

Viewing logs for GitHub Copilot in your environment

In this article

Collecting log files

Enabling debug mode

Viewing logs in Visual Studio

Further reading

Viewing and collecting log files

Viewing network connectivity diagnostics logs

Enabling debug mode

Viewing Electron logs

Further reading

Checking if GitHub Copilot is operational

View logs to troubleshoot GitHub Copilot-related errors in your IDE.

JetBrains IDEs Vim/Neovim Visual Studio Visual Studio Code

GitHub Copilot can be managed through personal accounts with GitHub Copilot for Individuals or through organization or enterprise accounts with GitHub Copilot for Business.

GitHub Copilot is free to use for verified students, teachers, and maintainers of popular open source projects. For more information, see "[About billing for GitHub Copilot](#)."

Collecting log files

The location of the log files depends on the JetBrains IDE you are using. For more information, see "[Configuring GitHub Copilot in your environment](#)."

These steps describe how to view and collect the log files for the following JetBrains IDEs:

- IntelliJ IDEA
- Android Studio
- GoLand
- PhpStorm
- PyCharm
- RubyMine
- WebStorm

The GitHub Copilot extension logs to the IDEA log location for IntelliJ plugins.

- 1 In your JetBrains IDE, open the **Help** menu.

- 2 Go to **Show Log in Finder**.
- 3 Open the `idea.log` in your preferred editor and look for any errors related to GitHub or GitHub Copilot.

For more information, see the [Locating IDE log files](#) in the IntelliJ documentation.

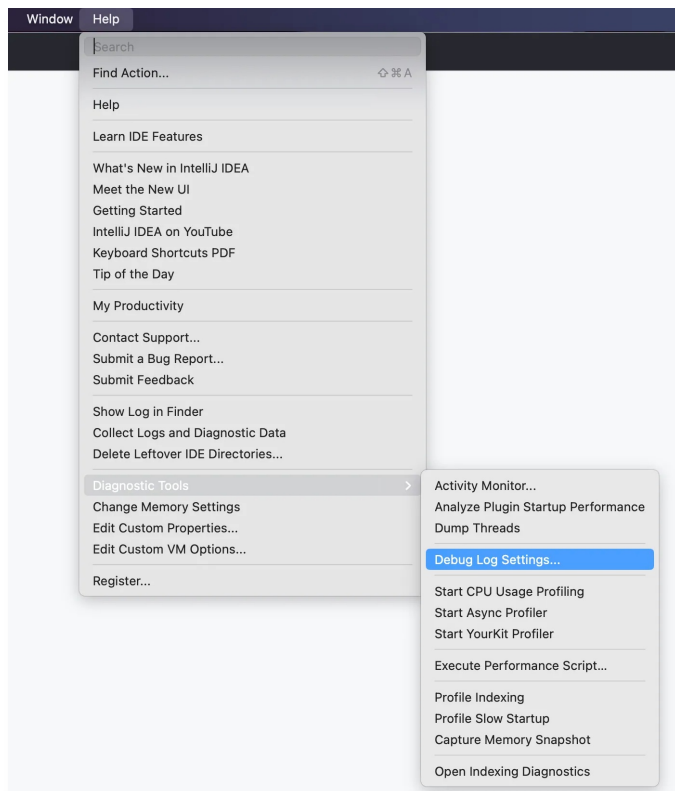
Collect log files from Rider [↗](#)

- 1 In Rider, open the **Help** menu.
- 2 Go to **Diagnostic Tools**.
- 3 Go to **Show Log in**.
- 4 Open the `idea.log` in your preferred editor and look for any errors related to GitHub or GitHub Copilot.

Enabling debug mode [↗](#)

If you find the log file doesn't contain enough information to resolve an issue, it may help to temporarily enable debug logging. This can be especially helpful for debugging network-related issues.

- 1 In the menu bar, click **Help**, select **Diagnostic Tools**, and click **Debug Log Settings....**



- 2 In the "Custom Debug Log Configuration" window, add a new line with the following content, then click **OK**.

Text



```
#com.github.copilot:trace
```

- 3 Keep using your IDE until you encounter the issue again, then collect the log file as described in "[Collecting log files](#)."
- 4 When you have the information you need, disable debug mode by removing `#com.github.copilot:trace` from the "Custom Debug Log Configuration" window.

Viewing logs in Visual Studio

The log files for the GitHub Copilot extension are stored in the standard log location for Visual Studio extensions.

- 1 Open the the **View** menu in Visual Studio.
- 2 Click **Output**.
- 3 On the right of the Output view pane, select **GitHub Copilot** from the dropdown menu.

Further reading

- "[Log all activity to the log file for troubleshooting](#)" in the Visual Studio documentation

Viewing and collecting log files

The log files for the GitHub Copilot extension are stored in the standard log location for Visual Studio Code extensions. The log files are useful for diagnosing connection issues.

- 1 Open the **View** menu in Visual Studio Code.
- 2 Click **Output**.
- 3 On the right of the Output view pane, select **GitHub Copilot** from the dropdown menu.

Alternatively, you can open the log folder for Visual Studio Code extensions in your system's file explorer. This is useful if you need to forward the log files to the support team.

- 1 Open the VS Code Command Palette
 - For Mac:
 - Use: `Shift + Command + P`
 - For Windows or Linux:
 - Use: `Ctrl + Shift + P`
- 2 Type "Logs", and then select **Developer: Open Extension Logs Folder** from the list.

Viewing network connectivity diagnostics logs

If you encounter problems connecting to GitHub Copilot due to network restrictions, firewalls, or your proxy setup, use the following troubleshooting steps.

- 1 Open the VS Code Command Palette
 - For Mac:
 - Use: `Shift + Command + P`
 - For Windows or Linux:
 - Use: `Ctrl + Shift + P`
- 2 Type "Diagnostics", and then select **GitHub Copilot: Collect Diagnostics** from the list. This opens a new editor with the relevant information that you can inspect yourself or share with the support team.
- 3 Check the section on **Reachability** to determine if GitHub Copilot can actually access the necessary services.

Note: If your error is related to certificates, it helps to check these logs for the `Custom Certificates:` line. If this line says `disabled`, you are not using GitHub Copilot for Business, so custom certificates are not supported. For more information, see "[About GitHub Copilot for Business](#)."

Enabling debug mode

If you find the log file doesn't contain enough information to resolve an issue, it may help to temporarily enable debug logging. This can be especially helpful for debugging network-related issues.

- 1 Open the VS Code Command Palette by pressing `Shift + Command + P` (Mac) / `Ctrl + Shift + P` (Windows/Linux).
- 2 Type `settings`, then click **Preferences: Open User Settings (JSON)**.
- 3 In the JSON object, insert the following content as a top-level property, then save the file.

JSON

```
"github.copilot.advanced": {  
  "debug.overrideLogLevels": {  
    "**": "DEBUG"  
  }  
},
```

- 4 Keep using your IDE until you encounter the issue again, then collect the log file as described in "[Viewing and collecting log files](#)."
- 5 When you have the information you need, disable debug mode by removing the content you added to your settings.

Viewing Electron logs [↗](#)

In rare cases, errors might not be propagated to the corresponding error handlers and are not logged in the regular locations. If you encounter errors and there is nothing in the logs, you may try to see the logs from the process running VS Code and the extension.

1 Open the VS Code Command Palette

◦ For Mac:

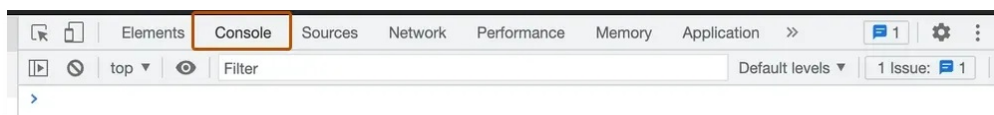
- Use: `Shift + Command + P`

◦ For Windows or Linux:

- Use `Ctrl + Shift + P`

2 Type "Toggle", and then select **Developer: Toggle Developer Tools** from the list.

3 In the Developer Tools window, select the **Console** tab to see any errors or warnings.



Further reading [↗](#)

- "[Troubleshooting network errors for GitHub Copilot](#)"
- "[Network Connections in Visual Studio Code](#)" in the Visual Studio Code documentation

Checking if GitHub Copilot is operational [↗](#)

To check if GitHub Copilot is operational, run the following command in Vim/Neovim:

```
:Copilot status
```

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)