GUIDE

GitHub Copilot in VS Code: April 2026 Updates

> 📋 **RECOMMENDED** — Useful improvements. Plan your upgrade.

> **Version:** v1.116-v1.119 | **Released:** 2026-04-30 | **Upgrade from:** unknown

---

Release at a Glance

The April 2026 updates for GitHub Copilot in Visual Studio Code (versions v1.116-v1.119) bring a significant evolution in AI-assisted development. This minor release series deepens Copilot's integration into the VS Code ecosystem and expands its capabilities, particularly in autonomous agent workflows and interactive chat.

Here's a quick rundown for developers:

- **Native Integration:** Copilot is now built directly into VS Code (v1.116), streamlining setup and improving core experience.
- **Agent-Centric Workflows:** Experience advanced AI agent capabilities with Autopilot for autonomous task execution, a dedicated Agents Window, and enhanced planning.
- **Expanded Chat:** Chat now supports generating Mermaid diagrams, integrates Anthropic models for diverse reasoning, and features an improved inline UI.
- **Data Policy Update:** A critical policy change regarding how GitHub uses Copilot interaction data for model improvement takes effect April 24, 2026. Review this.
- **Performance Concerns:** Be aware of reported performance regressions, with some users experiencing slow responses, especially with Opus models.

Headline New Features

This release cycle introduces a suite of features designed to make Copilot a more integral and powerful part of your daily development workflow. The focus is clearly on elevating Copilot from a code completion tool to a collaborative AI agent.

Native VS Code Integration (v1.116)

Why this matters: Starting with v1.116, GitHub Copilot is no longer just an extension; it's now natively built into Visual Studio Code. This change simplifies installation, ensures tighter integration with the editor's core functionalities, and lays the groundwork for more seamless AI experiences. For developers, this means less friction in getting started and a more cohesive environment.

Advanced AI Agent Workflows

The April releases significantly push the boundaries of Copilot's agent capabilities, enabling more autonomous and structured AI assistance.

- **Autopilot for Autonomous Sessions:** Currently in public preview, Autopilot allows Copilot to manage and execute multi-step development tasks autonomously. Instead of just suggesting code, it can now plan, execute, and iterate on solutions with minimal human intervention.
    - **Real-world insight:** Imagine delegating a refactoring task or the implementation of a small feature to Copilot, letting it manage the file changes and code generation based on your high-level prompt. This shifts the developer's role towards higher-level problem definition and oversight.
- **Dedicated Agents Window:** A new, dedicated UI window within VS Code provides a centralized hub for interacting with and managing your AI agents. This improves visibility into ongoing agent tasks, their progress, and their decision-making processes.
- **Enhanced Planning Capabilities:** Copilot's underlying planning engine has been improved, allowing agents to better understand complex requests, break them down into manageable sub-tasks, and strategize more effectively before generating code or actions.

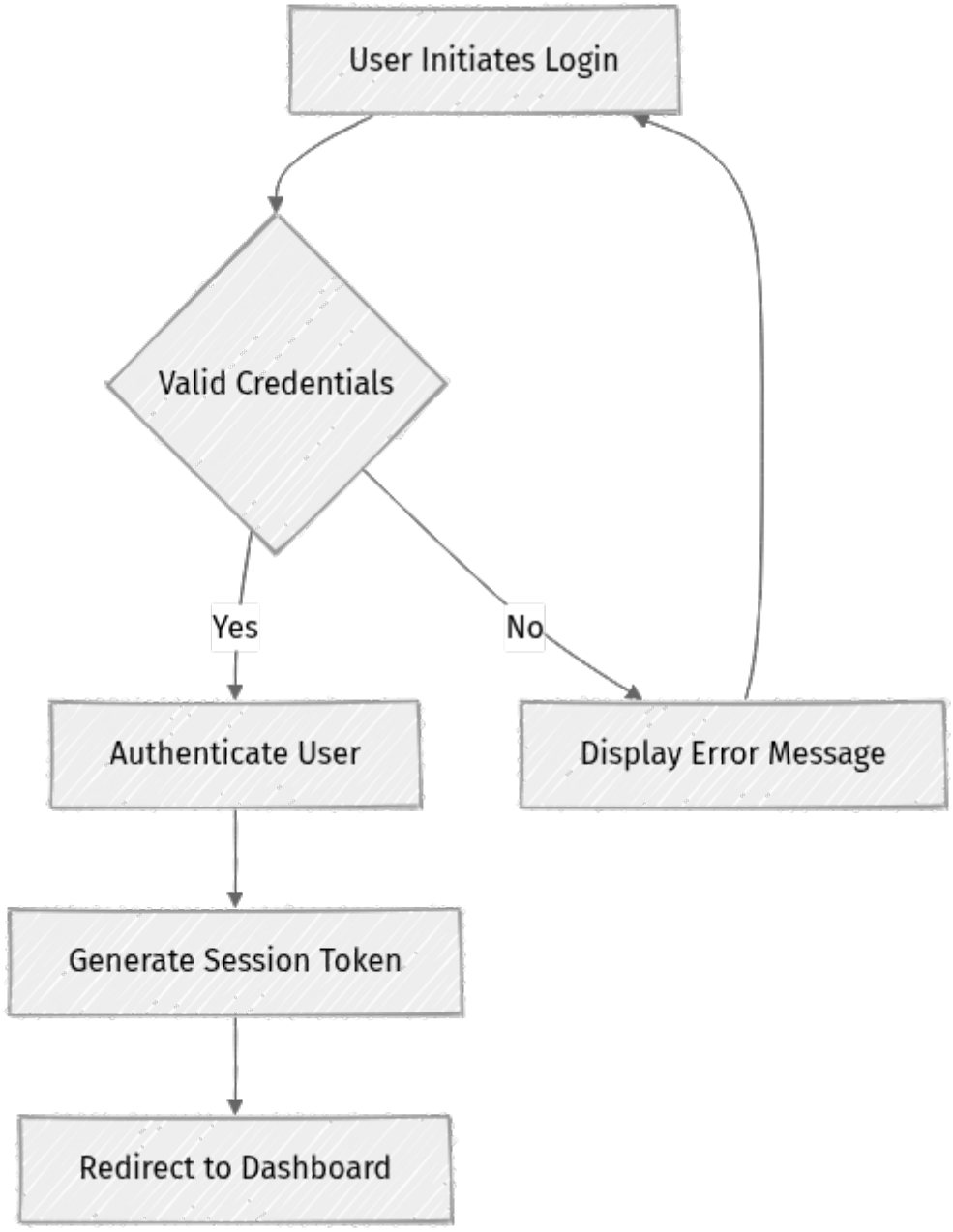
Expanded Chat Capabilities

Copilot Chat continues to evolve, becoming a more versatile tool for understanding, debugging, and documenting code.

- **Mermaid Diagram Generation:** Copilot Chat can now generate and render Mermaid diagrams directly within chat responses. This is invaluable for visualizing system architectures, flowcharts, and state diagrams without leaving your IDE.
    - **Core concept:** Instead of manually drawing diagrams or switching tools, you can simply ask Copilot to visualize a process or system based on your code or description.
    - **Example:**

```
    User: "Can you generate a flowchart for a typical user login process?"
```

```
Copilot might respond with:
```



- **Anthropic Model Integration:** Support for Anthropic's AI models has been added, offering developers access to different reasoning styles and potentially more nuanced or detailed responses for certain types of queries. This expands the underlying intelligence available to Copilot.
- **Improved Inline Chat UI:** The inline chat experience, which allows you to interact with Copilot directly within your code editor, has received UI enhancements for better usability and less disruption to your coding flow.

Copilot for CLI & Remote Control

Copilot's reach extends beyond the VS Code editor to your command line and remote development environments.

- **Copilot for CLI:** Get AI assistance directly in your terminal for shell commands, Git operations, and more. This accelerates command-line productivity by suggesting commands or explaining complex ones.
- **Remote Control for Copilot CLI:** Manage and interact with Copilot CLI sessions across remote machines, allowing for consistent AI assistance regardless of your development environment.

Semantic Codebase Search

This new capability allows Copilot to perform search queries that understand the meaning and context of your code, rather than just keyword matching.

- **Why this matters:** Instead of searching for "user service" and getting every file with those words, you can ask "Find the service responsible for user authentication logic," and Copilot will leverage its understanding of your codebase to pinpoint relevant sections, even if the exact keywords aren't present. This significantly boosts productivity in large or unfamiliar codebases.

Changes to Functionality & Integration

Beyond the headline features, these releases bring a deeper integration of Copilot within VS Code and offer more control for specific environments.

- **Deeper VS Code Integration:** The native integration (v1.116) means Copilot is now more tightly coupled with VS Code's core APIs and UI elements. This enables more fluid interactions, better context awareness, and potentially faster response times as overhead is reduced.
- **Enhanced Controls for Managed Environments:** Visual Studio Code 1.117 introduced expanded controls specifically for business and enterprise users. These controls provide administrators with more granular options for managing Copilot's behavior, data usage settings, and feature access within managed development environments. This is crucial for organizations with strict compliance or security requirements.

Important Policy Update

A significant change in GitHub's data usage policy for Copilot takes effect this release cycle.

- **Effective Date:** Starting **April 24, 2026**, GitHub will begin using your Copilot interaction data to improve its underlying models.
- **What is "Interaction Data"?** This includes inputs you provide to Copilot (e.g., prompts, code snippets you ask it to analyze or generate from) and the outputs Copilot produces.
- **Why this matters:** This policy update aims to enhance the quality and relevance of Copilot's suggestions and agent capabilities over time. However, developers and organizations should be aware of this change, especially concerning proprietary code or sensitive information. Review your organization's policies and Copilot settings to ensure compliance.
    - 🧠 **Important:** If your organization has strict data governance policies, it is imperative to review the updated GitHub Copilot terms and consider how this change impacts your development practices.

Known Issues & Performance Notes

While the April updates bring exciting new features, some users have reported performance regressions that warrant attention.

- **Degraded Performance with Opus Models:** Several reports indicate that GitHub Copilot, particularly when utilizing Opus models, has become "excruciatingly slow" for some users in Visual Studio 2026. Responses that previously took seconds are now reportedly taking minutes to generate.
    - ⚠️ **What can go wrong:** This can severely impact developer productivity, turning an AI assistant into a bottleneck.
- **General Performance Degradation for Copilot Pro:** Some Copilot Pro users have also noted a general degradation in performance and "missing models" during April 2026. This suggests broader issues beyond specific model integrations.

Developers experiencing these issues are encouraged to:

1. **Check for Updates:** Ensure both your VS Code installation and Copilot extension are on the latest available versions within the v1.116-v1.119 range.
2. **Report Feedback:** Provide detailed feedback to GitHub through official channels, including specific scenarios where performance is impacted.
3. **Monitor Official Channels:** Keep an eye on GitHub's official changelog and community discussions for updates or workarounds related to these performance concerns.

How to Upgrade

Upgrading GitHub Copilot in Visual Studio Code is straightforward, as it's now a native component or a well-integrated extension.

1. **Update Visual Studio Code:** Ensure your VS Code installation is up to date. The native integration for Copilot (v1.116) is part of the VS Code core.
    - Open VS Code.
    - Go to `Help` > `Check for Updates` (or `Code` > `Check for Updates` on macOS).
    - Follow the prompts to install any pending VS Code updates.
2. **Update Copilot Extension (if applicable):** If you were using Copilot as a separate extension before v1.116, ensure it's updated.
    - Open the Extensions view (`Ctrl+Shift+X` or `Cmd+Shift+X`).
    - Search for "GitHub Copilot".
    - If an update is available, click the `Update` button.
3. **Restart VS Code:** After updating, restart VS Code to ensure all changes take effect.

For those managing VS Code installations programmatically or in enterprise environments, you can typically use the VS Code CLI to update:

```
code --force --install-extension GitHub.copilot
```

This command ensures you have the latest version of the Copilot extension. However, for the core native integration, updating VS Code itself is the primary step.