YUI Library: Uploader

2009-9-8 **v2.8**

Instantiating the Uploader

```
<div id="upldrContainer"
     style="width:200px;height:50px"></div>
<script>
var myUploader = new YAHOO.widget.Uploader(
"upldrContainer", "btnSprite.jpg");
</script>
```

Instantiates a new Uploader object, `myUploader`, which is bound to a div whose id attribute is `'upldrContainer'`. The result will be a Flash button, skinned by *btnSprite.jpg* (an image of four-equal sized button skins stacked vertically: up, hover, down, disabled). If no skin URL is passed, the uploader will render as a transparent button. In both cases, clicking the button invokes the “Browse” dialog.

Constructor

```
YAHOO.widget.Uploader(str html id,
                      str btnSkinURL);
```

Arguments:

(1) **HTML element (string or object):** A reference to an HTML id string or element object binds the Uploader to an existing page element. This parameter is required.

(2) **Button Skin URL (string):** A url of an image consisting of four equal-sized button skins stacked vertically. The top-to-bottom order is: **up, hover, down, disabled.** This parameter is optional.

Limitations

1. The Uploader can only send data to servers in the same security sandbox as the `uploader.swf` file. If `uploader.swf` hosted by `yui.yahooapis.com` is used, then the server must contain a cross-domain permissions file allowing `yui.yahooapis.com` to upload files.
2. The intended behavior of the uploader is not to send any cookies with its requests. As a workaround, we suggest either using a cookieless upload method or appending `document.cookie` to the upload request.
3. When the uploader is rendered as a transparent layer, it does not respond to keyboard. When the uploader is rendered as an image, it receives "Space" and "Enter" key presses as triggers, but only if the focus is on the uploader component itself.
4. The uploader does not support basic authentication.
5. The behavior when working through a proxy server is inconsistent and unreliable. Also, when uploading to HTTPS servers, be aware that Flash does not support self-signed certificates.

Simple Use Case

```
myUploader.setAllowMultipleFiles(true);

myUploader.setFileFilters([{description:"Images", extensions:"*.jpg,
*.gif"}]);

myUploader.addListener("fileSelect", onFileSelect);

function onFileSelect (event:Object) {
   myUploader.uploadAll("YOUR UPLOAD URL");
}
```

Sets the permission to browse for multiple files, and allows the users to select files with either **jpg** or **gif** extension (on Windows, the filtering is suggestive, rather than strict.) The “Browse” dialog with these settings comes up when the uploader control is clicked. When files are selected, they are queued and uploaded to the specified URL using automatic queue management.

Solutions

Track upload progress and log it in the YUI Logger (must be included on the page):

```
myUploader.addListener("uploadProgress", onUploadProgress);
function onUploadProgress (event:Object) {
   YAHOO.log(event.id + ": " + event.bytesLoaded + "/" +
             event.bytesTotal);
}
```

Send custom variables in the same POST request as the file submission:

```
myUploader.upload("file0", "YOUR UPLOAD URL", "POST", {var1:
"foo", var2: "bar", var3: "baz"});
```

Modify the file form field name from the default “Filedata”:

```
myUploader.upload("file0", "YOUR UPLOAD URL", POST, null,
"DifferentFileFieldName");
```

Accept the file upload using PHP on the server side:

```
<?php
foreach ($_FILES as $fieldName => $file) {
move_uploaded_file($file['tmp_name'], "./" . $file['name']);
echo (" ");
}
exit();?>
```

Dependencies

YUI: Yahoo Global **Object**, **Dom**, **Event**, **Element** and uploader.swf.
Client: Flash **9.0.45** or later installed on their browser.

YAHOO.widget.Uploader Properties:

SWFURL (str)

YAHOO.widget.Uploader Events:

contentReady
fileSelect
uploadStart
uploadProgress
uploadCancel
uploadComplete
uploadCompleteData
uploadError

When transparent:
mouseUp
mouseDown
rollOver
rollOut
click

YAHOO.widget.Uploader Methods:

setAllowLogging()
setAllowMultiple()
setFileFilters()
enable()
disable()
upload()
uploadAll()
cancel()
clearFileList()