

How to Disable OpenCode Auto-Updates and Fix Database Corruption Issues

TechRounder PDF Edition

Live article:

<https://www.techrounder.com/how-to/how-to-disable-opencode-auto-updates-and-fix-database-corruption-issues/>

By Vipin PG | Published February 17, 2026 | Updated March 9, 2026 | Format: Guide | 3 min read

Quick answer

To stop OpenCode from automatically updating and potentially corrupting your development environment, the most reliable method is setting the system-level environment variable `OPENCODE_DISABLE_AUTOUPDATE=true`. Alternatively, users can modify the `opencode.json` configuration file to set "autoupdate" to false, though this is less robust, especially when OpenCode was installed via package managers like Homebrew or npm.

Key points

- Recommended Fix (Environment Variable): The most bulletproof way to disable updates is by adding 'export OPENCODE_DISABLE_AUTOUPDATE=true' to your shell profile (macOS/Linux) or using PowerShell on Windows to set the variable globally.
- Alternative Fix (Config File): You can edit the 'opencode.json' file (located in '~/.config/opencode/') by setting "autoupdate": false' or "autoupdate": "notify" to get alerts without installing.
- Package Manager Conflicts: If you installed OpenCode via Homebrew, npm, or Bun, use the environment variable method to prevent the internal updater from fighting with your system package manager.
- Advanced Lockdown: For strict stability, you can use additional variables like 'OPENCODE_DISABLE_LSP_DOWNLOAD=true' (stops language server downloads) or 'OPENCODE_DISABLE_MODELS_FETCH=true' (stops remote model queries).
- Important Warning: Do not mix installation methods (e.g., using curl scripts alongside Homebrew) as conflicting binaries often ignore configuration files.

If you've ever fired up OpenCode to work on a project only to find your database corrupted or your disk space mysteriously vanishing, you know how frustrating it can be. This often happens because OpenCode, by default, aggressively auto-updates itself in the background to keep up with the latest AI models. The good news is that you can take back control of your development environment with a few simple configuration changes.

Method 1: The Environment Variable (Recommended Fix)

While there is a configuration file setting for this, it can sometimes be unreliable or overridden by other processes. The most robust, "bulletproof" method to stop OpenCode from updating itself is using a system-level environment variable.

macOS and Linux

You need to add the variable to your shell profile script.

- Open your terminal and edit your profile (e.g., '~/.zshrc' or '~/.bashrc').
- Add the following line to the bottom of the file: `export OPENCODE_DISABLE_AUTOUPDATE=true`
- Save the file and run 'source ~/.zshrc' (or the file you edited) to apply the changes immediately.

Windows

You can set this globally using PowerShell.

- Press Win + R , type 'powershell' , and press Ctrl + Shift + Enter to run as Administrator.
- Run the following command: '[Environment]::SetEnvironmentVariable('OPENCODE_DISABLE_AUTOUPDATE', 'true', 'User')
- Restart your terminal or PC for the change to take full effect.

Method 2: The Configuration File

If you prefer not to mess with environment variables, you can edit OpenCode's JSON configuration file. Note that this method is slightly less reliable than Method 1 if you installed OpenCode via a package manager like Homebrew.

How to Edit the Config

1. Locate your configuration file.

- macOS/Linux: '~/.config/opencode/opencode.json'
- Windows: '%USERPROFILE%.config\opencode\opencode.json'

2. Open the file in your text editor.

3. Add or modify the autoupdate key.

To completely disable updates:

```
{  
  "autoupdate": false  
}
```

To get notifications but block automatic installation:

```
{  
  "autoupdate": "notify"  
}
```

Specific Fixes for Package Managers

If you installed OpenCode using a package manager (like Homebrew, npm, or Bun), the internal updater often fights with the system manager. This causes version mismatches where the CLI reports one version but the TUI reports another.

- The Fix: Use Method 1 (Environment Variable).
- Why: This forces OpenCode to yield control back to your package manager. You will only receive updates when you manually run commands like 'brew upgrade opencode' or 'npm update -g opencode-ai' .

Advanced: Locking Down Your Environment

For enterprise environments or strict stability, you might want to disable other automated behaviors to create a "static" tool. You can add these additional variables to your environment just like in Method 1:

- OPENCODE_DISABLE_LSP_DOWNLOAD=true Prevents the tool from untrusted downloading of Language Server Protocols (helps if you satisfy dependencies manually).
- OPENCODE_DISABLE_MODELS_FETCH=true Stops the tool from constantly querying remote providers for new model definitions.

- `OPENCODE_DISABLE_DEFAULT_PLUGINS=true` Prevents automated plugin fetching, useful for debugging crashes.
- `OPENCODE_DISABLE_TERMINAL_TITLE=true` Stops OpenCode from renaming your terminal tabs or windows.

What NOT to Do

- Don't ignore database corruption errors: If you see schema mismatch errors, stop all instances immediately. You may need to delete the `./global/storage/` directory to reset the database.
- Don't mix installation methods: Avoid installing via `'curl'` scripts if you already removed a Homebrew version without a full cleanup. Conflicting binaries often ignore configuration files.

Conclusion

Taking control of OpenCode's update cycle is essential for a stable workflow. While the auto-update feature keeps you on the bleeding edge, it causes too much instability for professional, multi-terminal work. We strongly recommend setting `OPENCODE_DISABLE_AUTOUPDATE=true` in your environment variables as the primary fix. This ensures your tools stay deterministic and only update when you are ready.