

## Coder un formulaire pour l'upload d'image

Catégorie : **Fiches techniques**

Publié par [Fooups](#) le 29/09/2005

- **Attention** : les variables `$_FILES`, `$_VARS`, etc ont été introduites à partir de PHP 4.10. Il ne faut plus utiliser `$HTTP_POST_VARS`, `$HTTP_POST_FILES`, etc.

### 1) Upload par la méthode Post

#### **1-1) Rappel du principe**

Lorsque l'on soumet un formulaire contenant un champ file, le fichier est tout d'abord transféré du PC client sur le serveur, dans le répertoire temporaire défini au niveau de la configuration. (Ex avec EasyPhp: `upload_tmp_dir = "C:\Program Files\EasyPHP1-7tmp"`).

Ce fichier doit ensuite être traité par le script exécuté après soumission du formulaire : -> vérification du fichier uploadé: taille, extension, ... ->enregistrement de ce fichier dans un répertoire spécifique. A la fin de l'exécution du script, le fichier uploadé est systématiquement effacé du répertoire temporaire où il avait été enregistré initialement. Cela signifie 2 choses : ->aucun contrôle du fichier ne peut être réalisé sur le PC client avant l'upload -> le fichier uploadé doit être enregistré dans un répertoire de votre espace web au cours de l'exécution du script, sinon il est 'perdu'. **2-2) La variable `$_FILES`**

Supposons que votre formulaire contienne un champ file :

Lorsqu'un formulaire contenant un champ 'file' est soumis, le fichier est uploadé, et vous disposez, pour traiter ce fichier, de la variable `$_FILES` (de type array) vous permettant de connaître: `$_FILES['my_file']['name']`

Le nom original du fichier, tel que sur la machine du client web. `$_FILES['my_file']['type']`

Le type MIME du fichier, si le navigateur a fourni cette information. Ex.: "image/gif", "image/png", etc. `$_FILES['my_file']['size']`

La taille, en octets, du fichier téléchargé.

`$_FILES['my_file']['tmp_name']`

Le nom temporaire alloué au fichier uploadé sur la machine serveur. `$_FILES['my_file']['error']`

Le code d'erreur error code associé au téléchargement de fichier. **3) Paramètres du fichier uploadé**

Prévoyez de préférence dans l'administration de votre module les options permettant de pouvoir fixer les valeurs maxi pour la hauteur, largeur et taille du fichier. Cette partie peut être réalisée par la suite.

Fichier `xoops_version.php`

```
$modversion['config'][1]['name'] = 'max_imgheight';  
$modversion['config'][1]['title'] = '_MI_MAX_HEIGHT' ,  
$modversion['config'][1]['description'] = '_MI_DESC_MAXHEIGHT';  
$modversion['config'][1]['formtype'] = 'textbox' ,  
$modversion['config'][1]['valuetype'] = 'int' ;  
$modversion['config'][1]['default'] = '1024' ,
```

```
$modversion['config'][2]['name'] = 'max_imgwidth';  
$modversion['config'][2]['title'] = '_MI_MAX_WIDTH' ,
```

```
$modversion['config'][2]['description'] = '_MI_DESC_MAXWIDTH';
$modversion['config'][2]['formtype'] = 'textbox' ,
$modversion['config'][2]['valuetype'] = 'int' ;
$modversion['config'][2]['default'] = '1024' ,
```

```
$modversion['config'][3]['name'] = max_imgsize;
$modversion['config'][3]['title'] = '_MI_MAX_SIZE' ,
$modversion['config'][3]['description'] = '_MI_DESC_MAXSIZE';
$modversion['config'][3]['formtype'] = 'textbox' ,
$modversion['config'][3]['valuetype'] = 'int' ;
$modversion['config'][3]['default'] = '100000' ,
```

#### 4) Le formulaire

##### 4-1) XoopsFormFile

Lorsque l'on insère un champ de type file dans un formulaire par cette méthode :

```
$form->addElement(new XoopsFormFile("Selectionner une image", "my_file", 30000));
```

xoops insère en réalité 3 champs: le champ file et 2 champs cachés :

```
color=#009900>
```

##### 4-2) Code du formulaire

```
$form = new XoopsThemeForm(_MD_TITLE_FORM, "myform", "submit.php");
$form->setExtra( 'enctype='multipart/form-data' ); //
```

// l'emplacement choisi pour le champ d'upload

```
$img_box = new XoopsFormFile(_MD_ADDIMG, "my_file", $max_imgsize);
$img_box->setExtra( "size ='65'" ); // optionnel : pour fixer la longueur de la textbox
$form->addElement($img_box);
```

##### 5) Traitement du formulaire soumis

Tout d'abord, initialisation des variables

// Taille maxi du fichier

```
$max_imgsize = 5000; // ou = $xoopsModuleConfig[max_imgsize]
```

// Largeur maxi de l'image

```
$max_imgwidth = 120; // ou = $xoopsModuleConfig[max_imgwidth]
```

// Hauteur maxi de l'image

```
$max_imgheight = 120; // ou = $xoopsModuleConfig[max_imgheight]
```

// Formats dont vous autorisez l'upload

```
$allowed_mimetypes = array('image/gif', 'image/jpeg', 'image/pjpeg', 'image/x-png');
```

// Répertoire d'upload

```
$img_dir = XOOPS_ROOT_PATH . "/modules/mymodule/images" ; // ou =
XOOPS_UPLOAD_PATH; (répertoire upload de xoops)
include_once(XOOPS_ROOT_PATH."/class/uploader.php");
```

// récupération du nom du fichier d'origine

```
// on pourrait également utiliser $field = $_FILES['my_file']['name']
```

```
$field = $_POST["xoops_upload_file"][0] ;
```

```

// test si le champ a été rempli :
if( !empty( $field ) || $field != "" ) {

// puis test si le fichier a été uploadé dans le répertoire temporaire:
if( $_FILES[$field]['tmp_name'] == "" || ! is_readable( $_FILES[$field]['tmp_name'] ) ) {

// non = l'upload a échoué on retourne au formulaire
redirect_header( 'submit.php' , 2, _MD_FILEERROR ) ;
exit ;
}
// création de l'objet uploader
$uploader = new XoopsMediaUploader($img_dir, $allowed_mimetypes, $max_imgsize,
$max_imgwidth, $max_imgheight);

// test si le fichier uploadé est conforme en dimension et taille, et bien copié du répertoire
temporaire au répertoire indiqué
if( $uploader->fetchMedia( $field ) && $uploader->upload() ) {

// L'upload a aussi - à adapter bien évidemment
echo 'File uploaded successfully!';
// nom du fichier uploadé
echo 'Saved as: ' . $uploader->getSavedFileName() ;
// chemin complet
echo 'Full path: ' . $uploader->getSavedDestination();

// sinon l'upload a échoué : message d'erreur
} else {
    echo $uploader->getErrors();
}
}
}

```

## 6) Nom du fichier uploadé

Dans l'exemple précédent votre image sera enregistrée dans le répertoire indiqué en conservant son nom d'origine.

Si vous voulez nommez ce fichier différemment (mais en conservant l'extension !), 2 possibilités:

Nom aléatoire avec préfixe spécifique (ex: **img**405415ee23bcb.jpg)

```

$uploader = new XoopsMediaUploader($img_dir, $allowed_mimetypes, $max_imgsize,
$max_imgwidth, $max_imgheight);
$uploader->setPrefix( 'img' ) ; Nom complet spécifique (ex: ma_photo.jpg)

```

```

// extraction extension du fichier
$ext = preg_replace( "/^+\.([\w.]+)$/sU" , "\1" , $_FILES["my_file"]["name"] ) ;

```

```

// création du nom complet avec l'extension

```

```

$name_img = 'ma_photo.'.$ext;

```

```
$uploader = new XoopsMediaUploader($img_dir, $allowed_mimetypes, $max_imgsize,  
$max_imgwidth, $max_imgheight);  
// spÃ©cification du nom  
$uploader->setTargetFileName($name_img);
```