Uploadify v2.1.0

Developed by Ronnie Garcia and Travis Nickels
www.uploadify.com
www.ronniesan.com.

Uploadify 可以做什么？

一个基于 jQuery 上传文件的插件，该插件引用 jQuery,Flash 和后台处理脚本（可以基于不同的语言进行开发脚本），上传文件 WEB 站点服务器上。

该怎么利用 Uploadify

Implementation of the plug-in is very easy and only relies on one JQuery function call to initiate.
Uploadify 用起来非常简单，只要调用 jQuery 的初始化函数即可。

1. 下载最新版本 uploadify:http://www.uploadify.com
2. 解压下载的压缩包到指定目录。
3. 在相应的页面 jQuery 脚本，swfobject script 和 Uploadify 插件。将以下代码段加到<head>中：

```
<script type="text/javascript" src="/path/to/jquery-1.3.2.min.js"></script>
<script type="text/javascript" src="/path/to/swfobject.js"></script>
<script type="text/javascript" src="/path/to/jquery.uploadify.v2.1.0.min.js"></script>
```

4. 在页面<head>中增加$.ready 事件，如下代码段：

```
<script type="text/javascript">
  $(document).ready(function() {
       $('#someID').uploadify({
            'uploader': '/path/to/uploadify.swf',
            'script': '/path/to/uploadify.php',
            'folder': '/path/to/uploads-folder',
            'cancelImg': '/path/to/cancel.png'
       });
  });
</script>
```

当页面装载初始化结束，$.ready 事件将会 input 隐藏，将用按钮 browse 代替。到此，该插件的使用准备工作全部就绪。

可用选项

可选项	备注
uploader	Upload.swf 文件的路径，默认为：uploader.swf.也可以用绝对路径前缀必须为‘/’或者‘http’来引用该文件。
script	用来处理上传文件的后台脚本。默认为:upload.php. 或者用’http’如：http://www.uploadify.com/file_fs/upload.jhtml.
checkScript	用于检查该文件是否已经存在服务器的后台处理脚本。无默认值。
scriptData	上传文件时需要传送到后台的参数。如：’${‘name’:’value’}’。
fileDataName	设置一个名字，在服务器处理程序中根据该名字来取上传文件的数据。默认为 Filedata
method **new**	上传文件的提交方式，可选值：‘get’或‘post’默认为：‘post’
scriptAccess	flash 脚本文件的访问模式，如果在本地测试设置为 always，默认值：sameDomain
floder	上传文件存放的目录
queueId **new**	文件队列的 ID，默认情况下，只有一个文件队列创建在按钮 browse 下面。
queueSizeLimit	文件队列的最大容量。默认为 999.
multi	设置为 true，允许上传多个文件。
auto	设置为 true，允许自动提交。
fileDesc	这个属性值必须设置 fileExt 属性后才有效，用来设置选择文件对话框中的提示文本
fileExt	允许上传文件的扩展名，如：‘*.ext1；*.ext2’.则只允许上传此扩展名的文件。如果使用该选项，则 FileDesc 属性必须使用。
sizeLimit	文件最大长度限制。
simUploadLimit	允许同时上传的个数，默认为 1.
buttonText	按钮的显示文本，默认为＝‘BROWSE’
buttonImg	按钮的引用图片。即图片的路径。
hideButton	设置为 true,可隐藏按钮图片
rollover	设置为 true，鼠标滑动到 browse button 时的动画效果。
width	按钮图片或 flash 文件的宽度。默认为 30
height	按钮图片或 flash 文件的高度。如果 rollover 设置为 true 时，那该高度设为实际文件高度的三分之一。默认为 110.
wmode	
cancelImg	取消按钮的图片路径，默认为“cancel.png”
onInit	当页面装载$.ready 脚本被触发时，该函数会被触发。默认事件处理为隐藏目标元素，将它替换为 flash 文件，紧随其后创建一个文件队列。如果该函数返回 false 其默认行为将不会被执行。绝大多数的功能，都可以访问 html 快速上传文件引用变量 flashElement 以下为原文： A function that triggers when the script is loaded. The default event handler hides the targeted element on the page and replaces it with the flash file, then creates a queue container after it. The default function will not trigger if the value of your custom function returns **false**. For custom functions, you can access the html for the flash file using the variable **flashElement**.

onSelect	每个元素被选择时，该函数将会被触发。默认函数处理为：函数生成一个 6 位唯一的字符串来标识该文件，为该文件创建一个文件队列。如果该函数返回 false，则以上将全部失效。 函数有三个参数： Event：事件对象； queueID：该文件所对应文件队列的ＩＤ。 fileObj：被选中文件的详细信息 ● name-文件名称。 ● size-文件的长度。 ● creationDate-创建日期 ● modificationDate-修改日期 ● type-文件扩展名。以‘.’开头
onSelectOnce	操作被选中的文件时，会触发该函数，无默认实现。 有两个参数： Event：事件对象； Data：包含所选中对应操作详细信息。 ● fileCount-文件队列的文件总数 ● filesSelected-被选中的文件总数 ● filesReplaced-在队列已被替换的文件总数 ● allBytesTotal-在文件队列所有文件的总长度。
onCancel	取消正在上传的文件或者移除在文件队列的文件。默认实现为：移除上传文件列队的文件。如果函数返回 false，不触发该函数。 有四个参数： event：事件对象； queueID：即将删除文件的唯一性标识。 fileObj：被选中文件的详细信息 ● name-文件名称。 ● size-文件的长度。 ● creationDate-创建日期 ● modificationDate-修改日期 ● type-文件扩展名。以‘.’开头 data：文件队列详细信息 ● fileCount-文件总数。 ● allBytesTotal 所有文件的总长度。
onClearQueue	调用 fileUploadClearQueue 将会调用该函数。默认为：清除上传文件队列所有文件。若返回 false 时，则不处理。 有一个参数： event：事件对象。
onQueueFull	如果文件队列达到最大容量时，将触发该函数。默认为：弹出对话框，显示用户队列长度。 有两个参数： ● event：事件对象；

	● queueSizeLimit：队列最大容量。
onError	处理上传文件时发生错误时，此函数将被触发。默认实现为： 有四个参数： event：事件对象； queueID：该文件的ＩＤ fileObj：被选中文件的详细信息 ● name-文件名称。 ● size-文件的长度。 ● creationDate-创建日期 ● modificationDate-修改日期 ● type-文件扩展名。以‘.’开头 errorObj：错误对象 ● type-‘ＨＴＴＰ’，‘ＩＯ’，或‘Ｓecurity’ ● info-错误的描述信息。
onOpen	点击上传时触发，如果 auto 设置为 true 则是选择文件时触发，如果有多个文件上传则遍历整个文件队列 有三个函数： event：事件对象； queueID：该文件的ＩＤ fileObj：被选中文件的详细信息 ● name-文件名称。 ● size-文件的长度。 ● creationDate-创建日期 ● modificationDate-修改日期 ● type-文件扩展名。以‘.’开头
onProgress	上传文件时，将会触发该函数。默认实现为：显示上传文件的进度。函数返回 false 将不会处理。 有三个参数： event：事件对象； queueID：该文件的ＩＤ fileObj：被选中文件的详细信息 ● name-文件名称。 ● size-文件的长度。 ● creationDate-创建日期 ● modificationDate-修改日期 ● type-文件扩展名。以‘.’开头 data：上传文件及队列的详细信息的对象 ● percentage-当前完成上传文件的百分比。 ● bytesLoaded-当前完成上传文件的字节数。 ● allBytesLoaded-当前完成所有上传文件的字节数。 ● speed-当前上传文件的速度 KB/s
onComplete	完成文件上传，该函数被触发。默认实现为：在文件队列中删除该文件。若该函数返回 false,则不处理。

	有五个参数： event：事件对象； queueID：该文件的ＩＤ fileObj：被选中文件的详细信息 ● name-文件名称。 ● size-文件的长度。 ● creationDate-创建日期 ● modificationDate-修改日期 ● type-文件扩展名。以‘.’开头 response：服务器返回信息。 data：文件队列详情 ● fileCount-队列文件总数。 ● speed-上传文件的速度。KB/S
onAllComplete	文件队列所有文件都上传完毕。则触发该函数。无默认实现 有两个参数： event：事件对象； data：所有处理上传文件的详细信息。 ● filesUploaded-被上传文件总数。 ● errors-上传中的错误总数。 ● allBytesLoaded-上传的总字节数。 ● speed-上传速度 KB/S
onCheck	该上传文件在服务器被检测已存在，则会触发该函数。 默认实现为：将弹出一个确认提示框。 有五个参数： event：事件对象； checkScript：检测脚本。或ＵＲＬ fileQueue：文件队列，将以 Key/value　健值对的形式存放，以 queueId 作为主健，而文件名作为 value； folder：上传文件时，所存放的文件夹。 single：如果为 true,则队列只有一个文件。

关联函数

uploadifySettings
changed
	A function used to change options for a fileUpload object.
	Two arguments are available, one is required:
	• **Setting** - The option to change.
	• **Value** - The value to change the option to. If left blank, this function will return the current value of the setting.
	
	The following example will change the upload folder to '/uploads':
	`$('#someID').uploadifySettings('folder','/uploads');`
	The following example will return the simUploadLimit:
	`$(' #someID').uploadifySettings('simUploadLimit');`
	The following options can be changed:
	buttonImg, buttonText, cancelImg, fileDesc, fileExt, width, height, folder, script, scriptData, simUploadLimit, sizeLimit, hideButton
	The following will return the current size of the queue
	`$('#someID').uploadifySettings('queueSize');`
	When using scriptData enter it in {key1 : value1, key2 : value2, …} format.
	`$('#someID').uploadifySettings('scriptData', {'name' : some.val()});`
	If the key already exists it will update it, if it doesn't exist it will add it. You no longer need to re-write every key/value pair. When requesting scriptData it will return the key/value pairs as an object.

uploadifyUpload changed	A function used to begin an upload of a single file or all files in the queue. One argument is optional: **queueID:** The unqiue queue identifier for the file to upload. The following example will upload all files in the queue: `$('#someID').uploadifyUpload();`
uploadifyCancel changed	A function used to remove a file from the queue or stop an upload in progress. One argument is required. **queueID:** The unique queue identifier of the file you wish to cancel. The following example will remove a file from the upload queue: `$('#someID').uploadifyCancel('NFJSHS');`
uploadifyClearQueue changed	A function used to clear all files from the upload queue. The following example clears the file upload queue: `$('#someID').uploadifyClearQueue();`

关于 Uploadify 跨域问题

当 Flex 访问 WebService 服务时，在本地能够正常访问，当部署到 web 容器中发布为 web 服务后，再调用 WebServicIE，此时就会被拒绝访问，这就是 Flash 跨域访问的沙箱问题，

为了解决 Flex 跨域访问 WebService 的问题，可采用如下方案：
首先，跨域访问被拒绝是因为提供服务方没有配置安全策略文件，即 crossdomain.xml，如果你不想用 crossdomain.xml 就要用到代理，即自己写一个后台读取 webservice，然后提供给自己的 flex 应用，因为在 flashplayer 中，要跨域必须要有策略文件。考虑到 flashplayer 升级到 9.124 之后，加强了安全性，之前的 crossdomain.xml 的写法发生了变化，以下就是该文件的完整写法：

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
    <cross-domain-policy>
        <allow-access-from domain="*" />
        <allow-http-request-headers-from domain="*" headers="*"/>
    </cross-domain-policy>
```

表示该服务允许任何外域来访问。
关于 crossdomain.xml 的放置目录问题，有如下解决方案，可放置在：
1） 如果这个目录是容器的根目录，可以通过以下的 url 访问 crossdomain.xml：
http://localhost:8080/crossdomain.xml 。
2） 如果 crossdomain.xml 不是放在根目录下，而是在某个 webapp 下面，在 flex 中就需要在初始化的时候应用

```
Security.loadPolicyFile("http:// localhost:8080/aaa /crossdomain.xml");
```

其中 aaa 为 webapp 的名称
这样，外部 Flex 访问该服务发布的 WebService 时，flashplayer 首先找的就是 crossdomain.xml 文件，若安全机制设置为允许访问，则访问成功。

参考链接：
http://hi.baidu.com/lotusxyhf/blog/item/f1c08a58b5771e88810a1870.html
http://www.imeetyou.net/article.asp?id=294