



# Fixing Class "JPlugin" not found in Joomla 5 and 6

Class "JPlugin" not found (or JFactory, JComponentHelper) means a Joomla extension lost the Backward Compatibility plugin. How to find and fix it.

Phil E. Taylor | 14 October 2026

A Joomla site that shows **0 - Class "JPlugin" not found** instead of its home page has an extension written against Joomla's old class names, running on a site where nothing provides those names any more. The same fault turns up as **Class "JFactory" not found**, **Class "JComponentHelper" not found**, **Class "JTable" not found** and a few hundred other variants. Joomla 5 and 6 only keep those names alive through a Backward Compatibility plugin, and the error appears the moment that plugin, or one specific option inside it, is switched off.

Across the Joomla 5 sites in the mySites.guru database, 97.8% still have the old Backward Compatibility plugin enabled, and Joomla's own pre-update check blocks the Joomla 6 upgrade until it is disabled, so this error is about to get much more common. Any of those sites running an extension that still needs the old names will break partway through its Joomla 6 upgrade. mySites.guru shows the state of both compatibility plugins on every connected site, switches them on and off remotely, and on Joomla 4 and 5 scans your files for the old class names before anything breaks.

This post covers how to find which extension is responsible, four ways to get the plugin back on when the site is down (including when you cannot reach the admin), the class names no setting can bring back, and the fix that lasts.

## What does Class "JPlugin" not found mean in Joomla?

**JPlugin** is the name every Joomla plugin extended up to Joomla 3. Joomla 4 renamed it **Joomla\CMS\Plugin\CMSPlugin** and kept the old name as an alias, loaded on every request from **libraries/classmap.php**, so nothing broke. Joomla 5 moved that alias list, around 470 names in all, into a plugin called Behaviour - Backward Compatibility. Joomla 6 moved it again, into Behaviour - Backward Compatibility 6. When the list is not loaded, PHP meets **class PlgSystemSomething extends JPlugin**, has no idea what **JPlugin** is, and stops.

The class name in the message tells you which old API the extension called:

| Error message                      | Modern Joomla class                    |
|------------------------------------|----------------------------------------|
| Class "JPlugin" not found          | Joomla\CMS\Plugin\CMSPlugin            |
| Class "JFactory" not found         | Joomla\CMS\Factory                     |
| Class "JComponentHelper" not found | Joomla\CMS\Component\ComponentHelper   |
| Class "JText" not found            | Joomla\CMS\Language\Text               |
| Class "JTable" not found           | Joomla\CMS\Table\Table                 |
| Class "JRoute" not found           | Joomla\CMS\Router\Route                |
| Class "JHtml" not found            | Joomla\CMS\HTML\HTMLHelper             |
| Class "JPluginHelper" not found    | Joomla\CMS\Plugin\PluginHelper         |
| Class "JModuleHelper" not found    | Joomla\CMS\Helper\ModuleHelper         |
| Class "JModelLegacy" not found     | Joomla\CMS\MVC\Model\BaseDatabaseModel |
| Class "JRegistry" not found        | Joomla\Registry\Registry               |

On PHP 8 the class name is in double quotes. Older forum threads quote it as **Class 'JPlugin' not found** because PHP 7 used single quotes, and they are the same error.

Where the extension lives decides how much of the site goes down. A system plugin loads on every request, front end and administrator alike, so a system plugin calling **JPlugin** takes the whole site and the admin login with it. A module only fails on pages where it is published, but when it does, Joomla replaces the entire page with the error page, not just the module position. **JComponentHelper** errors often come from a component or module helper, which is why they tend to break some pages and leave others working.

## Where the JPlugin error comes from in Joomla 4, 5 and 6

The same message has different causes depending on the Joomla version, and the default settings differ in each. That is why “enable the compat plugin” works for one

person in a forum thread and does nothing for the next.

| Joomla version         | Where <b>JPlugin</b> comes from                              | Default state                                                     |
|------------------------|--------------------------------------------------------------|-------------------------------------------------------------------|
| 4.4                    | <b>libraries/classmap.php</b> , loaded on every request      | Always on. The 4.4 compat plugin is an empty placeholder          |
| 5.0 to 5.4             | Behaviour - Backward Compatibility, option Classes Aliases   | Enabled, Classes Aliases on, for both fresh installs and upgrades |
| 6.x, upgraded from 5.4 | Behaviour - Backward Compatibility 6, option Classes Aliases | Enabled, Classes Aliases on                                       |
| 6.x, fresh install     | Behaviour - Backward Compatibility 6, option Classes Aliases | <b>Plugin disabled, Classes Aliases off</b>                       |

## Joomla 5: someone switched the plugin off

On Joomla 5 the old names only exist while Behaviour - Backward Compatibility is enabled with Classes Aliases on. Most sites hit the error because someone followed the advice to disable the plugin, which is good advice, before checking what depended on it. We covered why leaving it on forever is a mistake in [the compat plugin is a crutch](#).

## Joomla 5.4: the pre-update check for Joomla 6

This is the one most sites will meet next. When you point Joomla Update at Joomla 6, the pre-update check lists "The 'Behaviour - Backward Compatibility' Plugin Must be Disabled" as a required setting and will not offer the upgrade until it is. The Backward Compatibility 6 plugin that 5.4 installs alongside it is, on 5.4, an empty class that listens to nothing and loads nothing. So the moment you disable the old plugin on 5.4, every **JPlugin** , **JFactory** and **JComponentHelper** reference on the site has no alias at all. The [Joomla 5.4 end of support post](#) has the full upgrade order.

## Joomla 6: a fresh install, or a migrated site

An upgraded Joomla 6 site gets Backward Compatibility 6 enabled with Classes Aliases on, so old extensions keep working. A fresh Joomla 6 install ships the same plugin disabled and Classes Aliases off, on the assumption that a new site has no old code. If you built a new Joomla 6 site and then installed an older extension, or migrated content and extensions into one, that is where the error comes from, and enabling the plugin alone does not fix it.

## How do I find which Joomla extension is causing it?

The error names the class, not the extension, so the first job is to get Joomla to print the file path, and when the administrator is down the only way to do that is to edit `configuration.php` by hand. Connect over FTP, SFTP or your host's file manager, open `configuration.php` in the site root, and change these two lines:

```
public $debug = true;
public $error_reporting = 'maximum';
```

Save it and reload the broken page. Under the error message Joomla now prints a table headed **Call stack**, with a Function column and a Location column. The Location column is the list of files PHP passed through on its way to the failure, newest first, and the first path in it that is not core Joomla is your answer: something like `/plugins/system/oldextension/oldextension.php` or `/administrator/components/com_something/helpers/something.php`. The folder name is the extension.

That table only exists when `$debug` is true. Setting `error_reporting` to `maximum` on its own makes PHP display its errors, but Joomla's error page still prints nothing more than the message.

### Use maximum, not development

Joomla 3 called this setting "development". Joomla 4 still accepted that value and converted it to "maximum" when you saved Global Configuration, but Joomla 5 and 6 no

longer recognise it. An unrecognised value is passed straight to PHP's `error_reporting()`, which on PHP 8 throws a type error because "development" is not a number. Copying an old tutorial into `configuration.php` on a Joomla 5 or 6 site gives you a second fatal error on top of the first.

## Ignore the Deprecated lines at the top of the page

The Deprecated lines are PHP warnings about code that will need changing for a future PHP version, and none of them is the cause. Maximum error reporting shows everything PHP has to say, and on PHP 8.4 or 8.5 that means a wall of lines like this before you reach the real error:

```
Deprecated: Using null as an array offset is deprecated, use an empty string instead in /path/to/file.php:123
```

A Deprecated notice does not stop the page. The one above comes from `libraries/src`, which is Joomla core, and a clean Joomla install produces some of these on the newest PHP; a template framework's system plugin can add more. The file paths in them look like evidence, and following them is the easiest way to lose an hour on this error. Scroll past the lot to the `Class "JPlugin" not found` message and look for the Call stack table under it.

## No Call stack table? Your template's error.php is hiding it

When debug is on and the page still shows a short "An error has occurred" message, or a branded "Something went wrong" page, the template is in the way. Joomla renders the active template's own `error.php` if the template has one, and many commercial templates and template frameworks ship an `error.php` that never prints the trace, whatever debug is set to.

Rename `templates/YOUR_TEMPLATE_NAME/error.php` to `error.php.bak` over FTP or your host's file manager and reload. With no template error page, Joomla falls back to `templates/system/error.php`, which prints the Call stack table whenever debug is on. Put the file back once you have the path.

Do not go looking in the log files instead. Joomla catches this error and logs it under the **error** category, but out of the box it only writes the **system** category to **administrator/logs/joomla\_core\_errors.php** . Unless Log Everything is on, or **error** is listed in the custom log categories, there is nothing in **administrator/logs** to find. PHP's own error log does not see it either, because Joomla handled it before PHP could.

## Reading the call stack: a worked example

A [Joomla forum thread](#) from October 2026 runs through every one of these steps on a real site. The owner was on Joomla 5.4.9 and PHP 8.5, had disabled Behaviour - Backward Compatibility because the Joomla 6 pre-update check requires it, and both the front end and the administrator went to **0 Class "JPlugin" not found** . Another forum member gave the standard advice: error reporting to maximum, debug on, and the call stack will name the plugin.

What came back was a page of **Deprecated** lines, first from the T4 template framework's system plugin and then from Joomla core, with no Call stack table anywhere. The owner had by then switched off a long list of components, plugins and modules by hand, from a Droppics image gallery, from My Maps Location and from others, and the site stayed broken.

The reply in the thread was to ignore the Deprecated messages, because the stack trace is the list of files the code passed through and that is the only part that matters, and when no trace shows at all, to delete **/templates/YOUR\_TEMPLATE\_NAME/error.php** so the template stops hiding Joomla's errors. With that file out of the way the trace appeared, and it pointed at **mymaplocations.php** , the system plugin of the My Maps Location extension. That plugin had to be removed or recoded before the site could run without the old class names.

The first debug output was posted at 11:15 and the plugin was named at 11:36, and almost all of those twenty-one minutes went on the Deprecated lines and the missing table.

## Why switching extensions off one by one rarely finds it

The owner in that thread had already switched off parts of the extension that turned out to be the cause, and the site was still down, which is a common result of this approach. Disabling a component in Joomla does not disable the plugins the same extension installed, and one extension can install a lot of them: My Maps Location alone put ten plugins on that site, on top of its modules and component. The system plugin keeps loading on every request however much of the rest you switch off. With the administrator down you are flipping rows in the database and reloading to see what happened, and on a site with more than thirty third-party plugins that is an afternoon's work. The Call stack table names the file on one page load.

With SSH access, a search is quicker than reading traces. This lists every PHP file outside the core that still extends `JPlugin` or calls `JFactory` :

```
grep -rLE "extends JPlugin|JFactory::|JComponentHelper::" --include=*.php
```

Joomla 5 and 6 core files do not use these names, so anything it returns belongs to a third-party extension. It also finds extensions that will break but have not yet, because the page that loads them has not been visited since the plugin went off.

When you are finished, set `$debug` back to `false` and `$error_reporting` back to `'default'` or `'none'` . A live site that prints stack traces is telling every visitor your server paths and extension list.

## Getting the Backward Compatibility plugin back on

Turning the plugin back on is a stopgap, but it gets a broken site back in seconds while you deal with the extension. Use the first method that works for you.

### From the Joomla administrator

If the admin still loads (a broken module or component usually leaves it alone), go to **System > Manage > Plugins**, search for "compat", and enable **Behaviour - Backward**

**Compatibility** on Joomla 5 or **Behaviour - Backward Compatibility 6** on Joomla 6. On Joomla 6, open the plugin and set **Classes Aliases** to Yes as well, then save.

## From mySites.guru, without logging in

Each connected Joomla 5 and 6 site has two audit checks under Extension Information, one per compatibility plugin, each with an on/off toggle. The toggle does not go through the site's normal page load. The mySites.guru connector loads only Joomla's behaviour plugins, never system plugins, and flips the plugin's **enabled** flag with a direct database update, so it should still work when a broken system plugin has taken the front end and the admin down. The same switch is on a single page for every connected site, which is how you check or roll back a batch of sites after an upgrade. The toggle changes the enabled state only, not the Classes Aliases option, so on a fresh Joomla 6 install you still need one of the methods below.

## See the Backward Compatibility 6 plugin on every Joomla site

mySites.guru checks every connected site for this automatically and flags it the moment it appears. It runs on every snapshot, twice a day.

[Check your site for free →](#)

## In phpMyAdmin

Open the site's database in phpMyAdmin (or Adminer, or your host's database tool), click the SQL tab, and run this. Replace **jos\_** with your table prefix, which is the **\$dbprefix** value in **configuration.php** :

```
UPDATE jos_extensions
SET enabled = 1,
    params = JSON_SET(COALESCE(NULLIF(params, ''), '{}'), '$.classes_alias
```

```
WHERE type = 'plugin'
AND folder = 'behaviour'
AND element IN ('compat', 'compat6');
```

It enables whichever compatibility plugins the site has, **compat** on Joomla 5 and **compat6** on Joomla 6, and switches Classes Aliases on in both, leaving the plugins' other options alone. After a Joomla 6 upgrade the old **compat** row no longer exists, because the upgrade uninstalls it, so on Joomla 6 the query only touches **compat6**. If your site has Joomla's System Cache switched on, clear the **cache/com\_plugins** folder afterwards, because Joomla caches the list of enabled plugins.

## With a script uploaded to the site root

If you have FTP but no database tool, upload a small PHP file next to **configuration.php** and open it in a browser. It reads the database details from **configuration.php**, makes the same change as the SQL above, clears the cached plugin list and then deletes itself. Give the file a hard-to-guess name, such as **fix-compat-7f3a9c.php**:

```
<?php
/**
 * Switches Joomla's Backward Compatibility plugins back on, with their
 * Classes Aliases option, then deletes itself.
 *
 * Upload to the folder that holds configuration.php and open it in a browser.
 */
header('Content-Type: text/plain; charset=utf-8');

define('_JEXEC', 1);
require __DIR__ . '/configuration.php';
$config = new JConfig();

// Joomla allows "host", "host:port" or "host:/path/to/mysql.sock"
$host    = $config->host;
$port    = '';
$socket  = '';
```

```

if (preg_match('/^(.+):(\d+)$/', $host, $m)) {
    [, $host, $port] = $m;
} elseif (preg_match('/^(.+):(\/.+)$/', $host, $m)) {
    [, $host, $socket] = $m;
}

if (stripos($config->dbtype, 'pgsql') !== false) {
    $dsn = "pgsql:host=$host;dbname={$config->db}" . ($port ? ";port=$port";
} elseif ($socket) {
    $dsn = "mysql:unix_socket=$socket;dbname={$config->db};charset=utf8mb4";
} else {
    $dsn = "mysql:host=$host;dbname={$config->db};charset=utf8mb4" . ($port ? ";port=$port";
}

try {
    $pdo = new PDO($dsn, $config->user, $config->password, [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]);
} catch (PDOException $e) {
    exit('Could not connect to the database: ' . $e->getMessage() . "\n");
}

$table = $config->dbprefix . 'extensions';
$rows = $pdo->query(
    "SELECT extension_id, element, params FROM $table
    WHERE type = 'plugin' AND folder = 'behaviour' AND element IN ('compatibility', 'backward_compatibility');"
->fetchAll(PDO::FETCH_ASSOC);

if (!$rows) {
    echo "No Backward Compatibility plugin found in $table.\n";
}

$update = $pdo->prepare("UPDATE $table SET enabled = 1, params = ? WHERE extension_id = ?");

foreach ($rows as $row) {
    $params = json_decode($row['params'] ?: '{}', true) ?: [];
    $params['classes_aliases'] = '1';
    $update->execute([json_encode($params), $row['extension_id']]);
    echo "plg_behaviour_{$row['element']}: enabled, Classes Aliases switch\n";
}

```

```
// With System Cache on, Joomla caches the list of enabled plugins
foreach (['/cache/com_plugins', '/administrator/cache/com_plugins'] as $dir) {
    foreach (glob(__DIR__ . $dir . '/*') ?: [] as $file) {
        is_file($file) && unlink($file);
    }
}

echo @unlink(__FILE__)
? "This script has deleted itself.\n"
: "Could not delete this script. Delete it from the server now.\n";
```

Visit <https://example.com/fix-compat-7f3a9c.php> and you should see one line per plugin it switched on, then confirmation that it deleted itself. It does not bootstrap Joomla, so the broken extension never loads while it runs. If the last line says it could not delete itself, delete the file over FTP straight away: a script that writes to your database should not sit in your web root.

If a client site is down right now and you would rather hand it over, [fix.mysites.guru](https://fix.mysites.guru) fixes broken and hacked Joomla sites for a single fixed fee of £120, usually the same day.

We ran both the SQL and the script against a copy of a fresh Joomla 6 extensions table, where Backward Compatibility 6 starts disabled with Classes Aliases off, and each left the plugin enabled with Classes Aliases on and every other plugin untouched.

## Why didn't enabling the Joomla compat plugin fix it?

When the plugin is on and the error stays, the cause is almost always one of these.

The most common is that Classes Aliases is off. On Joomla 6 the plugin has three options, and only Classes Aliases loads the old J-prefixed names. The other two, Include Deprecated Classes and Removed Assets, cover the **Input** and **Filesystem** classes Joomla 6 dropped and some removed JavaScript assets. A fresh Joomla 6 install has Classes Aliases set to No, so enabling the plugin restores the deprecated

classes and none of the aliases. This is the answer to a long run of forum threads where someone on a new Joomla 6.1 site enabled the plugin and nothing changed.

Next is a cached plugin list. With System Cache on, Joomla can keep serving the old list of enabled plugins after a database change. Clearing `cache/com_plugins` fixes it, and the script above does that for you.

Last, the class may have no alias at all, which is the next section.

## Joomla class names no plugin can bring back

Some old names were never in either alias list. They were real classes in Joomla 3, removed in Joomla 4.0, and no version of the Backward Compatibility plugin defines them:

| Error message                                   | What it was                                                          |
|-------------------------------------------------|----------------------------------------------------------------------|
| <code>Class "JDispatcher" not found</code>      | Joomla 2.5's event dispatcher, kept as a deprecated stub in Joomla 3 |
| <code>Class "JEventDispatcher" not found</code> | Joomla 3's event dispatcher                                          |
| <code>Class "JRequest" not found</code>         | Joomla 1.5 to 3 request input, replaced by <code>JInput</code>       |
| <code>Class "JError" not found</code>           | Joomla 1.5 to 3 error handling                                       |
| <code>Class "JApplication" not found</code>     | The legacy application class                                         |
| <code>Class "JDatabase" not found</code>        | The pre-3.0 database class                                           |
| <code>Class "JArrayHelper" not found</code>     | Array utilities, moved to the Joomla Framework                       |
| <code>Class "JString" not found</code>          | String utilities, moved to the Joomla Framework                      |
| <code>Class "JSite" not found</code>            | The old site application alias                                       |

Seeing one of these means the extension was written for Joomla 3 or earlier and has not been touched since, often on a site that was migrated rather than rebuilt. Smart

Slider 2 leftovers, old Search plugins and abandoned gallery extensions are typical sources. There is nothing to switch on, so update the extension to a version written for your Joomla, replace it, or uninstall it. If the site is still on Joomla 3, the [options for staying, patching or moving](#) are laid out on [joomla3security.com](#), and our post on [migrating from Joomla 3](#) covers the extension audit to do first.

## Fixing the Joomla extension for good

The compatibility plugin keeps the site up today. Joomla's own source comments say the Joomla 6 alias list "will likely be removed in Joomla 7.0", and a [change merged into the Joomla 7.0 branch](#) in June 2026 already removes Backward Compatibility 6 in favour of a Backward Compatibility 7. An extension that only runs because of Classes Aliases is due to break again at the next major version.

In order of preference:

1. Update the extension. Most maintained extensions dropped the old names years ago. Check the vendor's site or the [Joomla Extensions Directory](#) for a release that lists your Joomla version, and update to it while the plugin is still on.
2. Ask the vendor. If no update exists, send them the file path from the stack trace and point them at the [Joomla 5.4 to 6.0 compatibility notes](#). The fix is usually a handful of `use` statements.
3. Replace it. An extension still calling `JDispatcher` or `JRequest` in 2026 is abandoned, and abandoned code is a security risk as well as a compatibility one. Our [extension management tools](#) show which sites run it, so you can plan the replacement across all of them.
4. Patch it yourself, as a last resort. For your own code, or to buy time, replace the old names with the modern ones from the first table: `extends JPlugin` becomes `extends \Joomla\CMS\Plugin\CMSPlugin`, `JFactory::getApplication()` becomes `\Joomla\CMS\Factory::getApplication()`, `JComponentHelper::getParams()` becomes `\Joomla\CMS\Component\ComponentHelper::getParams()`. A patched third-party extension is overwritten by its next update, so keep a note of what you changed.

### Stuck at the Joomla 6 pre-update check

If the only thing between a Joomla 5.4 site and Joomla 6 is an extension that breaks without the old plugin, and the vendor has a Joomla 6 release but not a Joomla 5 one, there is a clean way through: disable that extension, disable Behaviour - Backward Compatibility, run the upgrade, then install the vendor's Joomla 6 build. Disabling the extension first means the site keeps working while the old plugin is off. Take a backup before any of it.

## Finding these Joomla extensions before they break a site

Every figure below is from the latest snapshot of each Joomla site in the mySites.guru database over the past 90 days, as of 12 October 2026.

**97.8%**

of Joomla 5 sites still have the old  
Backward Compatibility plugin on  
Joomla 6 will not install until it is off

**84.2%**

of Joomla 6 sites still have Backward  
Compatibility 6 on  
Expected to be removed in Joomla 7

**1.8%**

of Joomla 6 sites still have the Joomla 5  
plugin enabled  
The Joomla 6 upgrade normally uninstalls it

Plan around the first number. Nearly every Joomla 5 site we monitor still runs with the old aliases loaded, so almost none of them have been tested without them yet. When those sites go through the Joomla 6 pre-update check, any site with an extension like this breaks at that moment, usually in the middle of a maintenance window that was booked as a routine upgrade.

Testing without the plugin on staging works for one site. For thirty, it is a week of work. mySites.guru does it in three steps:

- Scan for the old names. On Joomla 4 and 5 sites, the audit check “Check Files That May Need Joomla 5 Compatibility Plugin (Beta)” searches every file on the site for the retired class names and flags the ones that use them, so you can open the list in the file manager and see which extensions they belong to before switching anything off.
- See the plugin state everywhere. The [Joomla 6 Compatibility Checker](#) shows the Backward Compatibility plugin state beside each site’s PHP version, database version and update channel, colour-coded against the [Joomla 6 technical requirements](#).
- Switch it from one place. The toggle on each site’s audit, or the all-sites page for each check, turns either plugin on or off without logging in, and turns it back on if something breaks.

## Find files that still use Joomla's old class names

mySites.guru checks every connected site for this automatically and flags it the moment it appears.

Check your site for free →

All of this is part of every [mySites.guru subscription](#), with unlimited sites. If you are working through a set of Joomla 5.4 sites before support for bug fixes ends, [the Joomla 6.2 post](#) has the rest of the upgrade picture. Check each site’s update channel too, so nothing [jumps to the next major version by accident](#) while you are still clearing extensions.

---

## Further Reading

- [Compatibility plugin, Joomla 4.4 to 5.0](#) - the Joomla 5 Backward Compatibility plugin and its options, from the Joomla manual
- [Compatibility plugin, Joomla 5.4 to 6.0](#) - why Backward Compatibility 6 is required for the upgrade and disabled on new installs
- [Removed backward compatibility, 4.4 to 5.0](#) - every API Joomla 5 dropped from the core
- [Removed backward compatibility, 5.4 to 6.0](#) - the Joomla 6 equivalent
- [The Joomla 6 classmap](#) - the full list of old names and the classes they point at

# Frequently Asked Questions

## What does Class "JPlugin" not found mean in Joomla?

An extension on the site still uses JPlugin, the old name for Joomla's plugin base class, and nothing on the site is providing that name. Joomla 5 and 6 only provide the old J-prefixed names through a Backward Compatibility plugin, so the error appears when that plugin, or its Classes Aliases option, is switched off.

## Which Joomla setting brings JPlugin, JFactory and JComponentHelper back?

On Joomla 5 it is the plugin Behaviour - Backward Compatibility with its Classes Aliases option on. On Joomla 6 it is Behaviour - Backward Compatibility 6, again with Classes Aliases on. A fresh Joomla 6 install ships that plugin disabled and the option off, so both need switching on.

## Can the compatibility plugin fix Class JDispatcher not found?

No. JDispatcher, JEventDispatcher, JRequest, JError, JApplication, JDatabase, JArrayHelper and JString were Joomla 3 classes removed in Joomla 4.0, and neither Backward Compatibility plugin has an alias for them. The extension has to be updated or replaced.

## Should error reporting be set to development to see the full error?

Not on Joomla 5 or 6. Development was a Joomla 3 value; Joomla 4 converted it to maximum when you saved Global Configuration, and Joomla 5 and 6 no longer recognise it at all. Set error\_reporting to maximum and debug to true in configuration.php.

## Why can't I see a stack trace with Joomla debug switched on?

Usually because the active template has its own error.php, and it does not print one. Rename templates/YOUR\_TEMPLATE\_NAME/error.php so Joomla falls back to templates/system/error.php, which prints a Call stack table whenever debug is true, then put the file back afterwards. Deprecated lines at the top of the page are PHP warnings and not the trace, so scroll past them.

## Why does the error not appear in the Joomla log files?

Joomla catches the error and renders its error page, and by default only the system category is written to joomla\_core\_errors.php. The class-not-found error is logged under the error category, so it only reaches a file if Log Everything is on or error is listed in the custom log categories.

### Do I have to fix these extensions before upgrading to Joomla 6?

Yes. Joomla 5.4's pre-update check will not offer the Joomla 6 upgrade until Behaviour - Backward Compatibility is disabled, and the Backward Compatibility 6 plugin on 5.4 is an empty placeholder that provides nothing. Any extension that needs the old class names breaks the moment you switch the old plugin off.

**AI MODIFIED**

Written and edited by a human, with AI assistance. [Our approach to AI](#)

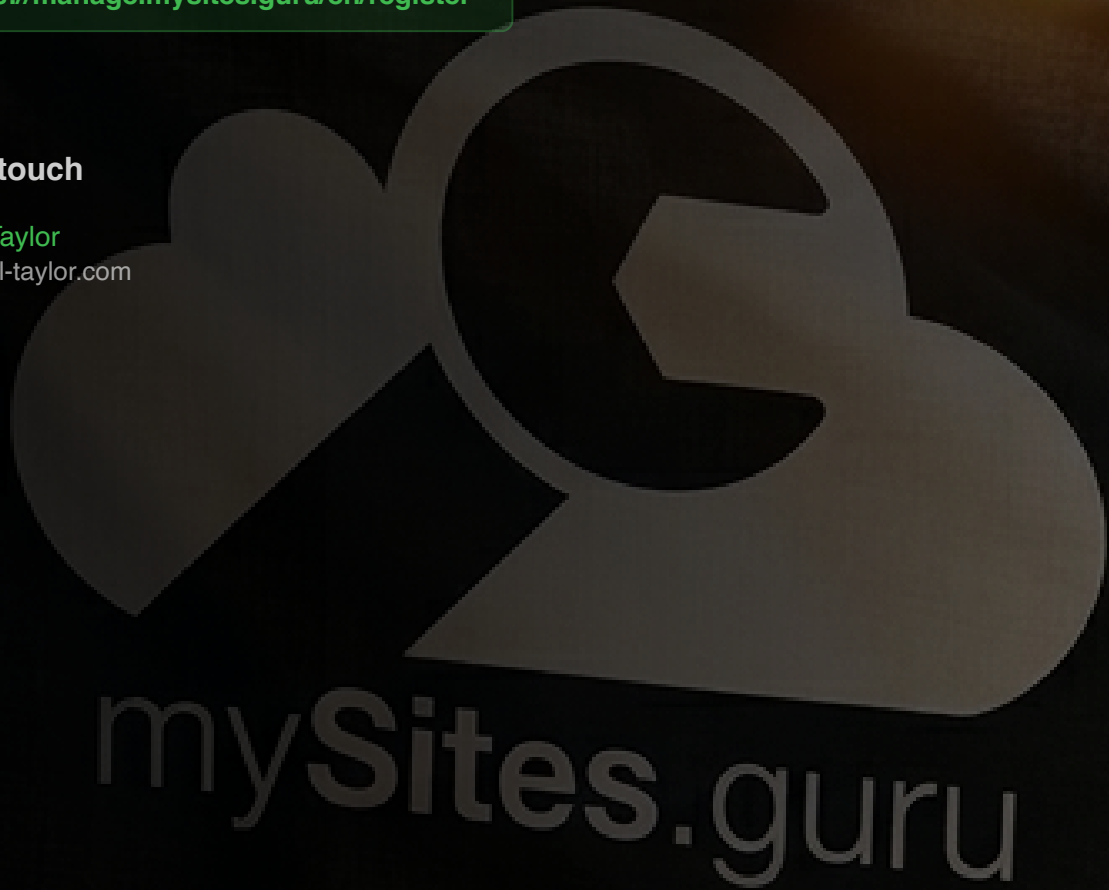
# Get Your Free Site Audit

See how your WordPress and Joomla sites measure up.  
No credit card required.

<https://manage.mysites.guru/en/register>

## Get in touch

Phil E. Taylor  
[phil@phil-taylor.com](mailto:phil@phil-taylor.com)



© 2026 Blue Flame Digital Solutions Limited. All rights reserved.

[mysites.guru](https://mysites.guru)

