MultiPHP INI Editor for WHM

(*WHM >> Home >> Software >> MultiPHP INI Editor*)

> **Important:**
> WHM's *MultiPHP INI Editor* interface **only** exists on systems that run EasyApache 4.

Overview

This interface allows you to configure your PHP settings. The PHP scripting language generates web pages.

For more information about a directive, search the php.net documentation.

> **Note:**
> cPanel users can view information about the server's PHP configuration in cPanel's *MultiPHP INI Editor* interface (*cPanel >> Home >> Software >> MultiPHP INI Editor*).

MultiPHP INI Editor

Select the PHP version that you wish to configure from the menu. The PHP directives will appear.

Select a tab to view more information about that section of the *MultiPHP INI Editor* interface.
Basic Mode Editor Mode

The *Basic Mode* section of this interface loads the directive values from the `php.ini` file for the version of PHP that you select. Make your changes to the desired directives and click *Save*.

The system saves the settings to the `/opt/cpanel/<your_php_package>/root/etc/php.d/php.ini` file, where `<your_php_package>` represents the name of the PHP package that contains the `.ini` file that you wish to edit.

> **Note:**
> The availability of the following directives depends on your version of PHP. If a directive does **not** appear in your interface, your version of PHP does **not** support it.

PHP directive	Description	PHP default
allow_url_fopen	Select whether PHP scripts can use URL-aware `fopen` wrappers.	*Enabled*
allow_url_include	Select whether PHP scripts can use URL-aware `fopen` wrappers with the following functions: • `include()` • `include_once()` • `require()` • `require_once()`	*Disabled*

asp_tags	Select whether PHP scripts can use ASP-like tags in addition to PHP tags. > **Note:** > This includes variable-value printing shorthand (for example, `<%= $value %>`).	*Disabled*
display_errors	Select whether PHP displays errors as output or hides them from the user.	*Disabled*
enable_dl	Select whether scripts can call the `dl` function. This function dynamically loads a PHP extension at runtime. > **Warning:** > **If** you set this directive to *Enabled*, it poses security risks to your server. We **strongly** recommend that you select *Disabled* unless you **absolutely** require the `dl` function.	*Disabled*
file_uploads	Select whether your PHP scripts can upload HTTP files.	*Disabled*
magic_quotes_gpc	Select whether PHP automatically escapes certain characters with a backslash (`\`).	*Disabled*
max_execution_time	Enter the maximum amount of time, in seconds, that your server allows a script to run before your server terminates it. This setting limits the amount of memory that poorly-written scripts may use.	`30`

max_input_time	Enter the maximum amount of time, in seconds, that your server allows a script to parse input data.	`-1`
max_input_vars	Enter the maximum number of input variables per request. Use this directive to deter Denial of Service (DoS) attacks that use hash collisions on input variable names.	`1000`
memory_limit	Enter the maximum amount of memory that scripts can allocate, in bytes. This setting helps to prevent poorly-written scripts that may use too much memory. cPanel & WHM sets the default value based on the system's memory during installation. **Note:** Use the character `M` to define the limit in Megabytes.	cPanel & WHM adjusts this setting at the time of install, based on system memory: • 1 GB — `32M` • 2 - 4GB — `64M` • Over 4 GB — `128M`

post_max_size	Enter the maximum size of post data that the system allows, in bytes. This setting affects file uploads. Set this value to `0` to allow unlimited post sizes. **Notes:** • We **strongly** recommend that you set this value larger than the `upload_max_filesize` value and smaller than the `memory_limit` value. • Use the character `M` to define the limit in Megabytes.	`8M`
register_globals	Select whether GET or POST input converts into variables that the script can use.	*Disabled*
safe_mode	Select whether to enable safe mode. Safe mode increases security for PHP scripts and enables user, group, and environment checks and controls.	*Disabled*
session.save_path	Enter the directory in which your server stores the files that PHP creates. **Important:**	`NULL`

- The system saves PHP sessions in the `/var/cpanel/php/sessions` directory by default. EasyApache 4 installations create subdirectories based on the version of PHP that exists on the server. If multiple versions of PHP exist on the system, a subdirectory exists for each version of PHP. The `root` user owns each directory, and each directory possesses `1733` permissions. This allows PHP to write session files, but users will not possess permissions to delete those sessions in the directory. Because PHP writes setting files, PHP cannot clean its own sessions. Therefore, the `root` user **must** run a cron job to delete the expired session files.

- Do not change the `session.save_path` setting to the `/tmp` directory or a directory that other users or programs use. If you change the `session.save_path` setting's value to another directory, the system will change the directory's permissions to `1733`. This will likely break other programs' functionality. If you must use the `/tmp` directory, create a subdirectory under the `/tmp` directory in which the system will save the PHP session files.

Note:
If you change the `session.save_path` setting in the `php.ini` configuration file, the cron job will continue to clean up the sessions files in the specified directory. When the system runs the cron job, it will ensure that the configuration file possesses the correct directory permissions.

session.gc_maxlifetime	Enter the maximum amount of time, in seconds, for the lifetime of a session file. If a session lasts longer than the `session.max_lifetime`, the system marks it for deletion the next time that it performs a session cleanup. Each version of PHP that you install on the system uses this setting. This setting defaults to `1440` seconds (24 minutes). **Note:** Session files can exist on the filesystem for longer than the `session.max_lifetime` setting. Because session cleanup occurs every 30 minutes, a session file could exist for up to its `session.max_lifetime` value plus 30 minutes.	`1440`
session.gc_probability	This setting defaults to `0`. Because a cron job cleans up the session files, the system does **not** require this setting. If you enable this setting, PHP will attempt to delete expired files, which will fail.	`0`
session.gc_divisor	This setting defaults to `0`. Because a cron job cleans up the session files, the system does **not** require this setting. If you enable this setting, PHP will attempt to delete expired files, which will fail.	`0`

upload_max_filesize	Enter the maximum file size for an upload, in bytes.	2M
	Note: Use the character `M` to define the limit in Megabytes.	
zlib.output_compression	Select whether to transparently compress pages when the browser sends an `Accept-Encoding: gzip` or `deflate` header.	*Disabled*

Warnings:
- Exercise extreme caution when you make changes in this interface, because errors could result in non-functional PHP scripts. **Only** advanced users should use this interface.
- Do **not** use the following reserved terms as directives:
  - `null`
  - `yes`
  - `no`
  - `true`
  - `false`
  - `on`
  - `off`
  - `none`
- Directives **cannot** contain any the following special characters: `?{}|&~![()^"`
- The following values set the directive value to `""`: `null`, `off`, `no`, and `false`.
- The following values set the directive value to `1`: `on`, `yes`, and `true`.

The *Editor Mode* section of this interface loads the contents of the `php.ini` file for your selected version of PHP. Make your changes to the desired directives and click *Save*.

The system saves the settings to the `/opt/cpanel/<your_php_package>/root/etc/php.d/php.ini` file, where `<your_php_package>` represents the name of the PHP package that contains the `.ini` file that you wish to edit. The interface attempts a rudimentary validation of directive names to ensure that they follow the proper format. This does **not** guarantee that your PHP installation will function. To reset your PHP settings to default values, delete the PHP version's `php.ini` file.

Additional documentation

Suggested documentation For cPanel users For WHM users For developers

- The rebuild_phpconf Script
- PHP-FPM User Pools
- PHP-FPM Code and FileSystem Layout for EasyApache 4
- Apache vs PHP-FPM Bottleneck with Child Processes
- PHP-FPM