From EULER Project

PmWiki: Uploads Administration

PmWiki includes a script called *upload.php* that allows users to upload files to the wiki server using a web browser. Uploaded files (also called *attachments*) can then be easily accessed using markup within wiki pages. This page describes how to install and configure the upload feature.

Some notes about security

PmWiki takes a somewhat, but justifiable, paranoid stance when it comes to the uploads feature. Thus, the default settings for uploads tend to try to restrict the feature as much as possible:

- The upload function is disabled by default
- Even if you enable it, the function is password locked by default
- Even if you remove the password, you're restricted to uploading files with certain names, extensions, and sizes
- The characters that may appear in upload filenames are (default) alphanumerics, hyphen, underscore, dot, and space (see also here).
- The maximum upload size is small (50K by default)

This way the potential damage is limited until/unless the wiki administrator explicitly relaxes the restrictions.

Keep in mind that letting users (anonymously!) upload files to your web server does entail some amount of risk. The *upload.php* script has been designed to reduce the hazards, but wiki administrators should be aware that the potential for vulnerabilities exist, and that misconfiguration of the upload utility could lead to unwanted consequences.

By default, authorized users are able to overwrite files that have already been uploaded, without the possibility of restoring the previous version of the file. If you want to disallow users from being able to overwrite files that have already been uploaded, add the following line to *config.php*:

```
$EnableUploadOverwrite = 0;
```

Alternatively, an administrator can keep older versions of uploads.

An administrator can also configure PmWiki so the password mechanism controls access to uploaded files.

Basic installation

The *upload.php* script is automatically included from *stdconfig.php* if the `$EnableUpload` variable is true in *config.php*. In addition, *config.php* can set the `$UploadDir` and `$UploadUrlFmt` variables to specify the local directory where uploaded files should be stored, and the URL that can be used to access that directory. By default, `$UploadDir` and `$UploadUrlFmt` assume that uploads will be stored in a directory called *uploads/* within the current directory (usually the one containing *pmwiki.php*). In addition, *config.php* should also set a default upload password (see PasswordsAdmin).

Thus, a basic *config.php* for uploads might look like:

```
<?php if (!defined('PmWiki')) exit();
```

```
##  Enable uploads and set a site-wide default upload password.
$EnableUpload = 1;
$DefaultPasswords['upload'] = crypt('secret');
```

If you have edit passwords and wish to allow all users with edit rights to upload, instead of $DefaultPasswords['upload'], you can set $HandleAuth['upload'] = 'edit'; in config.php.

Important: do NOT create the uploads directory yet! See the next paragraph.

You may also need to explicitly set which filesystem directory will hold uploads and provide a URL that corresponds to that directory like:

```
$UploadDir = "/home/foobar/public_html/uploads";
$UploadUrlFmt = "http://example.com/~foobar/uploads";
```

Upload directory configuration

Uploads can be configured site-wide, by-group, or by-page by changing $UploadPrefixFmt. This determines whether all uploads go in one directory for the site, an individual directory for each group, or an individual directory for each page. The default is to organize upload by group.

For site-wide uploads, use

```
$UploadPrefixFmt = '';
```

To organize uploads by page, use:

```
$UploadPrefixFmt = '/$Group/$Name';
```

The upload directory

For the upload feature to work properly, the directory given by $UploadDir must be writable by the web server process, and it usually must be in a location that is accessible to the web somewhere (e.g., in a subdirectory of *public_html*). Executing PmWiki with uploads enabled will prompt you with the set of steps required to create the uploads directory on your server (it differs from one server to the next). *Note that you are likely to be required to explicitly create writable group- or page-specific subdirectories as well!*

Uploading a file

Once the upload feature is enabled, users can access the upload form by adding "?action=upload" to the end of a normal PmWiki URL. The user will be prompted for an upload password similar to the way other pages ask for passwords (see Passwords and PasswordsAdmin for information about setting passwords on pages, groups, and the entire site).

Another way to access the upload form is to insert the markup "Attach:filename.ext" into an existing page, where filename.ext is the name of a new file to be uploaded. When the page is displayed, a '?-link' will be added to the end of the markup to take the author to the upload page. (See Uploads for syntax variations.)

By default, PmWiki will organize the uploaded files into separate subdirectories for each group. This can be changed by modifying the $UploadPrefixFmt variable. See Cookbook:UploadGroups for details.

Versioning Uploaded Files

PmWiki does not manage versioning of uploaded files by default. However, by setting $EnableUploadVersions=1; an administrator can have older versions of uploads preserved in the uploads directory along with the most recent version.

Upload restrictions

Restricting uploaded files for groups and pages

Uploads can be enabled only for specific groups or pages by using a per group customization. Simply set $EnableUpload=1; for those groups or pages where uploading is to be enabled; alternately, set $EnableUpload=1; in the config.php file and then set $EnableUpload=0; in the per-group or per-page customization files where uploads are to be disabled.

Restricting total upload size for a group or the whole wiki

Uploads can be restricted to an overall size limit for groups. In the group configuration file (i.e., local/Group.php), add the line

$UploadPrefixQuota = 1024000; # limit group uploads to 1000K
This will limit the total size of uploads for that group to 1000k --any upload that pushes the total over the limit will be rejected with an error message. This value defaults to zero (unlimited).

Uploads can also be restricted to an overall size limit for all uploads. Add the line

$UploadDirQuota = 10240000; # limit total uploads to 10000K
This will limit the total size of uploads for the whole wiki to 10000k --any upload that pushes the total over the limit will be rejected with an error message. This value defaults to zero (unlimited).

Restricting uploaded files type and size

The upload script performs a number of verifications on an uploaded file before storing it in the upload directory. The basic verifications are described below.

filenames
: the name for the uploaded file can contain only letters, digits, underscores, hyphens, spaces, and periods, and the name must begin and end with a letter or digit.

file extension
: only files with approved extensions such as ".gif", ".jpeg", ".doc", etc. are allowed to be uploaded to the web server. This is vitally important for server security, since the web server might attempt to execute or specially process files with extensions like ".php", ".cgi", etc.

file size
: By default all uploads are limited to 50K bytes, as specified by the $UploadMaxSize variable. Thus, to limit all uploads to 100K, simply specify a new value for $UploadMaxSize in *config.php*:

```
$UploadMaxSize = 102400;
```

However, maximum file sizes can also be specified for each type of file uploaded. Thus, an administrator can restrict "`.gif`" and "`.jpeg`" files to 20K, "`.doc`" files to 200K, and all others to the size given by `$UploadMaxSize`. The $UploadExtSize array is used to determine which file extensions are valid and the maximum upload size (in bytes) for each file type. For example:

```
$UploadExtSize['gif'] = 20000; # limit .gif files to 20K
```

Setting an entry to zero disables file uploads of that type altogether:

```
$UploadExtSize['zip'] = 0; # disallow .zip files
```

You can limit which types of files are uploadable by disabling all defaults and specifying only desired types Setting the variable $UploadMax to zero will disable all default file types. Individual file types may then be enabled by setting their maximum size with the variable $UploadExtSize.

```
# turns off all upload extensions
$UploadMaxSize = 0;

# enable only these file types for uploading
$aSize=102400; // 100 K file size limitation
$UploadExtSize['jpg' ] = $aSize;
$UploadExtSize['gif' ] = $aSize;
$UploadExtSize['png' ] = $aSize;
```

Adding new file types to permitted uploads

To add a new extension to the list of allowed upload types, add a line like the following to a local customization file:

```
$UploadExts['ext'] = 'content-type';
```

where *ext* is the extension to be added, and *content-type* is the "MIME type", or content-type (which you may find here(approve sites) or on the lower part of this page(approve sites)) to be used for files with that extension. For example, to add the '`dxf`' extension with a Content-Type of '`image/x-dxf`', place the line

```
$UploadExts['dxf'] = 'image/x-dxf';
```

Each entry in $UploadExts needs to be the extension and the mime-type associated with that extension, thus:

```
$UploadExts = array(
  'gif' => 'image/gif',
  'jpeg' => 'image/jpeg',
  'jpg' => 'image/jpeg',
  'png' => 'image/png',
  'xxx' => 'yyyy/zzz'
);
```

For the types that PmWiki already knows about it's not necessary to repeat them here (the *upload.php* script adds PmWiki's defaults to whatever the administrator supplies). See also Cookbook:UploadTypes for additional types.

Other file size limits

There are other factors involved that affect upload file sizes. In Apache 2.0, there is a LimitRequestBody

directive that controls the maximum size of anything that is posted (including file uploads). Apache has this defaulted to unlimited size. However, some Linux distributions (e.g., Red Hat Linux) limit postings to 512K so this may need to be changed or increased. (Normally these settings are in an *httpd.conf* configuration file or in a file in */etc/httpd/conf.d*.)

Problem noted on Red Hat 8.0/9.0 with Apache 2.0.x, the error "Requested content-length of 670955 is larger than the configured limit of 524288" was occurring under Apache and a "Page not found" would appear in the browser. Trying the above settings made no change with PHP, but on Red Hat 8.0/9.0 there is an additional PHP config file, /etc/httpd/conf.d/php.conf, and increasing the number on the line "LimitRequestBody 524288" solves the issue.

PHP itself has two limits on file uploads (usually located in /etc/php.ini). The first is the `upload_max_filesize` parameter, which is set to 2M by default. The second is `post_max_size`, which is set to 6M by default.

With the variables in place--PmWiki's maximum file size, Apache's request-size limits, and the PHP file size parameters, the maximum uploaded file size will be the smallest of the three variables.

Password protecting uploaded files

Setting a read password for pages (and groups) will prevent an attached file from being seen or accessed through the page, but to prevent direct access to the file location (the uploads/ directory) one can do the following:

- In local/config.php set `$EnableDirectDownload`=0;
- If you use per-group upload directories (PmWiki default, see `$UploadPrefixFmt`), add to config.php `$EnableUploadGroupAuth = 1;`
- Deny public access to the uploads/ directory through moving it out of the html/ or public_html/ directory tree, or through a .htaccess file.

See Cookbook:Secure attachments

Other notes

- If uploads doesn't seem to work, make sure that your PHP installation allows uploads. The *php.ini* file (usually */etc/php.ini* or */usr/local/lib/php.ini*) should have

```
file_uploads = On
```

Note that if you change this value, httpd must generally be restarted. Another way to check if uploads are allowed by the server is to set `$EnableDiag` to 1 in *config.php*, and set ?action=phpinfo on a URL. The "`file_uploads`" variable must have a value of 1 (if it says "`no value`", that means it's off).

How do I disable uploading of a certain type of file?

Here's an example of what to add to your *local/config.php* file to disable uploading of .zip files:

```
$UploadExtSize['zip'] = 0; # Disallow uploading .zip files.
```

How do I attach uploads to individual pages or the entire site, instead of organizing them by wiki group?

Use the $UploadPrefixFmt variable (see also the Cookbook:UploadGroups recipe).

```
$UploadPrefixFmt = '/$FullName'; # per-page
$UploadPrefixFmt = ''; # site-wide
```

For $UploadDirQuota - can you provide some units and numbers? Is the specification in bytes or bits? What is the number for 100K? 1 Meg? 1 Gig? 1 Terabyte?

Units are in bytes.

```
$UploadDirQuota = 100*1024;         # limit uploads to 100KiB
   $UploadDirQuota = 1000*1024;        # limit uploads to 1000KiB
   $UploadDirQuota = 1024*1024;        # limit uploads to 1MiB
   $UploadDirQuota = 25*1024*1024;     # limit uploads to 25MiB
   $UploadDirQuota = 2*1024*1024*1024; # limit uploads to 2GiB
```

Is there a way to allow file names with Unicode or addtiional characters?

Yes, see $UploadNameChars

Where is the list of attachments stored?

It is generated on the fly by the

markup.

How can I find orphaned or missing attachments

See Cookbook:Attachlist enhanced

How can I prevent hotlinking of my uploaded images

See Cookbook:Prevent Hotlinking

I have limited the max upload size to 8 Mb in config.php, however only files smaller than 2M can be uploaded.

Check your php.ini for *upload_max_filesize*

```
upload_max_filesize = 8M
```

Retrieved from https://www-sop.inria.fr/mascotte/EULER/wiki/pmwiki.php/PmWiki/UploadsAdmin
Page last modified on July 16, 2009, at 03:20 AM