

Server Settings

Joomla specifies certain settings that are recommended for proper functioning of the system. A list of the recommended and actual settings is displayed when you install Joomla. One of the recommended settings is to have 'Display Errors' switched on. This is very useful when developing and debugging a site, but there is a security vulnerability in PHP (not Joomla, but the language in which Joomla was written) which may allow cross-site-scripting attacks when the display errors option is enabled, if you have a script which produces an error.

Beginning with Joomla 1.0.8, you can suppress error messages by going to Site->Global Configuration, and clicking on the 'Server' tab. Set the 'Error Reporting' option to 'None'. If you are not using the very latest version of Joomla, it would be a very good idea to upgrade!

Otherwise, to turn off display of errors, you need to change some settings in a file called php.ini - you might not have access to this file if you use shared hosting, but it might be possible to add your own php.ini file to the root folder of your website which will only affect your site and nobody elses (or you might need to add it to every folder that contains php files). Alternatively, depending on the configuration settings on your server, you might be able to override individual php.ini settings in your .htaccess file.

The settings that need to be specified in php.ini are:

```
display_errors = Off
```

```
html_errors = Off
```

```
display_startup_errors = Off
```

```
log_errors = On
```

For additional security it may be worthwhile disabling certain PHP functions. The following 2 lines, when added to php.ini will prevent the listed functions from working. If you have a third party script that relies on one or more of these functions, it will break when you turn them off like this. Joomla does not use these functions, but some third party components might do. Disabling these functions will help to protect your site from hackers though.

```
allow_url_fopen = Off
```

```
disable_functions = show_source, system, shell_exec, passthru, exec, phpinfo, popen, proc_open, tempnam
```

If you don't have access to the global php.ini file, you might be able to add your own. More information about doing this can be found here: <http://www.washington.edu/computing/web/publishing/php-ini.html>. You might need to ask your host

to restart the Apache web server before your overridden settings will take effect (this does not mean rebooting the server, just restarting Apache - which only takes a few seconds).

Note: If you encode your PHP files with Zend Optimizer, adding your own local php.ini file can cause PHP to think that Zend Optimizer is not installed even if it is.

If your server configuration allows it, you may be able to just add the following lines to your .htaccess file to override the settings without needing your own php.ini file. Try adding the following to the end of your .htaccess file (if your server does not recognise the directives, you will get an error message when you try to access your site):

```
php_flag display_errors "0"
```

```
php_flag html_errors "0"
```

```
php_flag display_startup_errors "0"
```

```
php_flag log_errors "1"
```

```
php_flag allow_url_fopen "0"
```

These settings will cause any PHP errors to be logged in a text file instead of being displayed in the user's browser window.

If your web server uses Apache, it is worthwhile installing the mod_security Apache module. This helps protect against various forms of attack over HTTP. Ask your web host whether this module is active, and if not, ask them if they will activate it. There may be some undesirable side-effects, but these can usually be mitigated by tweaking the rules in mod_security.