Debugging WordPress: Turning On WP_DEBUG and debug.log from WP Toolkit or wp-config.php, Maintenance Mode, and Reading the Error

WordPress Guides · https://www.hyyat.com/en/knowledgebase/debug-wordpress-wp-toolkit/

In short Turn on debugging with errors saved to a file and **not shown to visitors**: in WP Toolkit switch on **Debugging** with `WP_DEBUG_LOG`, or add the lines to `wp-config.php` by hand. Reproduce the problem, then open `wp-content/debug.log` and read the last error: its path tells you which plugin or theme is at fault. Turn debugging off when you are done.

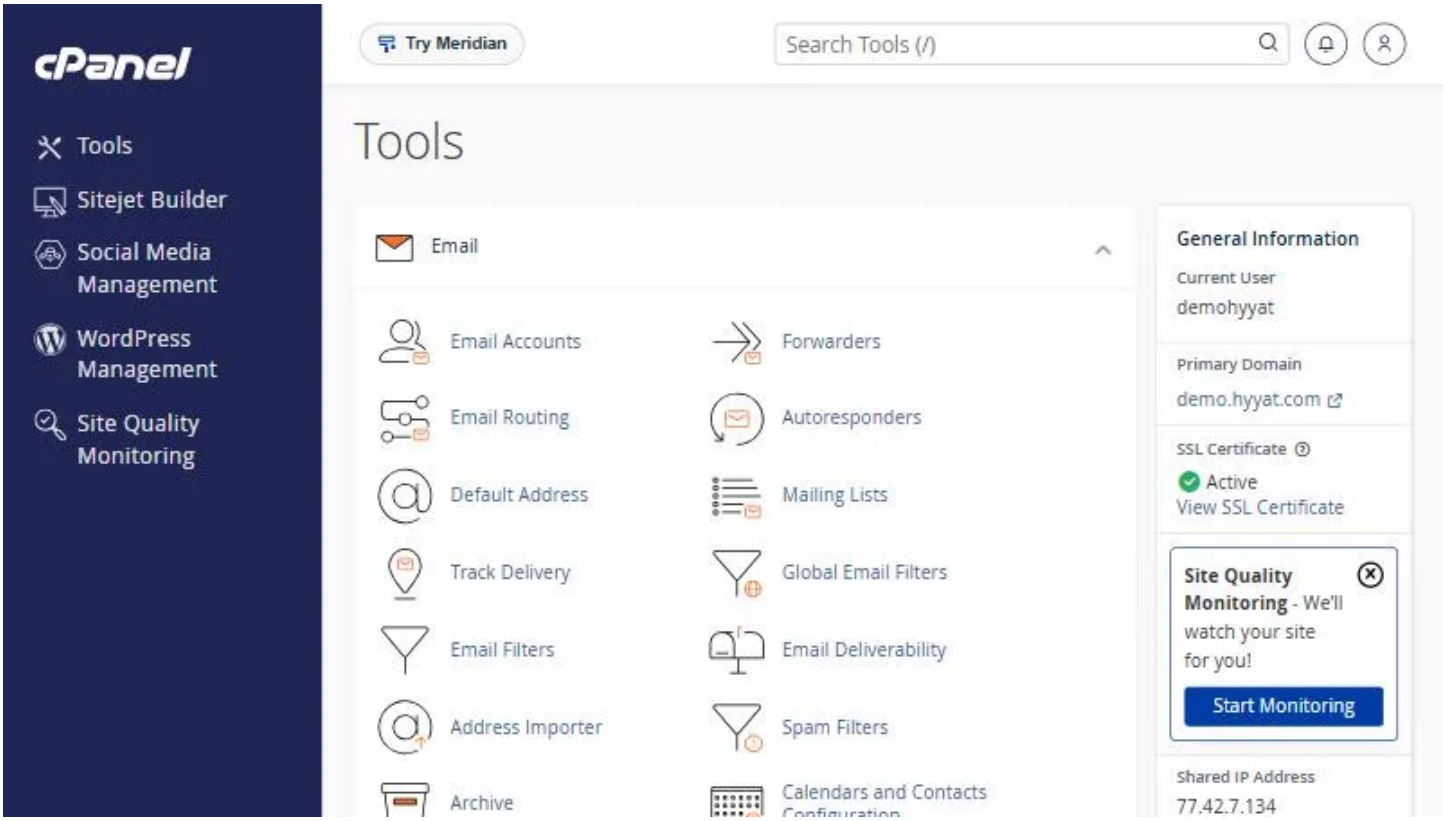
Embedded media: https://www.hyyat.com/wp-content/uploads/2026/11/hyyat-wp-debug-en.mp4?_=1

“There has been a critical error” tells you nothing. Behind it is a real PHP error with a line and a path, and debug mode is how you see it. The idea: make WordPress write its errors to a file you read, not on the page in front of visitors.

The debug constants

Constant	What it does	Suggested value while debugging
`WP_DEBUG`	Turns debug mode on	true
`WP_DEBUG_LOG`	Writes errors to `wp-content/debug.log`	true
`WP_DEBUG_DISPLAY`	Shows errors on pages	false on a live site
`SCRIPT_DEBUG`	Uses unminified CSS and JavaScript	Developers only
`SAVEQUERIES`	Stores database queries for analysis	Developers only; slows the site

Method 1: WP Toolkit



1. **Open WordPress Management** then your site's card.
2. **Debugging** switch it on, then open its settings from the link beside it.
3. **Choose options** turn on WP_DEBUG and WP_DEBUG_LOG, and turn off WP_DEBUG_DISPLAY.
4. **Save** then repeat whatever causes the problem on your site.

Method 2: By hand in wp-config.php

Open `wp-config.php` in File Manager. Find the line `define( 'WP_DEBUG', false );` and replace it with these lines, above the line that says `That's all, stop editing`:



A common mistake: Defining WP_DEBUG twice in the file. You get a new warning, and WordPress may not use the value you wrote. Edit the existing line rather than adding a second one.

Reading debug.log

Open `wp-content/debug.log` and go to the end. The key line looks like: `PHP Fatal error: ... in /home/user/public_html/wp-content/plugins/plugin-name/file.php on line 120`.

If the path shows	The cause	Next step
`wp-content/plugins/name`	A plugin	Disable it: disabling plugins
`wp-content/themes/name`	The theme	Switch to a default theme briefly and contact the theme developer
`Allowed memory size`	Out of memory	the memory error guide
`syntax error`	A typo in code you edited	the syntax error guide

Notice and **Deprecated** lines are hints for developers and do not usually break the site. Focus on **Fatal error**.

Maintenance mode while you fix it

On the site card in WP Toolkit switch on **Maintenance mode**, so visitors see an “under maintenance” page while you work. Switch it off when you are done.

After the fix

- Turn WP_DEBUG off, or set the lines back to false.
- Delete or empty `debug.log`; it can hold paths and details that should not stay around.
- If you cannot find the cause, send the last 20 lines of the file in a support ticket.

Other places errors show up

Source	Where	When it helps
Recovery Mode email	The admin email address	On a fatal error WordPress names the plugin or theme at fault and sends a safe login link
Site Health	Tools › Site Health	Configuration problems: PHP version, updates, scheduled tasks
Server error log	cPanel › Errors	Apache and PHP errors that happen before WordPress runs
error_log file	Sometimes in the site folder itself	PHP errors when WP_DEBUG_LOG is not on

A full example from start to finish

1. **The symptom** after updating plugins, the site showed “There has been a critical error”.
2. **Turning it on** in WP Toolkit: Debugging with WP_DEBUG_LOG, display off.
3. **Reproducing** we loaded the home page once.
4. **Reading** the last line of debug.log: a Fatal error in a file under `wp-content/plugins/` belonging to one of the updated plugins.

5. **Fixing** we deactivated that plugin and the site came back, then restored its previous version from the WP Toolkit backup until a fixed update is released.
6. **Cleaning up** we turned debugging off and deleted debug.log.

Errors you will read often, in plain words

Message	What it means
`Call to undefined function`	Code calls a function that does not exist: often a plugin needing another plugin that is off, or an unsuitable PHP version
`Cannot redeclare`	The same function is defined twice: often two copies of one plugin
`Maximum execution time exceeded`	A task took longer than allowed, such as a big import
`headers already sent`	A space or blank line before `<?php` or after the end of a file you edited

Looking for hosting built for WordPress? See Hyyat Host WordPress hosting.

Our customers asked

debug.log was not created
Make sure WP_DEBUG and WP_DEBUG_LOG are both true and that you repeated the problem after turning them on. wp-content also needs to be writable.

Should I leave debugging on?
No. The file grows fast and reveals details about your site. Turn it on only when needed.

The site is a white page and I cannot reach WP Toolkit
Edit wp-config.php by hand in File Manager; that always works.

I found the error but do not understand it
Copy the whole Fatal error line and send it to the plugin developer, or to us in a ticket.

Need this service from Hyyat Host?Our team handles it for you, with support replying to tickets within 24 hours at most.
WordPress HostingWordPress design