

Configure Owl for Large Files

Written by Owl DEV Team

Wednesday, 05 August 2009 04:23 - Last revised Wednesday, 05 August 2009

Though PHP presents a very versatile and user friendly interface for handling file uploads, the default installation is not geared for working with files in excess of 2 Mega Bytes. This article will help you configure your PHP engine for handling such large file transfers.

The php.ini File

All the configuration settings for your installation are contained in the php.ini file. Sometimes these setting might be overridden by directives in apache .htaccess files or even with in the scripts themselves. However you cannot over ride some of the settings that effect file uploads with .htaccess directives in this way. So let's just concentrate on the ini file.

You can call the phpinfo() function to find the location of your php.ini file, it will also tell you the current values for the following settings that we need to modify

- file_uploads
- upload_max_filesize
- max_input_time
- memory_limit
- max_execution_time
- post_max_size

To run this function you can write a simple php script, drop it into your webspace, and call it from your browser.

```
<?php  
phpinfo();  
?>
```

Now let's look at the parameter that interests us.

file_uploads

This parameter is fairly obvious if you set this off, uploading is disabled for your installation, must be set to On.

Configure Owl for Large Files

Written by Owl DEV Team

Wednesday, 05 August 2009 04:23 - Last revised Wednesday, 05 August 2009

upload_max_filesize and post_max_size

Files are usually POSTed to the webserver in a format known as 'multipart/form-data'. The `post_max_size` sets the upper limit on the amount of data that a script can accept in this manner. Ideally this value should be larger than the value that you set for `upload_max_filesize`. It's important to realize that `upload_max_filesize` is the sum of the sizes of all the files that you are uploading. `post_max_size` is the `upload_max_filesize` plus the sum of the lengths of all the other fields in the form plus any mime headers that the encoder might include. Since these fields are typically small you can often approximate the upload max size to the post max size. According to the PHP documentation you can set a `MAX_UPLOAD_LIMIT` in your HTML form to suggest a limit to the browser.

This is accomplished by the following Owl Parameter, the parameter takes a value in bytes, so if for example you wish to set a limit of 5MB you calculate the number of bytes by the following formula $\langle \text{number of MB} \rangle * 1024 * 1024$.

From the Administration Panel, under Site Feature:

memory_limit

When the PHP engine is handling an incoming POST it needs to keep some of the incoming data in memory. This directive has any effect only if you have used the `--enable-memory-limit` option at configuration time. Setting too high a value can be very dangerous because if several uploads are being handled concurrently all available memory will be used up and other unrelated scripts that consume a lot of memory might effect the whole server as well.

max_execution_time and max_input_time

These settings define the maximum life time of the script and the time that the script should spend in accepting input. If several mega bytes of data are being transferred `max_input_time` should be reasonably high.

Additional Comments

Apache Settings

The apache webserver has a `LimitRequestBody` configuration directive that restricts the size of all POST data regardless of the web scripting language in use. Some RPM installations sets limit request body to 512Kb. You will need to change this to a larger value or remove the entry altogether.

On our install of Fedora that limit is set in the `/etc/httpd/conf.d/php.conf`

```
#
```

```
# PHP is an HTML-embedded scripting language which attempts to make it  
# easy for developers to write dynamically generated webpages.
```

Configure Owl for Large Files

Written by Owl DEV Team

Wednesday, 05 August 2009 04:23 - Last revised Wednesday, 05 August 2009

```
#
LoadModule php4_module modules/libphp4.so
LimitRequestBody 102400000
#
# Cause the PHP interpreter to handle files with a .php extension.
#
AddType application/x-httpd-php .php
# AddType application/x-httpd-php-source .phps
#
# Add index.php to the list of files that will be served as directory
# indexes.
#
DirectoryIndex index.php
```

Trouble Shooting

The apache error log is a great source of information when trouble shooting file Upload issues, usually located `/var/log/httpd/error_log`

Take a look for messages that most time will tell you which of the above directive has been exceeded.

If you make any changes, to the `php.ini` file or any of the `httpd.conf` files don't forget to restart the web server process.