava1 = new EditLiveJava("ELApplet1", "700", "400");
      ...
      editlivejava1.show();
   </script>
   <input type="submit" value="Submit"/>
</form>
```

What Form Fields are Files Uploaded In?

Local files which are uploaded to the server by EditLive! are placed within a form field with the name image, or to the field specified with the "uploadFileFieldName" attribute. When implementing an upload acceptor script specifically to receive files from EditLive! the script should be made to accept files submitted within the image field.

Dynamically Setting the Image or Object URL

If you don't want every local image specified to have its URL changed to the **base** attribute specified, it is possible to dynamically assign the URL for images or objects embedded within the content as they are uploaded. This functionality is achieved via the POST acceptor for files. If the POST acceptor returns a string containing a URL, the URL is inserted into the document source code as the URL for the file. In this way it is possible to dynamically assign the directory files are uploaded to without having to dynamically generate the configuration file.

For this functionality to operate correctly the relevant upload acceptor script must only return a single line string with the URL corresponding to the location for the uploaded file.

When setting the file URL in this manner the URL returned by the POST acceptor script to EditLive! takes precedence over the **base** attribute of the <httpUpload> element in the EditLive! configuration file.

For examples of using a POST acceptor script to return a URL to an instance of EditLive!, see the ASP.NET, ASP, Cold Fusion, JSP, and PHP example Image Upload Scripts packaged with this SDK.

⚠ The URL returned by the POST acceptor must be exactly the URL to be used for the relevant file in EditLive!.

⚠ Ensure that the URL returned by the POST acceptor does **not** include a newline character. Note that newline characters can be introduced by using a method such as println or writeln. Ensure you are using methods which do **not** introduce a newline character such as print or write.

Example Image Upload Scripts

EditLive! for Java is packaged with sample upload scripts for ASP, ASP.NET, JSP, PHP and ColdFusion that enable uploads for common image formats. The source for these scripts can be found in the *SDK_INSTALL/webfolder/uploadscripts/* directory where SDK_INSTALL represents the directory to where the EditLive! SDK is installed. The upload scripts can also be extended to be used with files associated with embedded objects (e.g. Macromedia Flash files). This is achieved by adding the relevant extensions to the list of known file types.

Documentation is also available for the upload acceptor scripts:

- ASP Upload Script
- ASP.NET Upload Script
- JSP Upload Script
- PHP Upload Script
- Cold Fusion Upload Script

Common Problems

There are a number of problems that may occur while attempting to use HTTP Upload. These can be complicated and vary from server to server. For more information on the settings relevant for your server, please consult the documentation for your server. Some common problems include:

- Server side settings - ensure that the HTTP upload script exists in a directory in which scripts can be executed.
- File system permissions - ensure that the file permissions of the directory in which files are to be placed has write and read permissions set.

See Also

- <httpUpload> Configuration File Element
- Example ASP Upload Script
- Example ASP.NET Upload Script
- Example JSP Upload Script
- Example PHP Upload Script
- Example Cold Fusion Upload Script