

inaflash

FLASH UPLOAD UTILITY

Documentation v1.0
Last Update: 9/23/2008

<http://www.inaflashuploader.com>

Introduction

In-a-Flash Flash Upload Utility is a free Flash application that works in conjunction with a PHP class which gives users an easy way to upload files and view their upload progress.

This documentation will show you how you can add the In-a-Flash Flash Upload Utility to your site and customize it with just a few lines of code. You can even use PHP to tailor the way the server handles your upload once it has been completed.

Getting Started

To use In-a-Flash, you simply need to have access to a server running PHP 4.0+. Site users will need to be using Flash 8.0+ (over 98% of web users have this). It is also important to note that, by default, most servers are setup to only allow a maximum file upload of 2MB. This may work for you, and it may not. If not you may have to change this configuration in both the PHP and your server's configuration files.

Also, your users will need version 8 or higher of the Flash player to use In-a-Flash Uploader. Flash limits file uploads to 100MB per file.

Need to update your PHP file upload configuration?

Method #1 (php.ini):

http://www.developershome.com/wap/wapUpload/wap_upload.asp?page=php2

Method #2 (.htaccess):

<http://sandalian.com/46/php/increasing-file-upload-limit-using-htaccess.htm>

Need to update your server's file upload configuration? Well, it's different for every server so just Google it!

<http://www.google.com>

Installation

To install In-a-Flash, just download the necessary files at <http://www.inaflashuploader.com>. The download is a .zip file. Unzip this file. It should contain a folder movie called "uploader".

Just copy this folder into the main directory of your site/application on your server and the installation is complete.

Now you just need to add this code where you want the In-a-Flash Uploader placed on a PHP page:

```
<?php
    require_once 'uploader/class.FlashUploader.php';
    IAF_display_js();
    $uploader = new FlashUploader('uploader', 'uploader/uploader',
'YOUR_SITE_URL_GOES_HERE/uploader/upload.php');

    //CUSTOMIZATION CODE GOES HERE!

    $uploader->display();
?>
```

The first parameter in the FlashUploader constructor is the ID of the div that will contain the uploader (you can change this to suit your needs, but it is not necessary that you do it).

In-a-Flash is setup by default to upload the files into a folder called 'dropbox' in the root directory of your site. You'll need to create a directory with write permissions called 'dropbox' in the root directory of your site (you can also change this directory by personalizing your 'uploader/upload.php' page). Right now it is just setup to copy the uploaded file into the dropbox folder. You can add your own code to resize images, move files to different folders, and do countless other things.

NOTE: *If you are having problems uploading files try using this code:*

```
$uploader = new FlashUploader('uploader', 'uploader/uploader',
'YOUR_SITE_URL_GOES_HERE/uploader/upload-debug.php');
```

- instead of the code above. The upload-debug.php file will print all passed variables (sent using pass_var) and all uploaded file information to a file called output.txt (you can change this below) in order to debug the utility. If nothing is printed to output.txt it is either a permission problem (make the folder that contains upload-debug.php writeable) or your path to the upload-debug.php file (the 3rd paramater of the \$uploader->create() function) is wrong. Make sure you use a full web path if you are having problems (such as <http://www.inaflashuploader.com/uploader/upload.php>)

Now you are ready to customize your In-a-Flash uploader (put all of the customization code where it says "//CUSTOMIZATION CODE GOES HERE!").

NOTE: Please make sure that any folders you are trying to upload files into possess the proper permissions to do so.

Can't open the .zip file?

WinRAR - <http://www.rarlab.com>

WinZIP – <http://www.winzip.com>

Stuffit (for Mac) = <http://www.stuffit.com>

Customization

You can customize the In-a-Flash Uploader by using the set function of the In-a-Flash Uploader.

set function prototype:

```
$uploader->set(property, value);
```

style

```
$uploader->set('style', 'bar');
```

The style property sets the style of the In-a-Flash Uploader. There are currently two styles available in the free version – bar and text.

The default style is bar (if you want the bar style then you don't have to use this code).

User these figures for the references below:



bg_color

```
$uploader->set('bgcolor', '0x000000');
```

The bg_color property sets the background color of the In-a-Flash Uploader (in this case, to #000000 or black). Just enter "0x" then hex code of your desired color.

The default background color is white (#FFFFFF) (if you want the background color to be white then you don't have to use this code).

bar bg color

```
$uploader->set('bar_bg_color', '0xF0E999');
```

The `bar_bg_color` property sets the background color of the In-a-Flash Uploader's progress bar (in this case, to #F0E999 or a mellow yellow). Just enter "0x" then hex code of your desired color.

The default bar background color is a bright blue (#6CC2F2) (if you want the bar background color to be this shade of blue then you don't have to use this code). Also, this code will have no effect if you are using the text style.

divider color

```
$uploader->set('divider_color', '0xF0E999');
```

The `divider_color` property sets the color of the In-a-Flash uploader's dividers (in this case, to #F0E999 or a mellow yellow). Just enter "0x" then hex code of your desired color.

The default divider color is a very light blue (#E9F6FD) (if you want the divider color to be light blue then you don't have to use this code).

txt title color

```
$uploader->set('txt_title_color', '0x000000');
```

The `txt_title_color` property sets the color of the In-a-Flash uploader's title text (in this case, to #000000 or black). Just enter "0x" then hex code of your desired color.

The default title text color is black (#000000) (if you want the title text color to be black then you don't have to use this code).

txt filename color

```
$uploader->set('txt_filename_color', '0x8A2A25');
```

The `txt_filename_color` property sets the color of the In-a-Flash uploader's filename text (in this case, to #8A2A25 or a sorta maroonish color). Just enter "0x" then hex code of your desired color.

The default title text color is a medium gray (#383838) (if you want the filename text color to be gray then you don't have to use this code).

txt_percent_color

```
$uploader->set('txt_percent_color', '0x8A2A25');
```

The `txt_percent_color` property sets the color of the In-a-Flash uploader's percent text (in this case, to #8A2A25 or a sorta maroonish color). Just enter "0x" then hex code of your desired color.

The default percent text color is a medium gray (#383838) (if you want the filename text color to be gray then you don't have to use this code).

txt_progress_color

```
$uploader->set('txt_progress_color', '0x8A2A25');
```

The `txt_progress_color` property sets the color of the In-a-Flash uploader's progress message text (in this case, to #8A2A25 or a sorta maroonish color). Just enter "0x" then hex code of your desired color.

The default progress text color is a medium gray (#383838) (if you want the progress text color to be gray then you don't have to use this code).

button_color

```
$uploader->set('button_color', '0x8A2A25');
```

The `txt_progress_color` property sets the color of the In-a-Flash uploader's progress message text (in this case, to #8A2A25 or a sorta maroonish color). Just enter "0x" then hex code of your desired color.

The default button color is a light blue (#E9F6FD) (if you want the button color to be this shade of blue then you don't have to use this code).

button_shadow

```
$uploader->set('button_shadow', '0x7A2A25');
```

The `txt_progress_color` property sets the color of the In-a-Flash uploader's progress message text (in this case, to #7A2A25 or a slightly darker maroonish color). Just enter "0x" then hex code of your desired color.

The default button shadow color is a slightly darker light blue (#c9e7f7) (if you want the progress text color to be this shade of blue then you don't have to use this code).

button title color

```
$uploader->set('button_title_color', '0xFF0000');
```

The `txt_progress_color` property sets the color of the In-a-Flash uploader's progress message text (in this case, to #FF0000 or really red). Just enter "0x" then hex code of your desired color.

The default button shadow color is black (#000000) (if you want the progress text color to be this shade of black then you don't have to use this code).

max file size

```
$uploader->set('max_file_size', 50);
```

The `max_file_size` property sets the maximum file size (in MB) the In-a-Flash uploader will accept (in this case 50 MB). This is a maximum for individual files not for the entire upload.

The default maximum file size is 100 MB which is the maximum amount that Flash will accept (if you want the maximum file size to be 100 MB then you don't have to use this code).

max files

```
$uploader->set('max_files', 1);
```

The `max_file` property sets the maximum number of files the utility will allow the user to select for upload (in this case 1).

The default maximum files is 0 (which means unlimited - if you want the maximum file size to be 100 MB then you don't have to use this code).

allow clear

```
$uploader->set('allow_clear', '0');
```

The `allow_clear` property dictates whether the In-a-Flash uploader will allow users to clear uploads (remove them from the upload screen) after their completion. A value of '0' will disallow clearing and a value of '1' will allow clearing.

The default value is '1' meaning users can clear uploads (if you want users to be allowed to clear their uploads then you don't have to use this code).

auto clear

```
$uploader->set('auto_clear', '1');
```

The auto_clear property dictates whether the In-a-Flash uploader will automatically clear uploads (remove them from the upload screen) after their completion. A value of '0' will disable automatic clearing and a value of '1' will enable automatic clearing.

The default value is '0' meaning auto clearing is automatically disabled (if you want to automatic clearing disabled then you don't have to use this code).

allow cancel

```
$uploader->set('allow_cancel', '0');
```

The allow_cancel property dictates whether the In-a-Flash uploader will allow users to cancel uploads (thus removing them from the upload screen) while they are uploading. A value of '0' will disallow canceling and a value of '1' will allow canceling.

The default value is '1' meaning users can cancel uploads (if you want users to be able to cancel their uploads then you don't have to use this code).

valid extensions

```
$uploader->set('valid_extensions', '*.jpg,*.gif,*.png');
```

The valid_extensions property dictates what file extensions the In-a-Flash uploader will allow users to upload. The extensions should be in the form *.extension and delimited by commas. *Please be warned that you should check the file types on the upload page to be sure the file extension is correct. Also, you should NEVER allow users to upload files that could do damage to your server (such as PHP, ASP, and other server side scripting pages).*

The default value is '*.jpg' meaning users can only upload jpeg files. This is basically to FORCE you to change this! *Please don't use '*.*' (all files) unless you really know what you are doing.*

set width

```
$uploader->set('set_width', '400');
```

The `set_width` property allows you to set the width the In-a-Flash uploader should be (in pixels). The uploader will dynamically update to fill the entire area.

The default value is '415' (if you want the In-a-Flash uploader to be 415px wide you don't need to use this code).

set height

```
$uploader->set('set_height', '40');
```

The `set_height` property allows you to set the height the In-a-Flash uploader should be (in pixels). The uploader will dynamically update to fill the entire area. The uploader also uses a javascript function to automatically expand to encompass all ongoing uploads.

The default value is '55'. **YOU SHOULD NOT NEED TO CHANGE THIS!**

click text

```
$uploader->set('click_text', 'New click text goes here');
```

The `click_text` property allows you to set the text that is displayed that tells the user to click to start the upload.

The default value is 'Click here to upload files'.

uploading text

```
$uploader->set('uploading_text', 'New uploading text goes here');
```

The `uploading_text` property allows you to set the text that is displayed that tells the user an upload or uploads are occurring.

The default value is 'Uploading'.

complete text

```
$uploader->set('complete_text', 'New complete text goes here');
```

The `complete_text` property allows you to set the text that is displayed that tells the user the upload is complete

The default value is 'Upload Complete'.

pending text

```
$uploader->set('pending_text', 'New pending text goes here');
```

The `pending_text` property allows you to set the text that is displayed that tells the user the upload is pending.

The default value is 'Upload Pending...`'.

max text

```
$uploader->set('max_text', 'New max text goes here');
```

The `max_text` property allows you to set the text that is displayed that tells the user the maximum number of uploads has been reached.

The default value is 'File Maximum Reached!`'.

valid text

```
$uploader->set('valid_text', 'New valid text goes here');
```

The `valid_text` property allows you to set the text that is displayed in the File Selection window that describes the type of files that are allowed to be uploaded.

The default value is 'Valid Text`'.

size failure text

```
$uploader->set('size_failure_text', 'New size failure text goes here');
```

The `size_failure_text` property allows you to set the text that is displayed that tells the user the maximum file size has been exceeded. You can use the token `'MAX_FILE_SIZE'`. This piece of text will be replaced with the value of the `max_file_size` property.

The default value is `'Failed: Size must be <= MAX_FILE_SIZE MB'`.

progress text

```
$uploader->set('progress_text', 'New progress text goes here');
```

The `progress_text` property allows you to set the text that is displayed that tells how much of the file has been uploaded. You can use the tokens `'BYTES_LOADED'` and `'BYTES_TOTAL'`. These pieces of text will be replaced with the value of the current bytes loaded and the total bytes of the file, respectively.

The default value is `'BYTES_LOADED of BYTES_TOTAL'`.

Passing Custom Variables to the Processing Page

Advanced functionality has been built into the In-a-Flash uploader so you can pass variables to the upload processing page through GET variables. This is done with the FlashUploader class's `pass_var` function.

The prototype for this function is:

```
$uploader->pass_var(VARNAME, VALUE);
```

You can pass any variable this way for example:

```
$uploader->pass_var('username', 'pfuploader');
```

This code would pass a GET variable called "username" with the value "pfuploader" to the upload processing page. Variables like this could be used to tell the processing page which folder to place the upload in, or other information on how to handle the file upload.

Notice: The sample upload code uses the variables "filename", "temp_name", "error", and "size." It is best to avoid naming any of your custom variables any of these names (your variables will be overwritten) unless you know enough to customize the PHP code to compensate.

Using a Callback Function

Do you need to call a javascript function after a file is uploaded? You can use In-a-Flash uploader's callback functionality to achieve this little trick.

It is easy to do

```
$uploader->set('callback', 'callback_fnc');
```

This will call the javascript function `callback_fnc`. So you would need this javascript code on your page (or linked to from your page). It returns the parameter filename which is the name of the file that has finished uploading.

```
<script type="text/javascript">
    function callback_fnc(filename) {
        //YOUR CODE GOES HERE!
    }
</script>
```

That is all there is too it. In-a-Flash takes care of the rest. **Remember this function is called after each file is uploaded**, so you may want to limit the # of files uploaded at one time (the `max_files` property) to 1. Look up a few pages to see how to use this property.

...And One More Thing

I hope the In-a-Flash uploader works for you, but if you're having trouble and can't figure something out you can email me at mike@vernerwebstudio.com, and I'll see if I can lend a hand. Also, if you are looking for a custom implementation – I'm your man, just contact me. Or if you have a suggestion or would like to send me millions of dollars or if you are a really, really smoking hot Betty... well, you get the point.