

## How to Turn Off, Suppress PHP Notices and Warnings - PHP error handling levels via php.ini and PHP source code

Author : admin



PHP Notices are common to occur **after PHP version upgrades** or where an *obsolete PHP code is moved from Old version PHP to new version*. This is common *error in web software using Frameworks which have been abandoned by developers*.

Having PHP Notices to appear on a webpage is pretty ugly and give a lot of information which might be used by malicious crackers to try to break your site thus it is always a good idea to **disable PHP Notices**. There are plenty of ways to *disable PHP Notices*

The easiest way to disable it is globally in all Webserver PHP library via **php.ini (/etc/php.ini)** open it and make sure [display\\_errors](#) is disabled:

```
display_errors = 0
```

or

```
display_errors = Off
```

Note that that some claim in *PHP 5.3 setting display\_errors to Off will not work as expected*. Anyways [to make sure where your loaded PHP Version display\\_errors is ON or OFF use phpinfo\(\)](#):

It is also possible to *disable PHP Notices and error reporting straight from PHP code* you need code like:

or through code:

```
ini_set('display_errors',0);
```

**PHP** has different levels of error reporting, here is complete list of possible error handling variables:

The level of logging could be tuned on Debian Linux via `/etc/php5/apache2/php.ini` or if necessary *to set PHP log level in PHP CLI through `/etc/php5/cli/php.ini`* with:

```
error_reporting = E_ALL & ~E_NOTICE
```

If you need to remove to **remove exact warning or notices from PHP** without changing the way PHPLib behaves is to set `@` in front of variable or *function that is causing NOTICES or WARNING*:  
For example:

```
@yourFunctionHere();  
@var = ...;
```

Its also possible to **Disable PHP Notices and Warnings using .htaccess file** (useful in shared hosting where you *don't have access to global php.ini*), here is **how**:

```
# PHP error handling for development servers
php_flag display_startup_errors off
php_flag display_errors off
php_flag html_errors off
php_flag log_errors on
php_flag ignore_repeated_errors off
php_flag ignore_repeated_source off
php_flag report_memleaks on
php_flag track_errors on
php_value docref_root 0
php_value docref_ext 0
php_value error_log /home/path/public_html/domain/php_errors.log
php_value error_reporting -1
php_value log_errors_max_len 0
```

This way *though PHP Notices and Warnings will be suppressed* errors will get logged into **php\_error.log**