



Manage content exclusions

7 minutes

The content exclusion feature in GitHub Copilot helps protect sensitive information by preventing the use of specific files, directories, or repositories to inform code-completion suggestions.

In this unit, you learn about:

- Enabling content exclusions from repositories and organizations.
- Analyzing the impact of content exclusions on generated code suggestions.
- Identifying scenarios where content exclusions might not be fully effective.

Configurations for content exclusion

To implement content exclusion strategies, repository administrators and organization owners can use the following configurations.

Configure content exclusions for repositories

1. On GitHub, go to the main page of the repository.
2. Under the repository name, select **Settings**.
3. In the sidebar, in the **Code & automation** section, select **Copilot**.
4. In the **Repositories and paths to exclude** section, specify the files or directories to exclude from Copilot suggestions.

Configure content exclusions for organizations

1. In the upper-right corner of GitHub, select your profile photo, and then select **Your organizations**.
2. Next to the organization, select **Settings**.
3. On the left sidebar, select **Copilot > Content exclusion**.
4. Enter the details of files or repositories to exclude from Copilot suggestions.

Impact of content exclusion on code suggestions

You can use content exclusions to configure GitHub Copilot to ignore certain files. When you exclude content from GitHub Copilot:

- Code completion is no longer available in the affected files.
- The content in affected files won't inform code completion suggestions in other files.
- The content in affected files won't inform GitHub Copilot Chat responses.

Content exclusions can significantly affect the quality and relevance of code suggestions that GitHub Copilot generates. When you exclude certain files or directories, GitHub Copilot won't use the content in those files to inform its suggestions. This action can lead to more secure and compliant code suggestions, but it might also reduce the overall context available to GitHub Copilot. This reduction could potentially affect the accuracy and usefulness of the suggestions.

For example, excluding a critical configuration file might prevent Copilot from suggesting relevant code snippets that depend on the configurations defined in that file. It's essential to carefully analyze which files should be excluded to balance security and functionality.

You can specify content exclusions only in the settings for an organization or repository. Content exclusion settings that are defined in an organization or repository within an enterprise apply to all members who are licensed as part of a GitHub Copilot Business or GitHub Copilot Enterprise subscription.

Limitations of content exclusions

Although content exclusions are a valuable tool for managing privacy and security, they might not be fully effective in some scenarios. For instance:

- **IDE limitations:** In some integrated development environments (IDEs), content exclusions might not apply when you're using certain features, such as Copilot Chat. For example, in Visual Studio Code and Visual Studio, content exclusions are not applied when you use the `@github chat` participant in your question.
- **Semantic information:** Copilot might still use semantic information from an excluded file if the IDE provides the information in a nonexcluded file. This includes type information and hover-over definitions for symbols or function calls used in code.
- **Policy scope:** Content exclusion settings apply only to members of the organization in which you configure the content exclusion. Anyone else who can access the specified files can still see code completion suggestions and Copilot Chat responses referencing the specified files.

Understanding these limitations is crucial for effectively managing content exclusions and ensuring that sensitive information is adequately protected.

Next unit: Troubleshoot common problems with GitHub Copilot

[< Previous](#)[Next >](#)