

Fixing the 'No Longer Eligible as Formatter' Error in Prettier for VS Code

TechRounder PDF Edition

Live article:

<https://www.techrounder.com/how-to/fixing-the-no-longer-eligible-as-formatter-error-in-prettier-for-vs-code-1769475997/>

By Vipin PG | Published January 27, 2026 | Updated March 9, 2026 | Format: Comparison | 4 min read

Bottom line

To fix Prettier's "no longer eligible as formatter" error in VS Code, make sure Prettier is installed in your project, then set "editor.defaultFormatter": "esbenp.prettier-vscode" in both your global settings and `.vscode/settings.json` (including `[json]` and `[markdown]`). If JSON/MD still won't format, add explicit `files.associations` for `*.json` and `*.md` .

Key points

The article explains that the "No longer eligible as formatter for many types of files" message in the VS Code Prettier extension usually means Prettier has stopped being registered as the default formatter for certain file types, often after an extension update. It calls out common causes like conflicts with other formatting tools, changes in Prettier's configuration, mismatches between workspace and global VS Code settings, and even version compatibility issues with Node.js or VS Code. In the author's case, the problem started after updating to a newer Prettier version while using a custom `.prettierrc`, and it showed up specifically when trying to format JSON and Markdown files that previously worked. The suggested fix begins by verifying Prettier is actually installed correctly in the project by running `npm list prettier` in the project terminal to confirm which version you're using.

If you've encountered the "No longer eligible as formatter for many types of files" error in the Prettier extension for Visual Studio Code, you're not alone. This issue can be frustrating, especially if you rely on Prettier to maintain consistent formatting across your projects. I ran into this a few weeks ago after an update to the Prettier extension, and it took some digging to get things back on track. Here's how I resolved it, and how you can fix it too.

Why This Happens

The error typically happens when Prettier is no longer registered as the default formatter for certain file types in VS Code. It could be due to:

- Conflicts with other formatting extensions or tools installed in your workspace
- Changes to Prettier's configuration or updates to the extension itself
- A mismatch between your workspace settings and global VS Code settings
- Issues with Prettier's version compatibility with your Node.js or VS Code version

In my case, the issue appeared after updating to a newer version of Prettier and running it with a project using a custom `.prettierrc` configuration file. The error popped up specifically for JSON and Markdown files, even though these were previously set to use Prettier for formatting.

Step-by-Step Fix

1. Verify Prettier Is Installed Correctly

First, confirm that Prettier is installed and correctly configured in your project. Open a terminal in your project directory and run:

```
npm list prettier
```

This will show you which version of Prettier is installed. If Prettier isn't listed, install it locally in your project:

```
npm install --save-dev prettier
```

For global installations, you can run:

```
npm install -g prettier
```

Once installed, confirm the version with:

```
prettier --version
```

2. Check VS Code Settings

In many cases, the issue lies in conflicting or missing settings in VS Code. Open your settings (File > Preferences > Settings or 'Ctrl+,') and search for "Default Formatter." Ensure Prettier is selected as the default formatter for your workspace and globally:

- Global Default: Open the Command Palette ('Ctrl+Shift+P'), type "Preferences: Open Settings (JSON)," and add/edit this configuration:

```
{
  "editor.defaultFormatter": "esbenp.prettier-vscode"
}
```

Workspace Default: Check '.vscode/settings.json' in your project directory and ensure it includes the same setting for the specific file types, e.g.:

```
{
  "[javascript]": {
    "editor.defaultFormatter": "esbenp.prettier-vscode"
  },
  "[json]": {
    "editor.defaultFormatter": "esbenp.prettier-vscode"
  }
}
```

3. Resolve File Type Association

Sometimes, VS Code doesn't associate certain file types with Prettier. You can explicitly set these associations in '.vscode/settings.json'. For example:

```
{
  "files.associations": {
    "*.json": "json",
    "*.md": "markdown"
  }
}
```

This ensures that files like `.json` and `.md` are recognized correctly and formatted by Prettier.

4. Update Prettier and Extensions

If you're using an outdated version of Prettier or the Prettier VS Code extension, compatibility issues can arise. Update both Prettier and the extension:

- Update Prettier: Run `npm update prettier` in your project directory.
- Update the Prettier Extension: Open the Extensions view in VS Code (`Ctrl+Shift+X`), search for "Prettier - Code formatter," and click the update button if available.

5. Check for Conflicting Extensions

Other formatting extensions or tools can conflict with Prettier. In my case, I realized I had ESLint's auto-fix feature enabled, which was overriding Prettier for specific files. To resolve this:

- Disable any extensions that might conflict (e.g., ESLint, Beautify) by going to the Extensions view (`Ctrl+Shift+X`) and toggling them off.
- Alternatively, configure the conflicting extension to work alongside Prettier. For ESLint, update your `.eslintrc` to include:

```
{
  "extends": [
    "plugin:prettier/recommended"
  ]
}
```

This ensures that ESLint respects Prettier's formatting rules.

6. Restart VS Code

Once you've made the changes, restart VS Code to ensure the new settings take effect. This can resolve lingering issues where VS Code doesn't immediately apply updated configurations.

Testing the Fix

After making these adjustments, test if Prettier is working as expected:

1. Create or open a file that should be formatted by Prettier (e.g., a `.json` or `.js` file).
2. Make some changes to the file that violate Prettier's formatting rules.
3. Save the file and check if Prettier automatically formats it. If not, manually trigger formatting with `Shift+Alt+F` (Windows/Linux) or `Shift+Option+F` (Mac).

Additional Tips

If you still encounter issues, consider these additional steps:

- Run Prettier directly from the terminal to check for errors:

```
prettier --write "src/**/*.{js,json,md}"
```

Check your project's `.prettierrc` file for invalid or unsupported configurations.

Ensure your project and VS Code are using compatible versions of Node.js and Prettier.

Conclusion

The "No longer eligible as formatter" error in Prettier can disrupt your workflow, but it's usually easy to fix with some configuration adjustments and updates. In my experience, double-checking VS Code settings and resolving conflicts with other extensions were the key steps. Once fixed, Prettier is back to doing what it does best-keeping your code clean and consistent. Happy coding!