How to Turn On Module Versioning in Joomla 6.1

Joomla 6.1 finally brings version history to modules. How to enable save_history manually, and how mySites.guru flips it on across every site.

Phil E. Taylor | 28 August 2026

Joomla 6.1 shipped yesterday with one feature that will feel overdue to anyone who has ever lost a module edit: version history for modules. Articles have had this since Joomla 3.2 in 2013. Menu items, categories, tags, contacts, newsfeeds, and user notes picked it up along the way. Modules were the glaring hole in the list. Joomla 6.1.0 finally closes it.

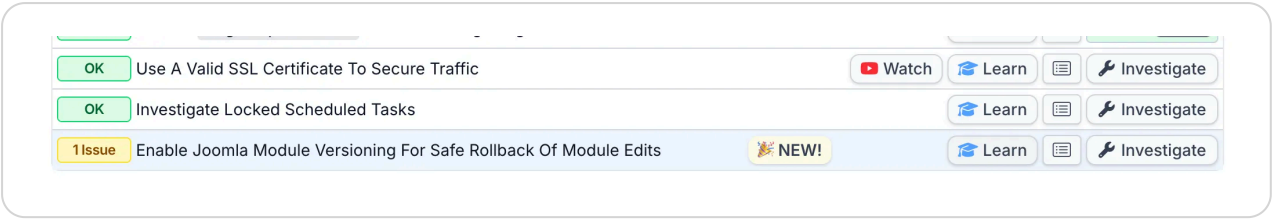
If you manage Joomla sites for clients, you have almost certainly been bitten by this. A typo in a custom HTML module, a bad class name on a menu module, a well-meaning client who "just tweaked the footer" and took their own site offline. Without versioning, the only way back is your last backup, or opening the module side by side with a staging copy and reconstructing the old content by hand.

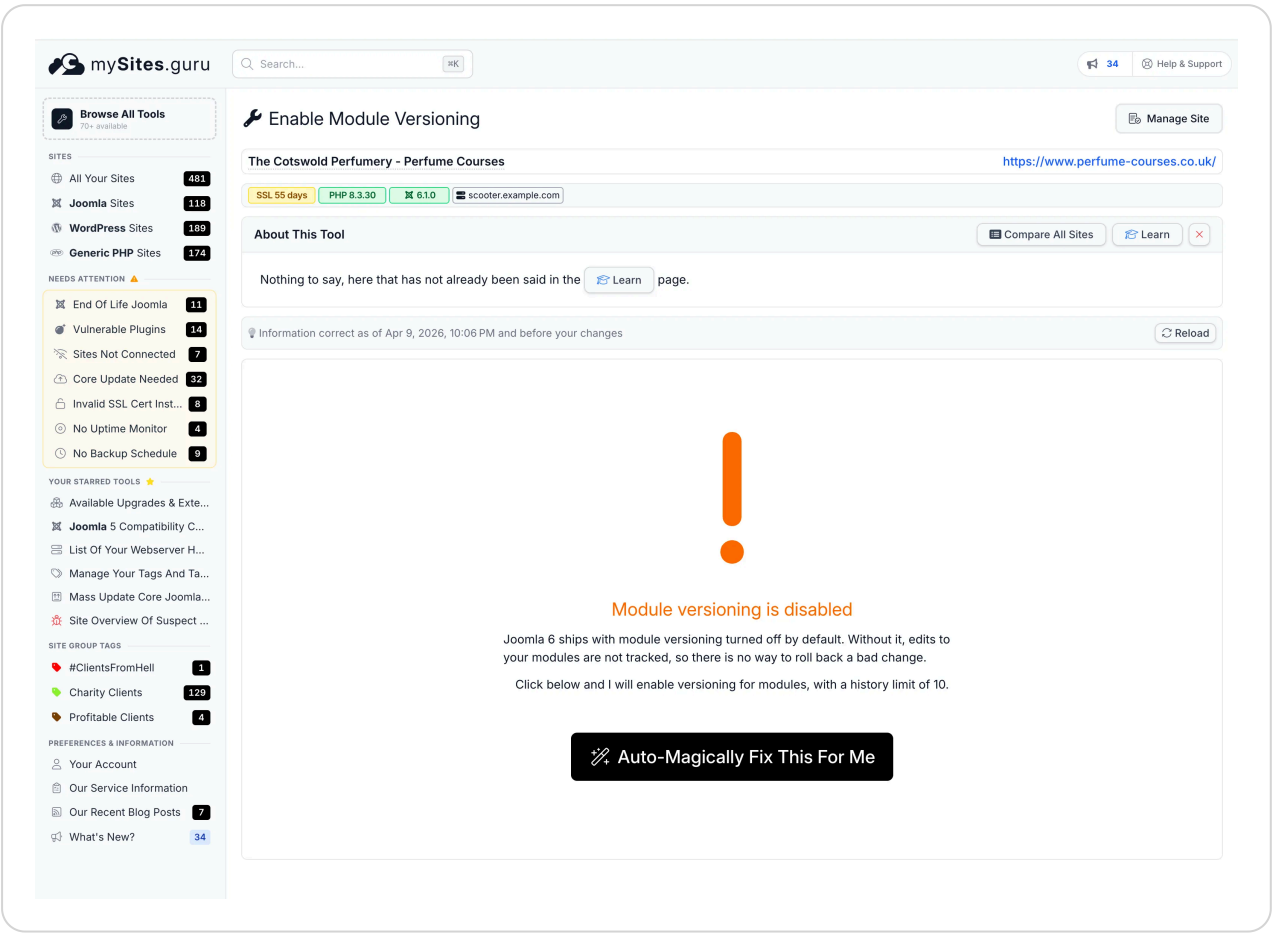
In 6.1, once you turn the feature on, every module save is captured. Open the Versions panel while editing and you get the same diff-and-restore UI you already use for articles. One click and the module is back.

How mySites.guru Enables Joomla Module Versioning on Every Site at Once

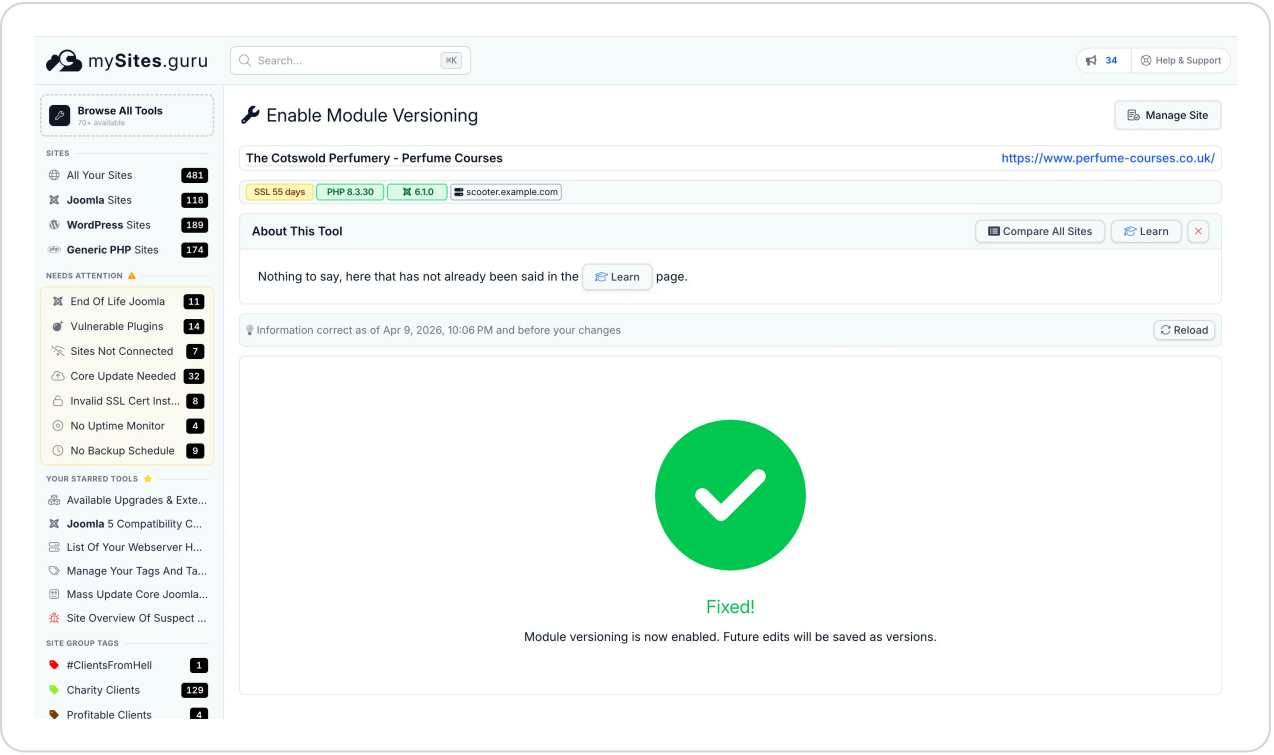
Joomla 6.1 ships with module versioning turned off. On a single site that is a two-click fix. On thirty client sites it is thirty logins, thirty Options dialogs, thirty Save clicks, and thirty reasons to put it off until next week.

mySites.guru now ships a new audit tool called **Enable Joomla Module Versioning For Safe Rollback Of Module Edits** that runs on every Joomla 6.1 snapshot. It reads the `com_modules` component params, checks whether `save_history` is set to `1`, and flags any site where it is not. The fix is a single click: the connector writes `save_history = 1` and sets a sensible `history_limit` of 10 if one is not already configured, directly to the `#__extensions` table. No admin login. No browser tab juggling.

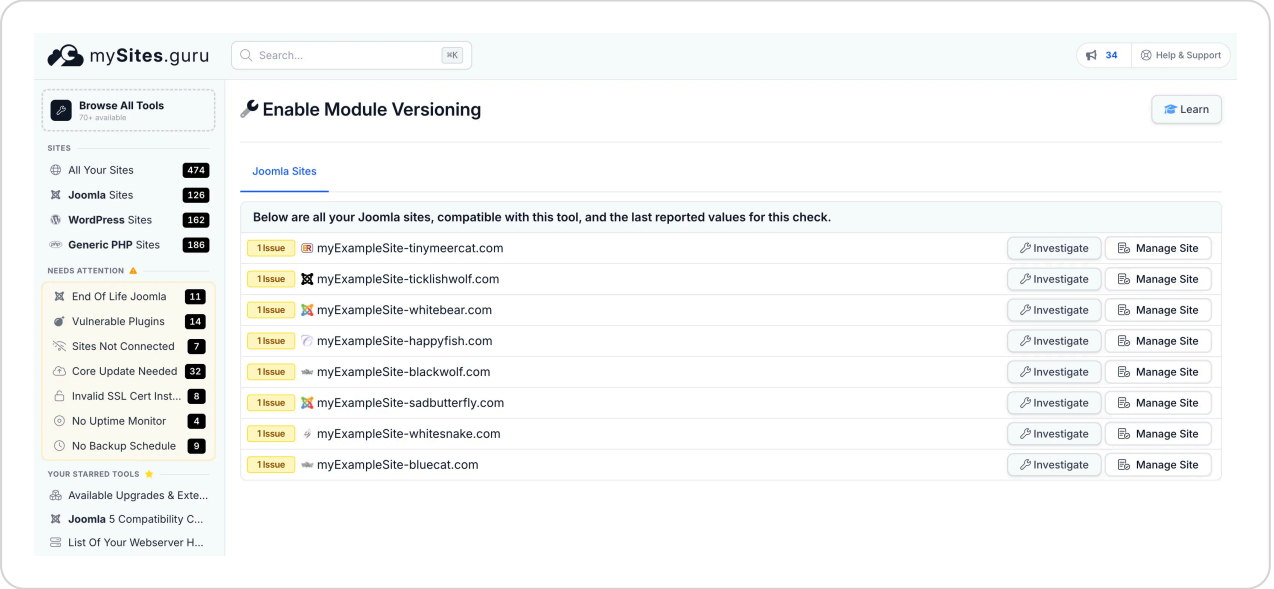




One click on **Auto-Magically Fix This For Me** and the same check comes back green:



From the all-sites tool view you can see the module versioning status for every connected Joomla 6.1 site in one table and toggle on the ones that are still off. If you manage multiple Joomla sites, this is thirty minutes of busywork collapsed into one button.



The audit tool only runs on Joomla 6.1 and later, since the feature itself does not exist on older versions. Older sites are simply skipped in the check.

How Do You Enable Joomla Module Versioning Manually in Joomla 6.1?

If you only have one Joomla site, or you want to understand what the setting does before pushing it out across your portfolio, it is a quick fix:

1. Log into the Joomla 6.1 admin.
2. Go to **Components**, then **Modules**.
3. Click the **Options** button in the top right of the toolbar.
4. On the **Component** tab, set **Save History** to **Yes**.
5. Leave **Maximum Versions** at the default of `10`, or bump it higher if you edit modules frequently and want a longer rollback window.
6. Click **Save and Close**.

That is it. From this point on, every module save is tracked.

One thing that trips people up: the setting is on the **component** level, not in **System → Global Configuration**. Joomla has a Global Configuration panel that covers site-wide settings like the site name and session lifetime. Component-specific settings like this one live on each component's own Options screen, reached through the Components menu. Some older tutorials for content history refer to Global Configuration, but for modules in 6.1 the field is specifically in the com_modules component config.

You can also reach the same setting through **Extensions → Modules → Options**. It is the same underlying config row in the database, so changing it in either location applies everywhere.

What Does Joomla 6.1 Module Versioning Actually Do Under the Hood?

The implementation is smaller than you might expect. Robert Deutz's pull request #46772 adds just 78 lines of production code. It works by:

1. **Adding modules as content type id 14** in the `#__content_types` table. This is the table the `com_contenthistory` system uses to know what kinds of items can be versioned. Pre-6.1 there were 13 types (articles, categories, contacts, tags, and so on). 6.1 adds `com_modules.module` as id 14.
2. **Flipping the save_history switch** in `com_modules` component params. When save_history is 1, the existing contenthistory hook captures every module save.
3. **Adding a history_limit field** in the same component params, defaulting to 10. This tells contenthistory how many versions to keep per module before pruning old ones.
4. **Adding a version_note text field** to the module edit form. When you save a module, you can optionally leave a short note describing the change. The note appears in the Versions list, so future-you (or whoever has to roll back) can see “fixed client logo width” instead of a timestamp.

Everything else already existed. The diff UI, the restore button, the Keep Forever flag, the `#__ucm_history` storage table - all of it was built for articles years ago and has been reused for every item type added since. Wiring modules into the same system is a few lines of XML and a content type row. The fact that it took twelve years to get here is more a story about how much is already on the Joomla project’s plate than about the difficulty of the feature.

> The Audit Trail Is the Quieter Benefit
> The rollback safety net is the obvious win, but the audit trail is the quieter benefit. Every version row records who made the change, when, and optionally why. On a shared admin account that is not great, but on a properly configured site with per-user Joomla logins, it becomes a simple answer to “who broke the homepage module last Tuesday.”

What Did Joomla Module Versioning Look Like Before Joomla 6.1?

If you have been on Joomla for a while, you know the workarounds. Most of them were ugly.

Third-Party Extensions

The Module Version extension by RickR2H has lived on the JED for years. It does roughly what 6.1 now does natively. Respectable piece of work, and if you are still on 5.x or earlier it is still the best option. From 6.1 forward you can drop it.

Snapshot Plugins

A pile of third-party backup and audit tools save module configurations on a schedule. That is backup, not version history. It works if you notice the problem within the backup window, not if a client discovers the bad edit three weeks later.

Manual Copy-Paste in Custom HTML Modules

The field hack: before every edit, copy the existing HTML into a text file with today's date. Works on one module. Does not work on thirty.

Staging Sites

A proper staging workflow solves this, but every small client edit through a staging cycle is its own kind of overhead. For a "just swap the phone number in the header module" change, most agencies skip staging and take the risk.

Native 6.1 versioning kills every one of these workarounds for the specific case of module edits.

Are There Things That Joomla Module Versioning in 6.1 Does Not Cover?

Three caveats worth knowing before you sell this feature to a client.

Frontend Editing Is Not Tracked

The 6.1.0 implementation is admin-only. If a user edits a module through the frontend editing UI, that save does not hit the contenthistory hook and does not create a version row. This is a known follow-up, raised by Brian Teeman in the PR review. Expect frontend support in a future 6.x release. For now, treat frontend module edits as unversioned, and steer important changes through the admin panel where possible.

Existing Modules Have No History Until You Enable the Feature

Versioning starts recording at the moment you flip save_history on. It is not retroactive. If you turn it on today and a module was broken last week, last week's good state is not in the history. This is why getting it enabled across your portfolio early is worth doing: every day you leave it off is a day of potential rollbacks you will never have.

The Versions Button Only Appears When com_contenthistory Is Enabled

On a default Joomla 6.1 install it is enabled. If you have been through a migration or disabled it for some reason, the Versions UI will not appear in the module edit toolbar even with save_history on. Re-enable com_contenthistory under Extensions → Plugins.

How Does mySites.guru Enable Module Versioning in Joomla 6.1?

For the curious, here is what happens when you click Fix This For Me on the Enable Module Versioning check.

The mySites.guru connector sits on each managed site as a lightweight PHP bridge. When the dashboard sends a fix request for module versioning, the connector:

1. Reads the current `params` JSON from the `#__extensions` row where `name = 'com_modules'` and `type = 'component'`.

2. Parses the JSON, or falls back to an empty object if the params are missing or malformed.
3. Sets `save_history` to `'1'`.
4. Sets `history_limit` to `'10'` if it is not already a positive integer (respecting a value you set manually).
5. Writes the updated params back to the `#__extensions` table in a single UPDATE query.
6. Reads back the saved value and returns it to the dashboard so the next snapshot reflects the change.

No filesystem writes. No Joomla admin session. No reliance on the web UI. It is the same mechanism we use for every other Joomla audit and fix tool, and it scales linearly: enabling versioning on 100 sites takes roughly as long as enabling it on 1.

Frequently Asked Questions

Does Enabling Module Versioning Slow Down the Admin?

No. The contenthistory hook is a lightweight insert into a single table on module save. There is no performance impact on admin page loads or frontend rendering. The only time you notice it is when you open the Versions panel, and even then it is a simple query against `#__ucm_history` filtered by item id.

Can I Set a Different history_limit Per Module?

Not in 6.1. The history_limit is set at the component level and applies to every module saved through com_modules. A higher per-module limit may come later, but for now set the component-level limit to the highest value you want for any module, and rely on the Keep Forever flag on individual versions to preserve the specific states you never want pruned.

What Happens to Versions When I Delete a Module?

The contenthistory rows for the deleted module stay in `#__ucm_history` by default. If you restore the module from the trash, the history is still there. If you permanently delete the module, the history rows become orphaned but do not cause harm. Joomla has no built-in cleanup for orphaned history rows, so a very long-lived site may accumulate some. It is a non-issue on any reasonably young site.

Does Module Versioning Apply to Menu Modules?

Yes. Menu modules are just modules from com_modules' perspective. Once save_history is on, menu module edits are tracked like any other. This is separate from the existing menu item versioning, which tracks the menu items themselves.

Further Reading

- PR #46772: Versions for Modules - Robert Deutz's pull request with the full implementation, reviewer discussion, and testing instructions.
- Mastering Joomla Content Versioning - Brian Teeman's October 2025 Joomla Community Magazine piece on content versioning. Covers articles specifically, but the Versions UI is identical for modules in 6.1.
- Joomla 6.0 to 6.1 migration guide - official developer notes for the 6.1 release.
- How to enable POW captcha in Joomla 6.1 - the other 6.1 feature worth turning on across your portfolio: a built-in proof-of-work captcha that replaces Google reCAPTCHA.
- Using Content History in your Component - docs on how the com_contenthistory system works under the hood, useful if you develop custom components.

Running a Joomla agency and thinking about 6.1 upgrades? The Joomla Agency Handbook covers update strategy, version management, and multi-site workflows. If you just want to know whether your sites are ready, the mySites.guru Joomla 6

compatibility tool checks PHP, database, and update channel across every connected site in one view.

Frequently Asked Questions

What is module versioning in Joomla 6.1?

Module versioning is a new feature in Joomla 6.1.0, released on 14 April 2026. When enabled, every edit to a module is saved as a version in the com_contenthistory system. You can open the Versions panel while editing a module, compare revisions with a green and red diff, and restore any previous version with one click. This is the same versioning UI that articles, menu items, and tags have had for years, now wired up to com_modules for the first time.

Why is module versioning turned off by default in Joomla 6.1?

The save_history option in com_modules defaults to No, matching the default for most other components that support versioning. The Joomla project treats history tracking as opt-in per component because each saved version writes a row to the #__ucm_history table, which grows the database over time. On sites with hundreds of modules edited often, you may prefer a lower history limit or leave it disabled. For most agency-managed sites the overhead is tiny and the rollback safety net is well worth it.

Where is the setting to enable module versions in Joomla 6.1?

In the Joomla 6.1 admin, go to Components, then Modules, then click the Options button. On the Component tab you will see Save History (Yes or No) and Maximum Versions (default 10). This is the per-component config for com_modules, not the System Global Configuration screen. The same setting also lives in Extensions, then Modules, behind the same Options button, and changes there apply globally.

Does module versioning work with third-party modules?

Yes. The Joomla 6.1 implementation hooks into the base JTable store event that all modules use, so any module type - core or third-party - gets versioning as soon as save_history is enabled. That is confirmed in Robert Deutz's pull request testing instructions, which explicitly call out testing with third-party modules. You do not need the module developer to add anything.

Does module versioning cover frontend editing?

Not at 6.1.0. The initial implementation is admin-only. Edits to modules made through the frontend editing UI are not tracked in the version history. Brian Teeman raised this in the PR review and it is a known follow-up. Expect frontend coverage in a later 6.x release. For now,

if you rely on frontend module edits, make sure the important modules are edited from the admin so they land in the version history.

How much database overhead does module versioning add?

Every module save writes one row to #__ucm_history with the serialized module row data. With the default history_limit of 10, you will never keep more than 10 versions per module, so the table stays bounded. A site with 200 modules edited occasionally will add a few hundred rows. The contenthistory system also prunes old versions when you save past the limit. You can lower history_limit per component if space is a concern, but on any reasonably sized site the overhead is noise.

How does mySites.guru enable module versioning in Joomla 6.1?

mySites.guru's connector writes directly to the com_modules component params in the #__extensions table, setting save_history to 1 and history_limit to 10 if it is not already set. The change takes effect immediately with no Joomla admin login required. The new Enable Module Versioning audit tool runs on every snapshot of a Joomla 6.1 site, flags any site where versioning is off, and offers a Fix This For Me button to turn it on in one click. From the all-sites tool view you can enable it everywhere at once.

Can I roll back a module edit in Joomla 6.1 after enabling versioning?

Yes, once save_history is on, any future module edit creates a version row. Open the module for editing, click the Versions button in the toolbar, and you will see a list of saved revisions with date, author, and an optional version note. You can preview any version, compare two versions side by side with a diff, restore a version with one click, or mark a specific version as Keep Forever so it survives past the history_limit cap. There is no retroactive rollback: versions only start saving from the point you enable the setting, so turn it on before your next batch of client work.

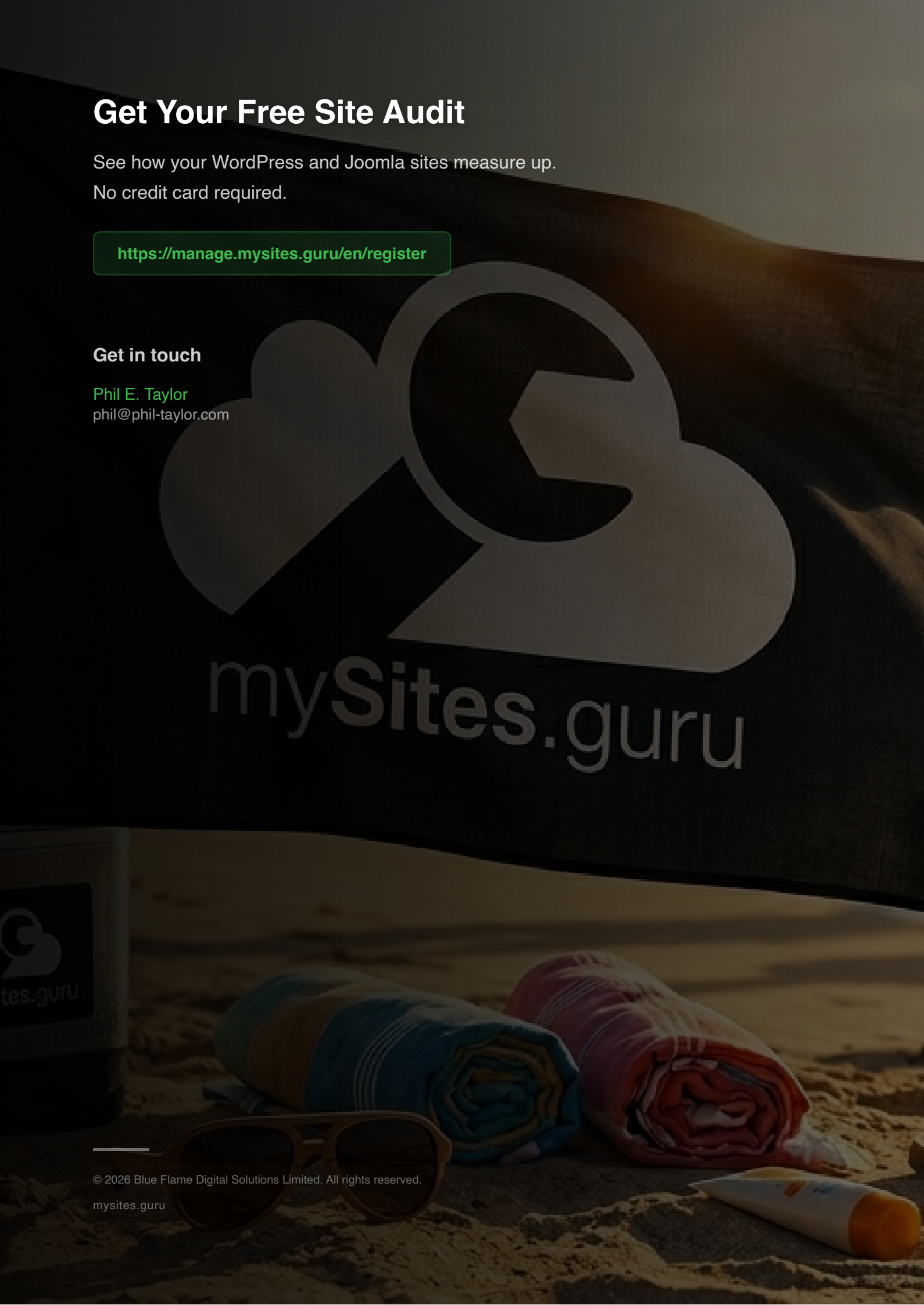
Written and edited by a human, with AI assistance. Our approach to AI

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

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