

How to upgrade to Tiki Wiki 27 (new Build System)

👤 Bernard Sfez - 2026-02-24 11:53



The Tiki 27+ Build System marks a significant evolution in the way Tiki Wiki CMS Groupware handles its development and deployment processes. This article aims to guide you through the essential steps and provide solutions, even as the process continues to stabilize.

Introduced in Tiki 27 and refined in subsequent versions, this build system brings numerous enhancements aimed at improving efficiency, consistency, and ease of maintenance for developers and users alike. However, it requires power users and administrators to familiarize themselves with the new installation method from Gitlab (Version Control System). This guide details the updated Tiki system build and setup procedure leveraging nvm for Node version management and npm for dependency handling, including troubleshooting for frequent installation and runtime issues.

As Tiki 27 and its installation process stabilize, this guide is an ongoing work and will be used to update the [https://doc.tiki.org|Official]([https://doc.tiki.org|Official]) Tiki Documentation[blank].

Tiki Wiki new System Build

The Tiki 27 plus Build System leverages modern development tools and practices, aligning Tiki with contemporary web development standards. This includes the integration of tools like Composer for PHP dependencies and Node.js for JavaScript and CSS dependencies. It modernizes the development workflow and eases the setup of several components required for Tiki Wiki. It also ensures consistency across different environments (development, staging, production). The automated setup process minimizes the risk of environment-specific issues, leading to more reliable deployments.

At the time of this article, the Tiki Wiki 27 plus Build System will mainly set up:

- Composer
- Node.js and npm
- ...

Global Installation and preliminaries to run the Tiki 27+ Build System

Ensure you have the necessary requirements installed, such as PHP, Composer, Node.js, and npm. Follow the Tiki documentation to install and configure these tools. You can download and install Node.js and npm from the official Node.js website.

Cloning and refreshing your code

Clone the Tiki repository from the Tiki Gitlab repository to your local development environment. This will provide you with the latest version of the Tiki codebase.

For more details about installation, check our tutorial on how to install Tiki Wiki. Explore our tutorials the articles we provide to support Tiki Wiki and check if a more recent articles has been published.

Run setup.sh

It is highly recommended you run this script using the same user as your Tiki Wiki web directory (e.g., `www-data`). Use `root` only as a last resort if you have an issue.

Tiki Wiki provides a script to automate the setup process and apply correct permissions.

Run the following command in your project directory:

If you have one PHP version

```
sh setup.sh
```

If you have several PHP versions installed, it is critical to select the proper PHP version (see Tiki Wiki requirements) to run the `setup.sh` script.

Here is a sample on Debian 12 using a Virtualmin LAMP setup:

If you have several PHP versions

```
sh setup.sh -p /usr/bin/php8.2
```

You will use the option `"b"` (build) and then `"f"` to set permissions.

The `"b"` option will install Composer packages and PHP dependencies, and npm will install JavaScript and CSS dependencies. You may see several warnings (it is still a work in progress), but only **errors** require your attention.

If all goes well, use the `"f"` option to apply permissions and ownership over your Tiki files and directories.

Compiling Your Theme and Public Generated Files

With the Tiki 27+ Build System, compiling your theme SCSS and CSS files is different from previous versions. Tiki 27+ uses a different directory and file pattern, and it relies on public generated files to ensure your Tiki instance looks and functions as intended. This involves processing your custom stylesheets, scripts, and other assets to make them ready for production use.

The last step of the script execution for `setup.sh` will run npm to complete the installation. You will need to `run npm run watch` each time you modify your custom CSS theme, templates, custom files, or custom language files to ensure all your public files are up-to-date.

There are changes and new location for custom theme top file. Consult the regularly updated Tiki Theme documentation for more information.

Troubleshooting

Clean rebuild (from scratch)

In case of issues—especially when logs aren't conclusive—it's a good reflex to reinstall packages and dependencies from scratch.

Manually remove previous directories and caches, then re-run the setup:

Manual install from scratch

```
rm -rf vendor_bundled/vendor
rm -rf node_modules
rm -f temp/cache/container.php
sh setup.sh -n build
```

Notes

- Each Tiki version is released with its own `package-lock.json` file and its own `composer.lock` file. It is important to keep these files and replace yours with the ones provided by the Tiki Wiki release team when you upgrade.

Otherwise, you will end up with a completely unsupported Tiki installation. These lock files are the safety net that guarantees everyone is using the exact same package versions.

- Note: The command `sh setup.sh -n build` does not remove existing dependencies by itself; the manual removals above ensure a pristine rebuild.

Your Theme CSS is Not Compiling Correctly

As the new Tiki Build System is still under ongoing development, occasional hiccups may occur, and you might encounter issues where your custom theme CSS stylesheet does not update correctly after running the `npm run watch` command, your site looks broken, or themes and template changes are not applied.

Steps to Resolve the Issue

- First, try a clean rebuild (see the section above).
- Confirm that your changes are present in the custom SCSS file:
Path: `_custom/shared/themes/yourtheme/css/yourtheme.scss`
- Next, verify that the changes appear in the generated CSS file:
Path: `_custom/shared/themes/yourtheme/css/yourtheme.css`

If your styles are in the `.scss` file but missing from the `.css` file, it is likely that npm has not detected the file changes.

Additional Tip

Sometimes, you may need to open a second terminal or shell:

- Run the `npm run watch` command in the first terminal.
- Force the SCSS file to recompile
To resolve this, you can "touch" the SCSS file, which updates the file's timestamp and forces the npm watch process to recompile the SCSS:

Touch scss file

```
touch _custom/shared/themes/yourtheme/css/yourtheme.scss
```

root or not root

Each developer has their own ways and preferences, and various environments can complicate the setup process. While we recommend never using root, there are cases where you might find yourself stuck.

In systems where you cannot modify the sudoers list or where the cost of making changes is too high, you may have to use root to install the correct npm and Node.js versions, as well as to run the command `npm run watch`.

Required version npm and node

Node.js and npm (Node Package Manager) are essential for installing and running Tiki Wiki 27 and later versions due to several key reasons related to modern web development practices.

As of the date of this article, Tiki Wiki requires the following versions of Node.js and npm to be installed on your system:

For Tiki27 (check package.json for your Tiki version)

- npm: Version 9.0.0 to 11
- Node.js: Version 18.0.0 to 22

To check what is installed on your server and to switch use the following

check nvm version, switch version

```
nvm use  
nvm use xx
```

These information are (were) valid for Tiki Wiki 27. You can find these exact versions you need for your Tiki version requirements specified in the `package.json` file.

If your packages are not within these version ranges, `setup.sh` and other npm commands such as `npm run watch` will fail.

Sample process used to install npm and Node.js

To upgrade or downgrade your npm and Node.js versions, visit [Node.js Downloads](#). Use the dropdown selector to choose the Node.js version suitable for your environment.

1. Follow the provided instructions to install the selected Node.js version.
2. Once installed, you should be ready to complete your Tiki installation.

Fixing generateiconlist command

When running the Tiki build system command (`sh setup.sh`) the process may be interrupted by an error while trying to execute the PHP command `build:generateiconlist`.

You may also try to run directly `/Applications/MAMP/bin/php/php8.2.20/bin/php console.php build:generateiconlist` and have the following error:

Failed to open the bootstrap css vendor file, bootstrap icons won't be synched. Maybe npm is not available at this stage.

To solve this issue I deleted "package-lock.json" and the "node-modules" directory.

Then I ran "npm i" (I also ran "npm audit fix --force" but it seems optional)

Then I launched the build system (b) command on `sh setup` or simply "npm run build" and it should complete as expected.

Fix System limit for number of file watchers reached

Linux uses the `inotify` package to observe filesystem events for individual files or directories. When working with projects that involve monitoring many files, such as Tiki Wiki, you may encounter warnings related to the `inotify` watch limit being reached. Increasing this limit can resolve these warnings.

To address this issue, follow the discussion on [Stackoverflow](#) and apply the solution that fits your specific case and environment.

I hope this information proves helpful for now. As Tiki Wiki 27 and its installation process stabilize, this guide will be used as a complement support the Tiki Documentation.