FlexUpload

Table of contents

1. Introduction

1.1. What does it do?

FlexUpload is an Adobe® Flex™ Application providing a Flash® gui to upload multiple files via HTTP at once and show nice information about the upload progress. It can be integrated in every web-application which often needs to upload multiple files e.g. image galleries.

1.2. Requirements

Client:

Web browser with installed Adobe Flash Player Version 9+

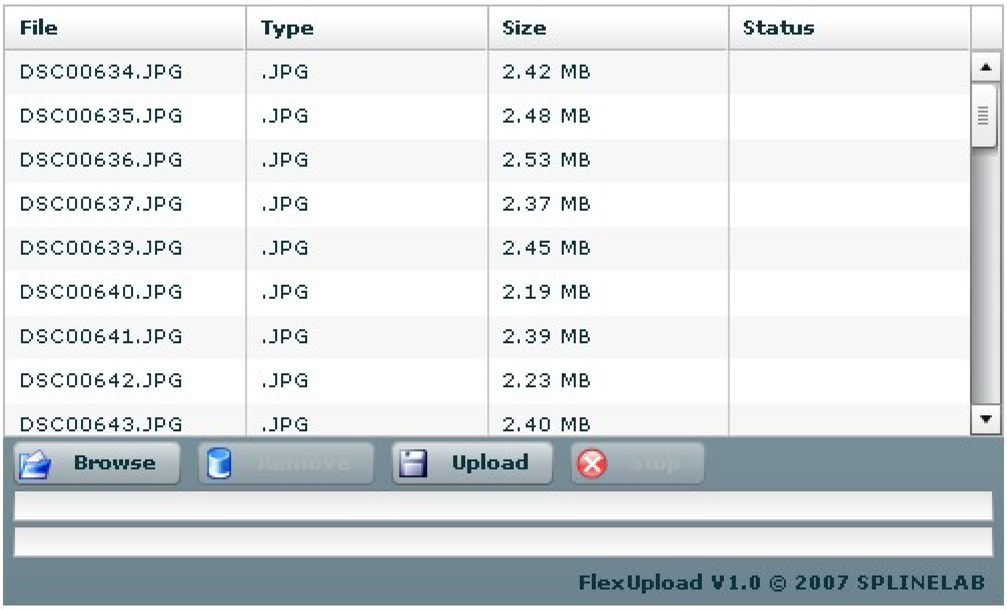
Server:

Any program or script that can handle HTTP POST requests as all common web-scripting languages like PHP, Perl, ASP, JSP can do.

2. User Manual

Even though usage of FlexUpload is really simple here are some notes:

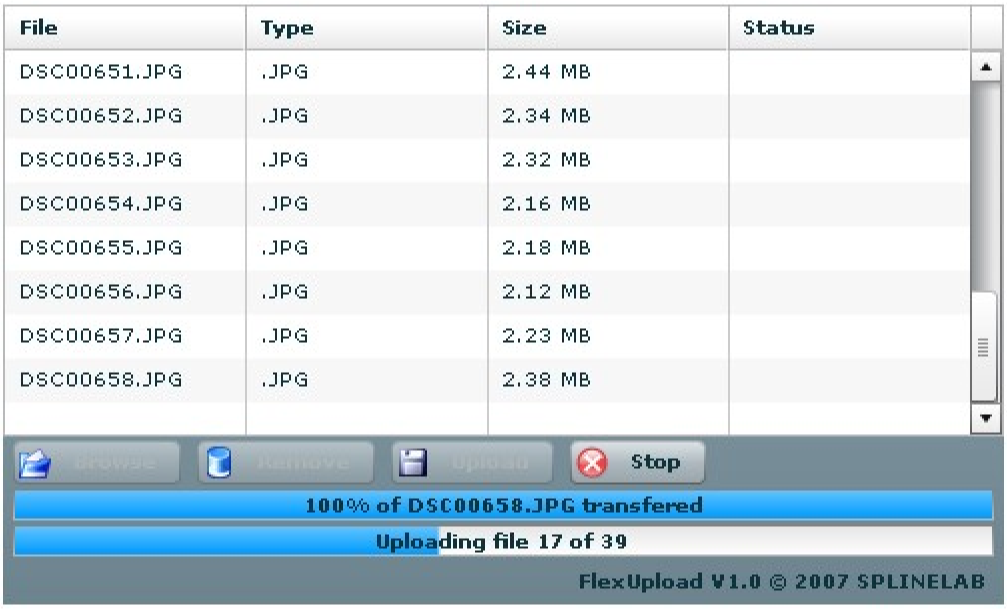
FlexUpload consists of 5 controls; a ListView and 4 buttons.



File	Type	Size	Status
DSC00634.JPG	.JPG	2.42 MB	
DSC00635.JPG	.JPG	2.48 MB	
DSC00636.JPG	.JPG	2.53 MB	
DSC00637.JPG	.JPG	2.37 MB	
DSC00639.JPG	.JPG	2.45 MB	
DSC00640.JPG	.JPG	2.19 MB	
DSC00641.JPG	.JPG	2.39 MB	
DSC00642.JPG	.JPG	2.23 MB	
DSC00643.JPG	.JPG	2.40 MB	

The ListView at the top shows you information like type and size of the files you selected to upload and also a “status” column which will show you a message if something went wrong with this file – e.g. file size is too big or the upload failed.

Use the “Browse” button to select the files you want to transfer to the server. Depending on configuration the maximum number of files, the maximum size per file, and the file types you can upload may differ. Ask your administrator or developer about any limitations. You can select the files in the ListView and click the “Remove” button – as you may suspect – to remove the selected files from the list. If you are satisfied with your choice hit the “Upload” button to start transferring files to the server. Use the “Stop” button to stop the upload. While uploading the files two progress bars at the bottom of FlexUpload will show nice information about what is going on.



File	Type	Size	Status
DSC00651.JPG	.JPG	2.44 MB	
DSC00652.JPG	.JPG	2.34 MB	
DSC00653.JPG	.JPG	2.32 MB	
DSC00654.JPG	.JPG	2.16 MB	
DSC00655.JPG	.JPG	2.18 MB	
DSC00656.JPG	.JPG	2.12 MB	
DSC00657.JPG	.JPG	2.23 MB	
DSC00658.JPG	.JPG	2.38 MB	

100% of DSC00658.JPG transfered

Uploading file 17 of 39

FlexUpload V1.0 © 2007 SPLINELAB

Successfully uploaded files are automatically removed from the list. Once the upload process has finished and there are still files in the list the “Status” column will provide information about the problem. Depending on the error resp., all errors but “file too big” you can try to upload the files again by clicking the “Upload” button.

3. Developers Manual

3.1. Overview of implementing FlexUpload

You can imagine the process of implementing FlexUpload like normal file uploads in HTML forms using the `<input type="file" name="Filedata" />` tag. Refer to the documentation of your programming language to get information on how to handle file uploads.

All examples in this documentation and shipped with FlexUpload are written in PHP but it should be easy to translate them into your favorite programming language. If you would like to publish examples in other languages than PHP and see your credits here don't hesitate to contact me.

3.2. Receiving files on the server

As described above FlexUpload sends the files exactly like “multipart/form-data“ encoded HTML forms. The name of the field used to transfer the data is “Filedata”. The files are transferred one by one, so your script is called multiple times if the user uploads multiple files and “Filedata” contains one file at a time and not an array of files.

After receiving the file and processing it for your needs you have to return a single line of text to inform FlexUpload about the success of your operation. If upload and all operations succeeded just return “OK”. In case of an error return a string describing the error.

Example:

```
...
$newFilePathAndName = ...
if (is_uploaded_file($_FILES['Filedata']['tmp_name'])) {
       if (move_uploaded_file($_FILES['Filedata']['tmp_name'], $newFilePathAndName)) {
              echo 'OK';
       } else {
              echo 'can not move the uploaded file!';
       }
} else {
       echo 'something went wrong!';
}
```

Have a look at “upload_example.php” for a more detailed example.

3.3. Displaying and configuring FlexUpload

You can display FlexUpload in your application like any other Flash movie using the `<object>` and `<embed>` tags. You also can use a wrapper like SWFObject to include FlexUpload. You have to pass some parameters in order to configure FlexUpload. Parameters are either passed as a query string to flexupload.swf:

```
<object ... >
       <param name='src' value='flexupload.swf?option1=value&option2=value'>
       ...
       <embed src='flexupload.swf?option1=value&option2=value' ... />
</object>
```

or the “flashVars” property in the `<object>` and <embed> tags:

```
<object ...>
       <param name='src' value='flexupload.swf'/>
       <param name='flashVars' value='option1=value&option2=value'/>
       ...
      <embed src='flexupload.swf' flashVars='option1=value&option2=value' ... />
</object>
```

In both cases the values have to be url encoded.

If you use PHP to display FlexUpload you can use the “FlexUpload PHP Class” to configure and print flexupload.swf. Usage is really simple and you can display FlexUpload using SWFObject as the wrapper with only 2 lines of PHP code:

```
$fup = new FlexUpload('your_upload_script.php');
$fup->printHTML();
```

Have a look at “The FlexUpload PHP class” for more information about that.

3.4. FlexUpload's parameters

Option	Description	Example	Default
postURL	The URL of your upload script	"pathto/myupload.php"	""
maxFileSize	The maximum size of a file in bytes. FlexUpload does not upload files bigger than this size.		2097152
maxFiles	The maximum number of files the user can transfer at once, or -1 for no limit		100
fileExtensions	Allow file extensions separated by semicolon or an empty string to allow all files	"" -> all files "*.txt" -> only .txt files "*.png;*.jpg" -> .png and .jpg files	"*.gif;*.jpg;*.jpeg;*.png"
locale	the language file. See “Localization” for details about language files or an empty String for the default language (english)	"http://www.example.com/flexupload/locale/de.xml"	""

3.5. The FlexUpload PHP class

The FlexUpload PHP class will help you to implement FlexUpload in your own applications. It uses SWFObject by default to display the Flash movie so you don't have to care about Flash detection and the user sees a message if his Flash Plugin is below version 9 or no Flash Player is installed at all.

All you have to do is to make an instance of the class, set some options and print it to the client:

```
<?php
require_once('class.flexupload.inc.php');
$fup = new FlexUpload('your_upload_script.php');
$fup->printHTML();
?>
```

Have a look at the “FlexUpload PHP Class Documentation” for details about methods and properties of the class.

3.6. Localization

Localization of FlexUpload is handled by language files. Language files are XML files containing the strings used in FlexUpload. You can use “locale/en.xml” as a starting point for your own translations. Note that the file has to be UTF-8 encoded. To use your language file just set the “locale” parameter of FlexUpload to your file.

If you did a translation into a language not shipped with FlexUpload and you want to publish it, don't hesitate to contact me.

3.7. Migration from JavaUpload

3.7.1. Why should I migrate

A few reasons to migrate:

- Most users reported they don't have the latest Java Plugin installed but they regular update Flash Player because many websites require it
- Some users reported that the applet doesn't show up in IE 7 under Windows Vista even if the latest Java Plugin is installed
- The Applet doesn't work if the user's connection is established over a proxy server
- We stop any (noncommercial) support for JavaUpload since FlexUpload is released
- Translations are now stored in external XML files and not in property lists compiled into the application so you can provide your own language files
- FlexUpload is more error resistant
- FlexUpload is loaded and initialized much faster
- FlexUpload is free software

3.7.2. Overview

Migration from JavaUpload to FlexUpload is quite simple. Just replace the code to display the Applet with the code to display the Flash movie resp. in PHP use the FlexUpload class instead of the JavaUpload class. You also have to make some minor changes to your upload script but in most cases migration can be done in just a few minutes. Please have a look at the examples to become familiar with these changes.

3.7.3. Migrating the Frontend

FlexUpload uses a different approach to handle localizations. Translations are now stored in external XML files and not in property lists compiled into the application. This is a great advantage because it's much easier to maintain the language files. To specify the locale you now have to set the name of the language file including the path to it!
e.g. in JavaUpload you wrote

```
$jup = new JavaUpload();
$jup->setLocale("de_DE");
```

in FlexUpload you write now

```
$fup = new FlexUpload();
$fup->setLocale('locale/de.xml');
```

3.7.4. Migrating the upload script

The field name has changed to "Filedata" (this is the default in Flex) so you have to use
$_FILES['Filedata'] to access the uploaded file.
The format of the return value also has changed. If you print something different than "OK"

FlexUpload assumes an error.
E.g. In JavaUpload you printed "`success=1\r\n`" if your upload succeeds and "`success=0\r\nSome error occurred`" if something went wrong.
Now you just print "OK" if your upload succeeds and "Some error occured". This helps to avoid parsing errors of the return value if for example your script raises some PHP warnings or errors.

4. Licenses and Credits

4.1. FlexUpload license

FlexUpload - upload multiple files at once

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

The GNU General Public License can be found at
http://www.gnu.org/copyleft/gpl.html.
A copy is found in the text-file GPL and important notices to the license from the author are found in the LICENSE file.

This copyright notice MUST APPEAR in all copies of the program!

4.2. Third-Party Licenses used in FlexUpload

SWFObject v1.4.4: Flash Player detection and embed – http://blog.deconcept.com/swfobject/

> SWFObject is © 2006 Geoff Stearns and is released under the MIT License:
> http://www.opensource.org/licenses/mit-license.php
> SWFObject is the SWF embed script formerly known as FlashObject. The name was changed for legal reasons.

4.3. Thanks

Astrid Lohse for proof-reading.