Welcome to the July 2023 release of Visual Studio Code. There are many updates in this version that we hope you'll like, some of the key highlights include:

* [**Accessibility improvements**](https://code.visualstudio.com/raw/#accessibility) - Accessible View support for notifications, chat responses, and hovers.
* [**VS Code Profiles**](https://code.visualstudio.com/raw/#profiles) - Finer control with partial profiles and "Apply to all profiles" options.
* [**New diff editor features**](https://code.visualstudio.com/raw/#diff-editor) - Collapse unchanged regions, better diff region text alignment.
* [**Git repositories with symlinks**](https://code.visualstudio.com/raw/#support-git-repositories-with-symbolic-links) - Support for repository paths with symbolic links.
* [**Notebook updates**](https://code.visualstudio.com/raw/#notebooks) - Search text in closed notebooks, "sticky scroll" displays Markdown headers.
* [**Python test discovery**](https://code.visualstudio.com/raw/#python) - Error tolerant pytest discovery continues across all files.
* [**Access Jupyter servers in GitHub Codespaces**](https://code.visualstudio.com/raw/#jupyter) - Connect to a remote Jupyter server in a codespace.
* [**GitHub pull request creation**](https://code.visualstudio.com/raw/#github-pull-requests-and-issues) - Better base branch detection, remember previous create PR options.
* [**Preview: GitHub Copilot UX**](https://code.visualstudio.com/raw/#github-copilot) - Quick Chat improvements, iterative /fix command.

If you'd like to read these release notes online, go to [Updates](https://code.visualstudio.com/updates) on [code.visualstudio.com](https://code.visualstudio.com/).

**Insiders:** Want to try new features as soon as possible? You can download the nightly [Insiders](https://code.visualstudio.com/insiders) build and try the latest updates as soon as they are available.

Accessibility

Accessible View improvements

The **Open Accessible View** (Alt+F2) command allows screen reader users to inspect content character by character, line by line. Last iteration, we added Accessible Views for editor hovers and chat panel responses.

There are now Accessible Views for notifications, inline chat responses, Jupyter notebook output, and workbench hovers.

When an element that offers an Accessible View is focused, a hint is provided in the ARIA label. That hint can be disabled per feature via the accessibility.verbosity setting.

There are **Show Next in Accessible View** (Alt+[) and **Show Previous in Accessible View** (Alt+´) commands for easy navigation between elements in the chat panel and notifications list.

Breadcrumb symbol information

After using the **Focus and Select Breadcrumbs** (breadcrumbs.focusAndSelect) command, the [Breadcrumb](https://code.visualstudio.com/docs/editor/editingevolved#_breadcrumbs) navigation now includes symbol information for screen readers.

Input control and result navigation consistency

The experience when navigating between an input control (for example, search or filter input) and its results is now consistent across the Extensions view, the Keyboard Shortcuts editor, and the Comments, Problems, and Debug Console panels using (Ctrl+DownArrow) and (Ctrl+UpArrow).

Accessible diff viewer

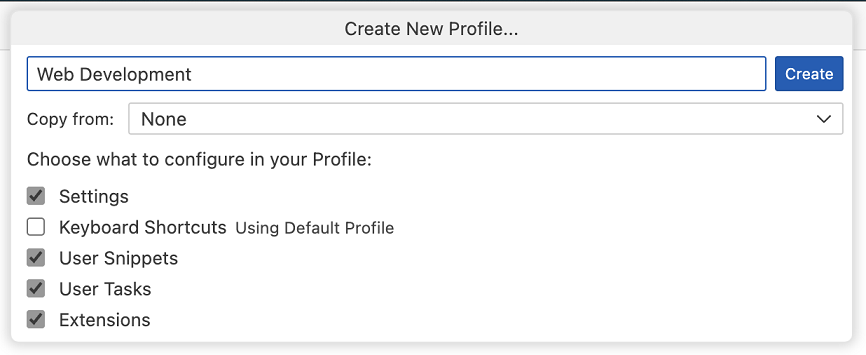
In the diff editor, the **Difference Review Pane** was renamed to **Accessible Diff Viewer**. The accessible diff viewer addresses accessibility shortcomings of the diff editor and can be opened with F7 or the **Open Accessible Diff Viewer** command from the diff editor **More Actions** menu. Along with this rename, we also fixed some bugs and added some new features, such as being able to launch the Accessibility Help dialog or jump to the modified editor from the viewer.

Profiles

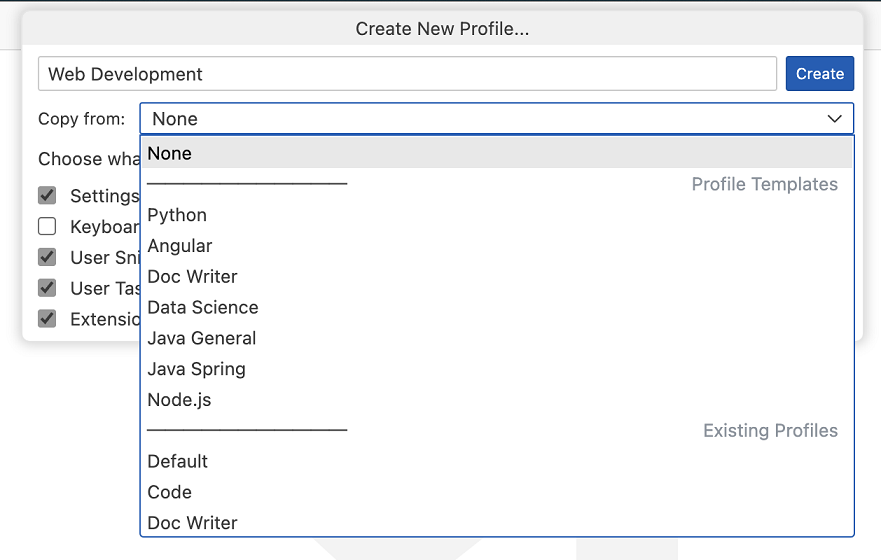
Support for partial profiles

You can now create a profile in which you customize only a subset of configurations (settings, keyboard shortcuts, snippets, tasks, and extensions). For example, you can create a profile with all configurations except for keyboard shortcuts, and VS Code applies the keyboard shortcuts from the Default Profile when this profile is active.

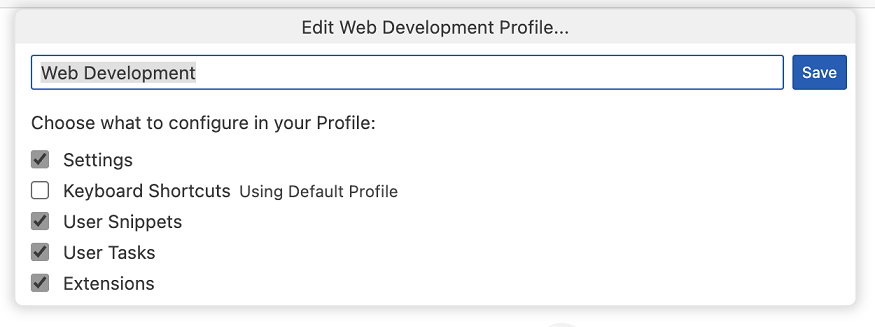
In the **Create New Profile** screenshot below, **Keyboard Shortcuts** is unchecked and a **Using Default Profile** message is displayed. Any keyboard shortcuts active during that VS Code session are not applied to the new profile.



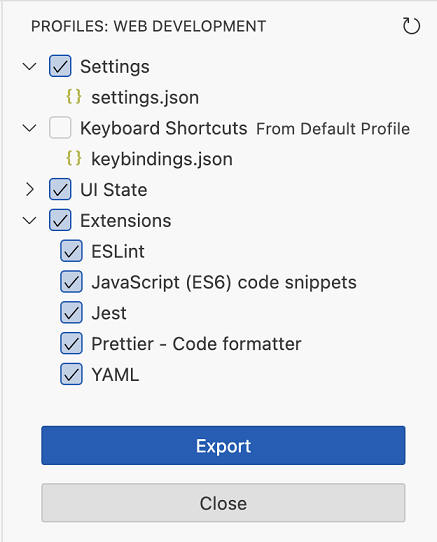
You can also create a partial profile when copying from a Profile Template or from an existing profile, using the **Copy from** dropdown menu.



You can also configure an existing profile to use configurations from the Default Profile by editing the profile in the **Edit Profile** UI.

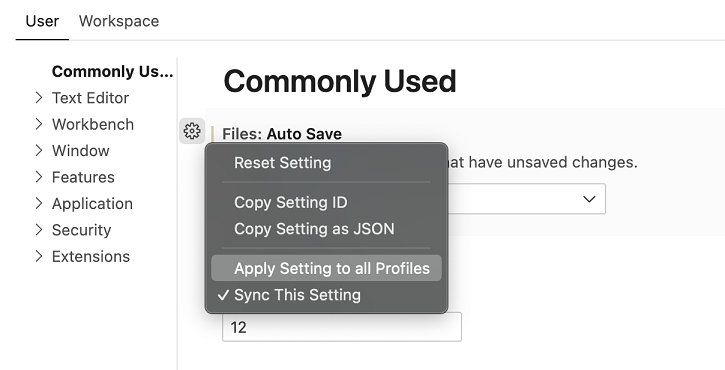


When exporting a partial profile, you can choose to export only the configurations that are customized in the profile or also includes the configurations from the Default Profile.



Apply a setting to all profiles

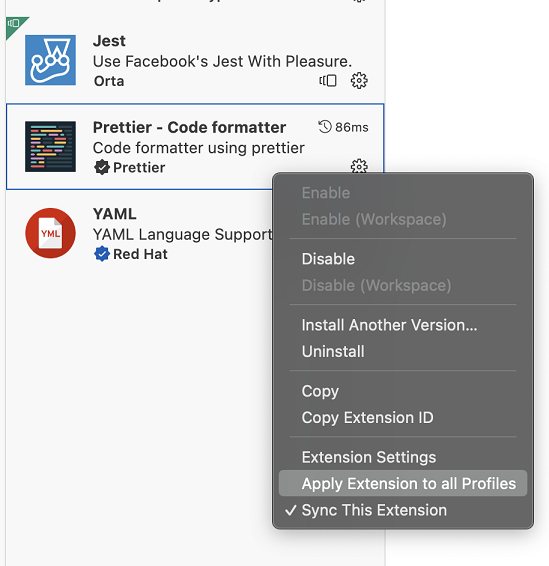
You can now apply a setting to all profiles by selecting the **Apply Setting to all Profiles** option in the Settings editor.



This applies the value of the setting to all profiles. Any updates to this setting from any profile are applied to all profiles. You can revert this synchronization by unchecking the **Apply Setting to all Profiles** option.

Apply an extension to all profiles

You can now include an extension in all profiles by selecting the **Apply Extension to all Profiles** option in the Extensions view.



This makes this extension available in all profiles. You can revert this by unchecking the **Apply Extension to all Profiles** option.

Settings JSON file command IDs adjusted on non-default profiles

The **Preferences: Open User Settings (JSON)** (workbench.action.openSettingsJson) command will now open the profile-specific settings JSON file when a user is in a non-default profile. To show application-scoped settings, use the **Preferences: Open Application Settings (JSON)** (workbench.action.openApplicationSettingsJson) command instead.

Workbench

Release notes for upcoming update

Whenever a monthly stable update is ready to be installed, you can now preview the release notes via the **Show Update Release Notes** command in the **Manage** menu (gear button) at the bottom of the Activity bar.

Editor

Markdown paste URLs as formatted links

The markdown.editor.pasteUrlAsFormattedLink setting now has a new option of smart. The smart paste feature can detect when you are pasting into text where you may not want automatic Markdown link formatting. Some situations ignored by smart pasting include pasting within a Markdown link, over another Markdown link, and within a code block.

By default, the pasteUrlAsFormattedLink feature is currently off (never), but you can change the setting to smart or always (always create a formatted URL link).

Asynchronous tokenization rollout

In the [1.76 update](https://code.visualstudio.com/updates/v1_76#_asynchronous-tokenization), we implemented editor document asynchronous tokenization on a web-worker to improve the performance of the renderer process. Since then, we have fixed several bugs caused by rare race conditions and with this release, we are starting to roll out this feature to Stable users.

If you do not want to wait, you can set editor.experimental.asyncTokenization to true to get asynchronous tokenization immediately.

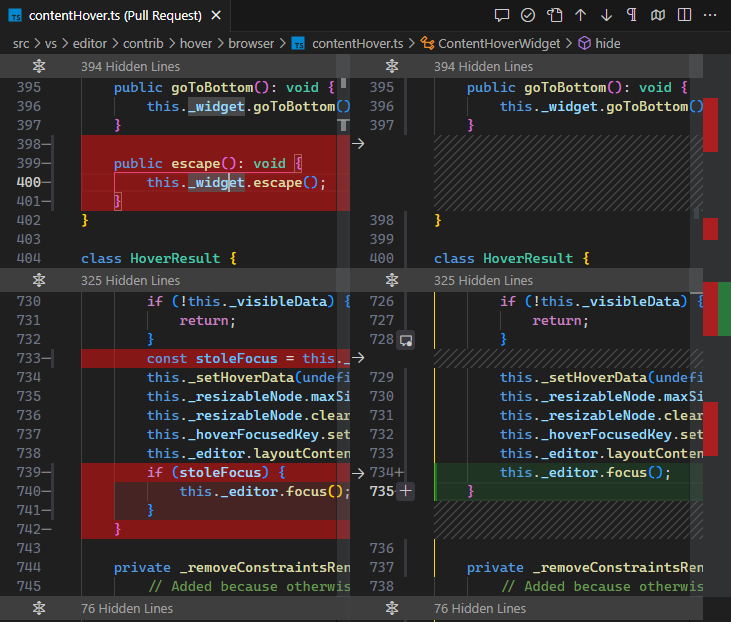
Diff editor

In this iteration, we continued to work on the new diff editor and plan to gradually roll it out to Stable users. You can try it out immediately by setting "diffEditor.experimental.useVersion2": true. The new diff editor includes several new features and bug fixes described below.

Collapse unchanged regions

Use diffEditor.experimental.collapseUnchangedRegions to enable hiding unchanged regions in the new diff editor. This feature is useful when reviewing large diffs with a lot of unchanged lines.

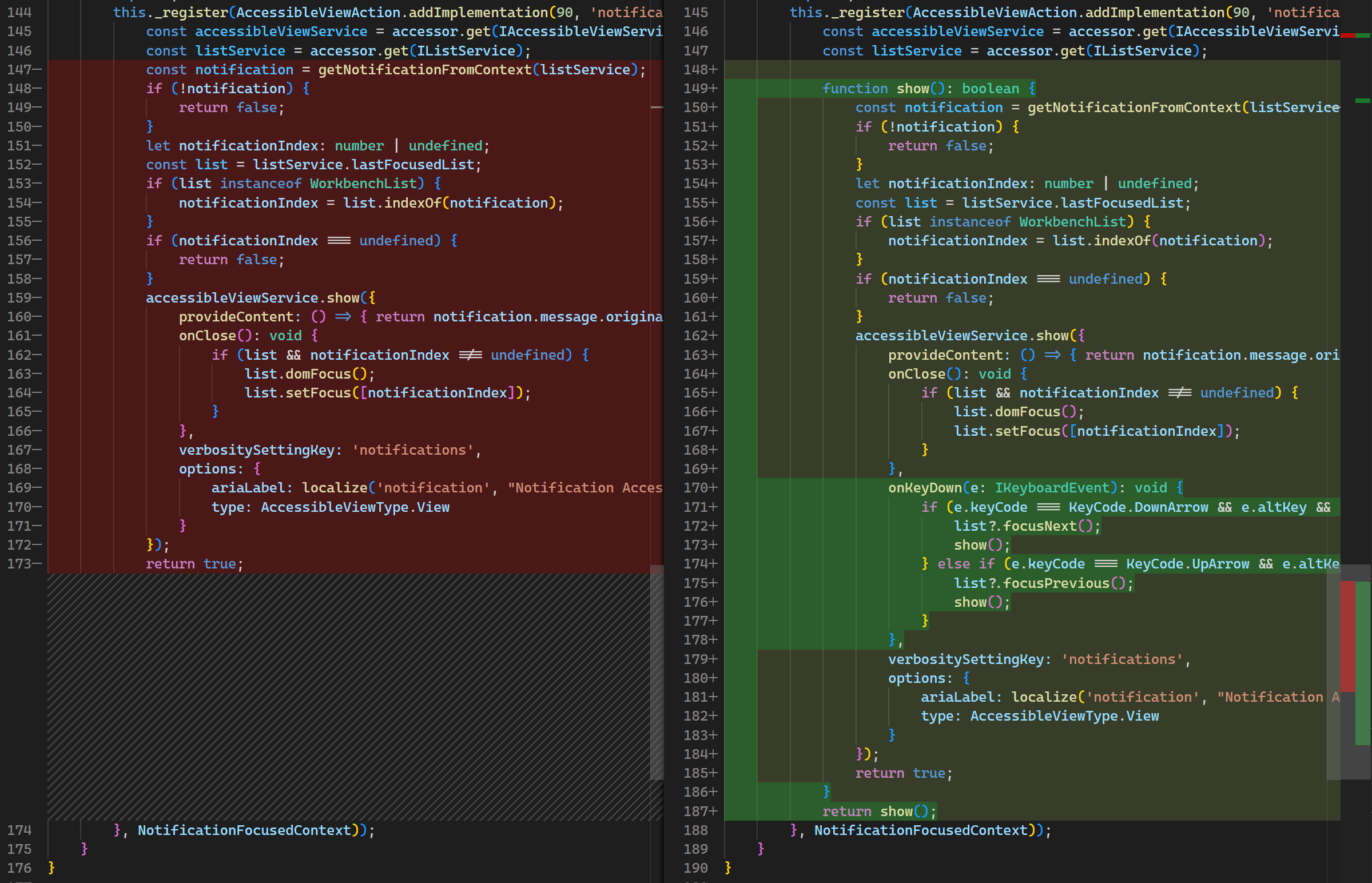
The borders of the hidden-lines blocks can be dragged or clicked to reveal text from the bottom or top.



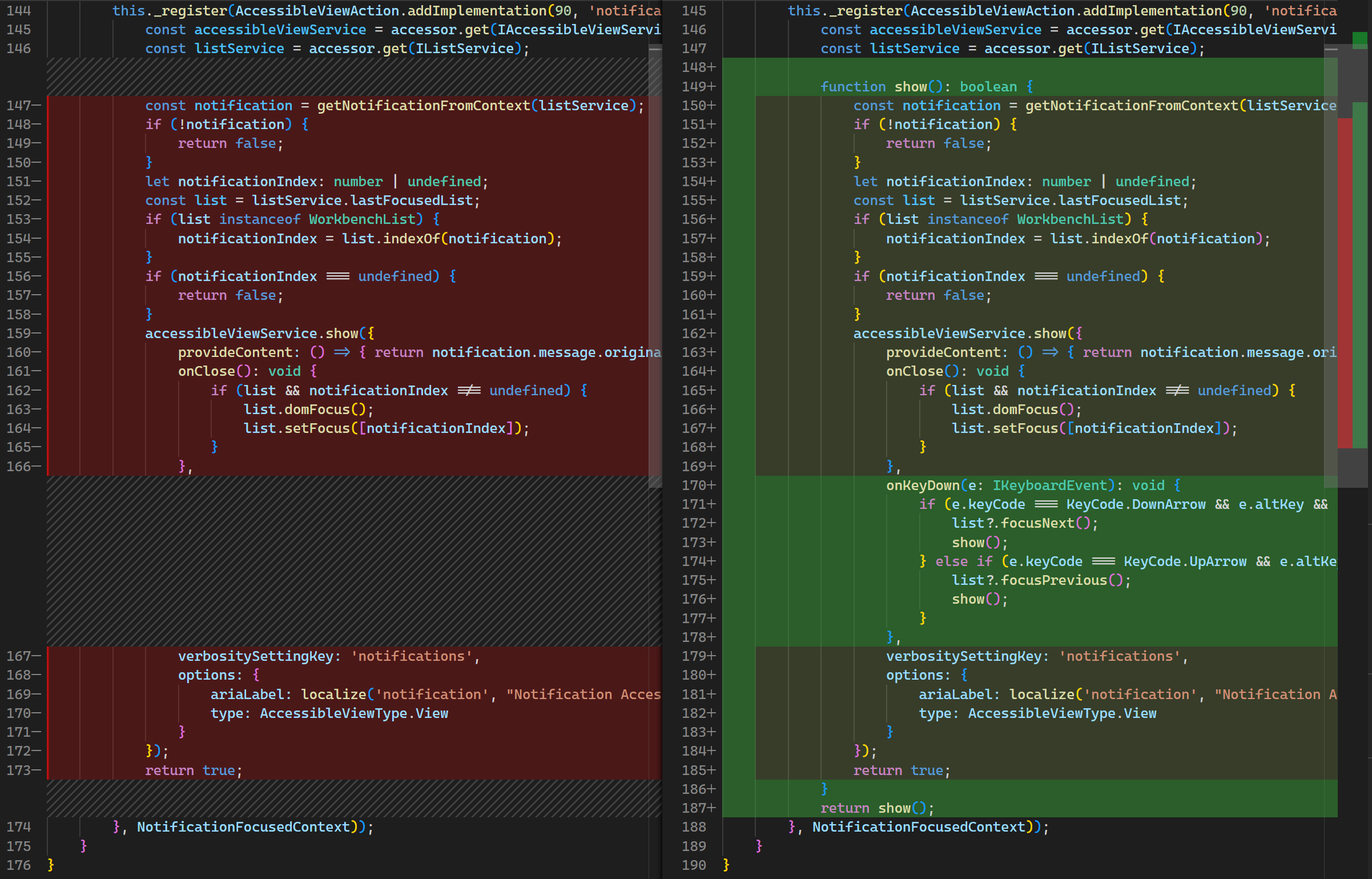
Diff region alignment

The new diff editor now aligns text within diff regions, making it much easier to review diffs where indentation has changed and lines have been inserted.

Before (without diff region alignment and whitespace diffing enabled):



After (with diff region alignment):



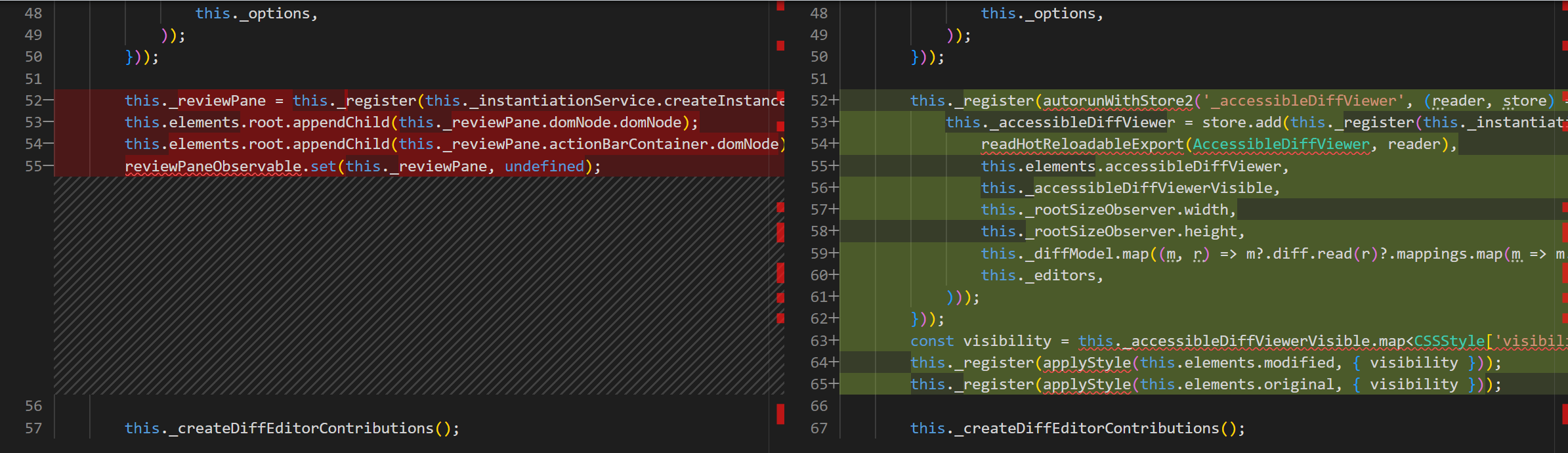
The new diff algorithm is now enabled by default

We have completed the roll-out of the new diff algorithm, which is now the default in VS Code. You can read about some of the algorithm improvements in [past release notes](https://code.visualstudio.com/updates/v1_78#_diff-algorithm-improvements).

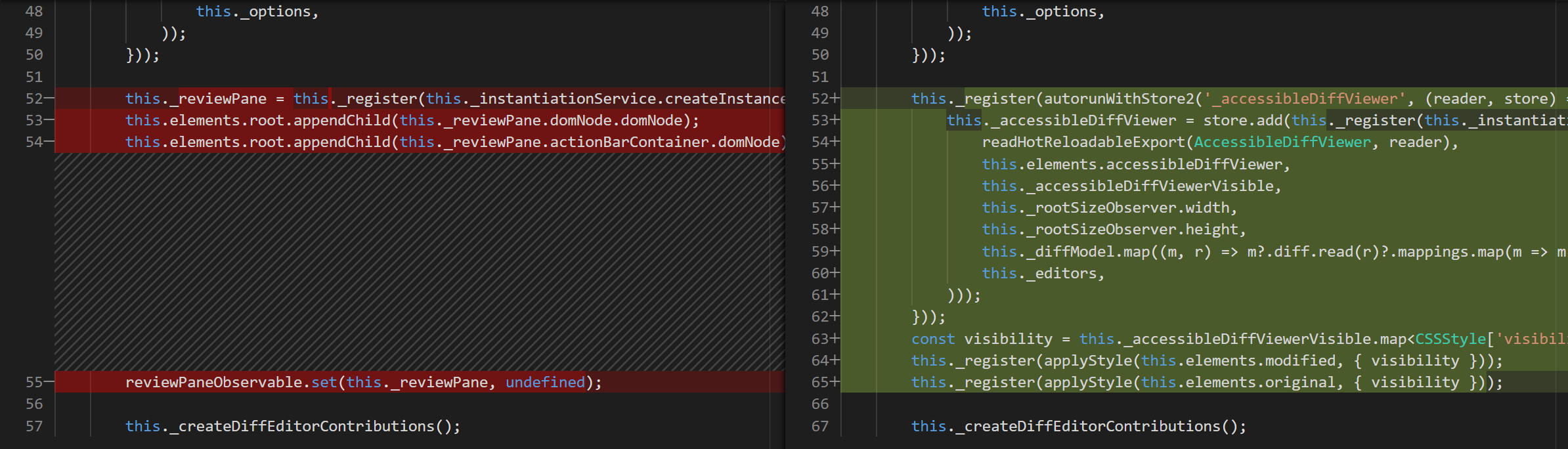
Diff noise reduction

In this update, we added heuristics to the new diff algorithm to reduce the probability of matching unrelated words.

This is how the diff editor looked before (note how this matches on both sides, even though the matching is not helpful):



This is how it looks now with the heuristic improvements (note that there are fewer unrelated matches):



Switch sides

When using the side-by-side view of the new diff editor, there is now a new command **Diff Editor: Switch Sides** that allows you to jump from one side to the other. When switching sides, the cursor position and the current selection in the active editor are mapped to the other side.

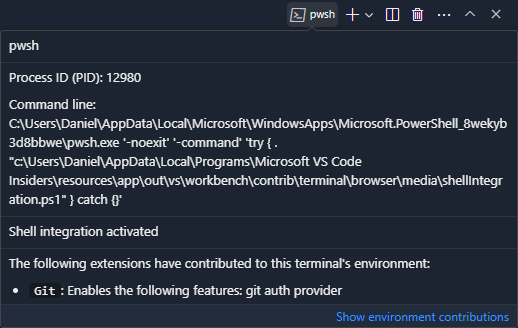
And there is more

Check out the [1.80 release notes](https://code.visualstudio.com/updates/v1_80#_new-diff-editor) for a more detailed description of what has changed in the new diff editor!

Terminal

Custom hover in inline terminal tab

The panel inline terminal tab that shows when there is a single terminal now uses the custom hover for a more consistent appearance and to support embedded links.



Source Control

Support Git repositories with symbolic links

Based on community feedback, there is now support for Git repositories that have symbolic links on their path by using the --path-format option, added to the rev-parse Git command in version 2.31. Starting with this release, all source control functionality (for example, the **Source Control** view, repository detection, added/modified/deleted lines editor decorations, etc.) should work as expected when opening such a repository.

Debug

JavaScript debugger

**Support for the Fast Node (Version) Manager**

The JavaScript debugger now supports the [Fast Node Manager](https://github.com/Schniz/fnm) (fnm) in its runtime version. Users of fnm can now set the runtimeVersion property of launch.json to select the Node.js version.

Notebooks

Rich content search for closed notebooks

In a previous release, we introduced rich content workspace search for open notebooks. There is now preview support to also see rich content results for closed notebooks.

When you search for text in the Search view, the result previews from notebooks should match how the string looks when you open the notebook editor. Note that this will only show results for inputs when a notebook is closed. To try this, enable search.experimental.closedNotebookRichContentResults in your settings.

Improved performance for large streaming outputs

We have made performance improvements for large cell outputs that are streamed during execution. This reduces the amount of rendering activity but currently only applies to scrollable cell outputs.

Notebook sticky scroll

This release brings the first iteration of "sticky scroll" for the notebook editor. The sticky scroll display at the top of the notebook editor shows the Markdown headers for the cell or cell output that are currently visible. Selecting a sticky scroll "line" focuses that cell, allowing you to navigate the notebook quickly. This feature is enabled via the notebook.stickyScroll.enabled setting or the **View: Toggle Notebook Sticky Scroll** command.

VS Code for the Web

Always load latest versions of built-in extensions

[vscode.dev](https://vscode.dev/) now always loads the latest versions of the built-in extensions. This means that you will no longer see the "Reload Required" notification to update built-in extensions to the latest version when you open vscode.dev.

Remote Development

The [Remote Development extensions](https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.vscode-remote-extensionpack), allow you to use a [Dev Container](https://code.visualstudio.com/docs/devcontainers/containers), remote machine via SSH or [Remote Tunnels](https://code.visualstudio.com/docs/remote/tunnels), or the [Windows Subsystem for Linux](https://learn.microsoft.com/windows/wsl) (WSL) as a full-featured development environment.

Highlights include:

* Install Remote Tunnels as a service from VS Code.
* Simplified @devcontainers/cli installation.
* Record Dev Container Feature dependencies in a lockfile.

You can learn about new extension features and bug fixes in the [Remote Development release notes](https://github.com/microsoft/vscode-docs/blob/main/remote-release-notes/v1_81.md).

Contributions to extensions

GitHub Copilot

**Quick Chat UX improvements**

The Quick Chat experience has been revamped this iteration.

Like previously, you can toggle this view with Ctrl+Shift+I. Specific improvements include:

* **Conversational chat**: The Quick Chat experience now shows your full conversation while still keeping the last question and answer in focus.
* **Slash command completion**: There are now completions for slash commands when you type /.

Additionally, there is an experimental setting, chat.experimental.defaultMode, to control what parts of chat are shown. It has three options:

* chatView - Show the Chat icon in the Activity bar to launch the chat in the Chat view (default).
* quickQuestion - Show the Chat icon in the title bar to launch Quick Chat.
* both - Both Chat icons are visible to allow you to open either chat UI.

We want to know your thoughts on this setting and the Quick Chat experience in general, so don't hesitate to open [issues](https://github.com/microsoft/vscode-copilot-release/issues) with feedback.

**Slash command improvements**

We've made the following usability improvements to slash commands in panel, inline, and Quick Chat:

1. Slash commands now render as blocks and can be deleted with a single backspace.
2. Accepting slash commands from the suggestion list, such as /clear and /help, now executes the command, removing the need to explicitly submit the chat request after accepting a completion.
3. The /createWorkspace panel chat slash command is repopulated in the chat input after it is first used in a question, making it easy to ask follow-up questions about your workspace.

The following recording illustrates all three improvements:

**Iterative fixing**

While using the /fix command, you can enable iterative fixing with the setting github.copilot.editor.iterativeFixing. When iterative fixing is enabled, if after the first fix, there remain diagnostic errors in the selection, the fix is run a second time. This setting is enabled by default.

**Notebook generation improvements**

While generating a notebook, there are additional steps to improve the quality of the generated code and also fix diagnostics detected in the generated code by language extensions. This is disabled by default, but can be enabled with following settings:

"github.copilot.notebook": {  
 "iterativeFixing": true,  
 "iterativeImproving": true  
}

**Inline Chat**

While using the inline chat, you can now enable and disable the diff view using the setting inlineChat.showDiff. You can right-click on the inline chat to display a context menu to change this setting.

**Terminal context**

Copilot Chat now knows about the active terminal's buffer and selection.

Python

**Error tolerant pytest discovery**

As part of the rollout of our new testing architecture, the Test Explorer panel now supports error-tolerant [pytest](https://docs.pytest.org/) discovery. When an error is thrown by pytest during discovery that can be handled, for example, an error from an unknown import, all other tests will still be discovered outside of the file with the error. This feature is only available on the new testing rewrite, which is still experimental. The rewrite is currently active for 100% of Insiders and 25% of Stable users but will be universally adopted shortly. In the meantime, you can opt in or out with the pythonTestAdapter setting.

**Debugger configuration for running files with arguments**

The new [Debugpy](https://marketplace.visualstudio.com/items?itemName=ms-python.debugpy) extension now provides a **Python File with Arguments** launch.json configuration, which is useful when you want to provide different input values for your Python file without modifying your code or the debugger configuration each time you run it.

To use this configuration, make sure you have the [Debugpy](https://marketplace.visualstudio.com/items?itemName=ms-python.debugpy) extension installed. Open the **Run and Debug** view by pressing Ctrl+Shift+D and select either **Create a launch.json file** or the gear icon to access the launch.json file. Select **Debugpy**, and then select **Python: File with Arguments** from the available configurations.

Now when you want to debug (F5) a Python file that requires command-line arguments, a prompt appears allowing you to enter the desired arguments to pass to the Python file.

After entering your arguments, press Enter, and the debugger will start, letting you step through your code.

*Theme:*[*Catppuccin Macchiato*](https://marketplace.visualstudio.com/items?itemName=Catppuccin.catppuccin-vsc)*(preview on*[*vscode.dev*](https://vscode.dev/theme/Catppuccin.catppuccin-vsc/Catppuccin%20Macchiato)*)*

**npm package for the Python extension API**

The [Python](https://marketplace.visualstudio.com/items?itemName=ms-python.python) extension now provides an npm package to make it easier for other extension authors to access and track changes in the Python extension API. Check out the [@vscode/python-extension](https://www.npmjs.com/package/@vscode/python-extension) npm module to work with Python environments available on the user's machine.

**Dev Container for the Python extension**

There is now a [Dev Container](https://code.visualstudio.com/docs/devcontainers/containers) for the Python extension with a Microsoft official Node.js image. With this [custom dev container](https://github.com/microsoft/vscode-python/tree/main/.devcontainer), users are able to open the [Python extension repository](https://github.com/microsoft/vscode-python) in a [GitHub Codespace](https://github.com/features/codespaces), and start working on developing and testing the Python extension instantly, with no installation required. Python versions 3.7, 3.8, 3.9, 3.10, and 3.11 are pre-installed so users can readily switch between Python versions using [pyenv](https://github.com/pyenv/pyenv).

Jupyter

**Connect to Jupyter server on Codespace**

When working on Jupyter notebook on [github.dev](https://github.dev/), you can now connect to a Jupyter server running on a GitHub Codespace instance through the kernel picker.

**Continue On with uncommitted changes**

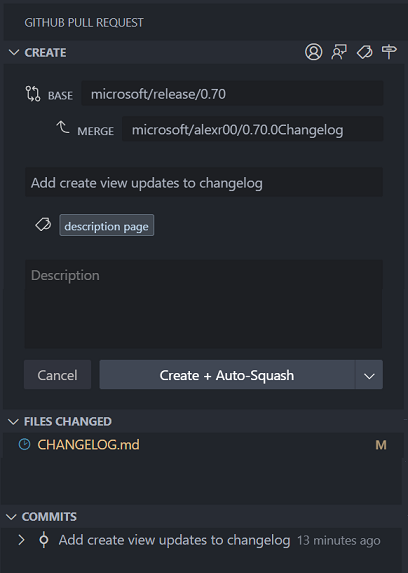
You can switch to the underlying codespace by selecting the **Continue On** button in the notebook toolbar, and any uncommitted changes in the workspace are brought to the codespace.

GitHub Pull Requests and Issues

There has been more progress on the [GitHub Pull Requests and Issues](https://marketplace.visualstudio.com/items?itemName=GitHub.vscode-pull-request-github) extension, which allows you to work on, create, and manage pull requests and issues. This iteration, we focused on updating the **Create** view to make it cleaner and more useful.

Improvements to the **Create** view include:

* The view tries to detect the best possible base branch for your PR instead of always using the default branch.
* You can add reviewers, assignees, labels, and milestones to your PR from the **Create** view.
* By default, your last "create option" is remembered (for example, draft or auto merge).
* The view performance is much faster.
* You can view diffs before publishing your branch.
* Once the branch is published, you can also view commits (this is coming soon for unpublished branches).



Review the [changelog for the 0.70.0](https://github.com/microsoft/vscode-pull-request-github/blob/main/CHANGELOG.md#0700) release of the extension to learn about the other highlights.

GitHub authentication

**Revoking GitHub OAuth tokens on sign out**

Previously, when you sign out of a GitHub account in VS Code, it would only remove OAuth tokens from the secrets storage. The token is still valid and able to make requests to GitHub's APIs. To improve the security of token management, we now attempt to [delete the token on the backend](https://docs.github.com/en/rest/apps/oauth-applications?apiVersion=2022-11-28#delete-an-app-token) making the token no longer valid after sign out.

Extension authoring

revealSetting and focusSearch added to IOpenSettingsActionOptions

Extensions can now pass the revealSetting object and focusSearch boolean into commands such as workbench.action.openSettingsJson that accept IOpenSettingsActionOptions args.

For example, the following command

vscode.commands.executeCommand('workbench.action.openSettingsJson', { revealSetting: { key: 'editor.renderWhitespace' }});

opens the user settings JSON file, and then reveals the editor.renderWhitespace setting if it exists.

New TestController.invalidateTestResults method

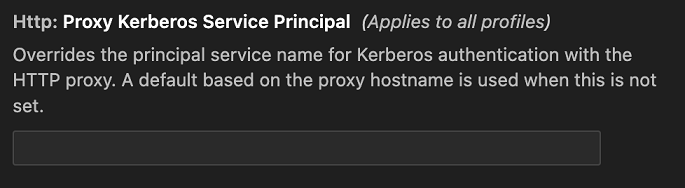
Test extensions can call the new method TestController.invalidateTestResults to indicate to the user that any results for one or more tests may no longer be applicable. For example, an extension might do something like:

onFileChange(uri => {  
 // invalidate tests in a file when a file changes:  
 testController.invalidateTestResults(getTestsInFile(uri));  
})  
onConfigurationChange(() => {  
 // invalidate results for all tests:  
 testController.invalidateTestResults();  
});

Kerberos authentication for network proxy

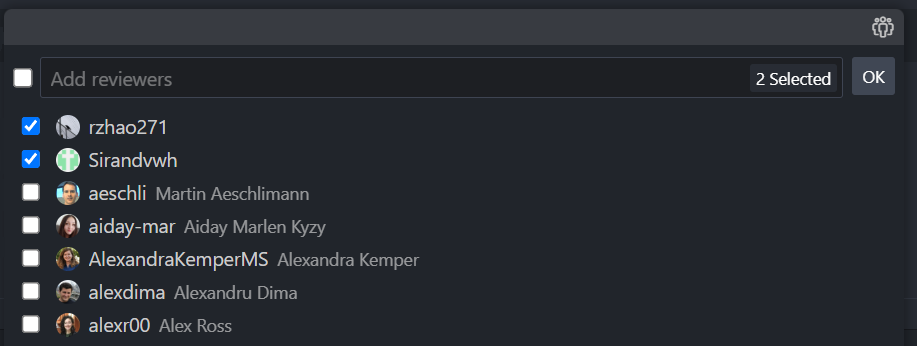
Extensions using the https Node.js module can now use network proxies that require Kerberos authentication. The local machine needs to have Kerberos configured but extensions do not need to do anything to use this feature.

In cases where the automatically selected service principal name does not work, the user can provide one with the http.proxyKerberosServicePrincipal setting:



Icons in QuickPickItem finalized

The API for icons in QuickPickItem has been finalized. You can now use the iconPath property to display icons in QuickPickItems.



Engineering

Shrinking VS Code with name mangling

A recent engineering effort reduced the size of VS Code's shipping JavaScript by 20%. In the [Shrinking VS Code with name mangling](https://code.visualstudio.com/blogs/2023/07/20/mangling-vscode) blog post, you can learn how VS Code engineers [Matt Bierner](https://hachyderm.io/@mattbierner) and [Johannes Rieken](https://twitter.com/johannesrieken) researched identifier name mangling to optimize code size and came up with effective solution using TypeScript.

More pty host improvements

Continuing on from [last month's improvements](https://code.visualstudio.com/updates/v1_80#_terminal-pty-host-improvements) to the pty host, the following improvements have been made:

* Performance
  + The pty host now only launches when it's needed, so users that never open the terminal will save around 50-100 MB.
  + Further parallelization of terminal reconnection, speeding up the time to initialize the terminal.
  + Terminal output messages are now buffered, reducing traffic going from the pty host and the window hosting the terminal.
* Diagnostics
  + Latency stats between each process are logged when in trace log level.
  + The window ID is logged to all logs in the **Terminal** output channel.
  + Pty host RPC logging now logs responses that failed as a result of an exception.

Notable fixes

* [133542](https://github.com/microsoft/vscode/issues/133542) Reattached to the wrong terminal
* [168021](https://github.com/microsoft/vscode/issues/168021) Better object completions in the JavaScript debugger
* [184894](https://github.com/microsoft/vscode/issues/184894) Cannot click and drag scroll bar in terminal

Thank you

Last but certainly not least, a big ***Thank You*** to the contributors of VS Code.

Issue tracking

Contributions to our issue tracking:

* [@gjsjohnmurray (John Murray)](https://github.com/gjsjohnmurray)
* [@starball5 (starball)](https://github.com/starball5)

Pull requests

Contributions to vscode:

* [@a-stewart (Anthony Stewart)](https://github.com/a-stewart): Increase the max width of action widgets to 80% of the browser width [PR #186157](https://github.com/microsoft/vscode/pull/186157)
* [@abhijit-chikane (Abhijit Chikane)](https://github.com/abhijit-chikane): Fix the case of the breakpoint placeholder [PR #185753](https://github.com/microsoft/vscode/pull/185753)
* [@andrewbranch (Andrew Branch)](https://github.com/andrewbranch): [typescript-language-features] Update autoImportFileExcludePatterns description [PR #186528](https://github.com/microsoft/vscode/pull/186528)
* [@colombod (Diego Colombo)](https://github.com/colombod): Add proposed api to support auto closing pairs on langauge configuration [PR #186567](https://github.com/microsoft/vscode/pull/186567)
* [@Dromantor (Andreas Pareis)](https://github.com/Dromantor): fixed #173858 minimap slider not scrolling to the top end [PR #180728](https://github.com/microsoft/vscode/pull/180728)
* [@EhabY (Ehab Younes)](https://github.com/EhabY): Expose the focused element and change event in the TreeView API [PR #184268](https://github.com/microsoft/vscode/pull/184268)
* [@gjsjohnmurray (John Murray)](https://github.com/gjsjohnmurray)
  + Use go-to-file codicon for test output inline action (fix #186369) [PR #188472](https://github.com/microsoft/vscode/pull/188472)
  + Show placeholder text in testing view progress area (fix #141181) [PR #188484](https://github.com/microsoft/vscode/pull/188484)
* [@hsfzxjy](https://github.com/hsfzxjy)
  + Generalize Ctrl+DownArrow and Ctrl+UpArrow to most input-result widgets (Fix #179967) [PR #187077](https://github.com/microsoft/vscode/pull/187077)
  + Fix context menu inaccurate positioning [PR #187157](https://github.com/microsoft/vscode/pull/187157)
* [@jeanp413 (Jean Pierre)](https://github.com/jeanp413)
  + Fixes extension activation fail after upgrade with Error: Cannot find module 'vscode' [PR #187090](https://github.com/microsoft/vscode/pull/187090)
  + Fix can't use cmd+up/down in terminal after closing find widget [PR #187781](https://github.com/microsoft/vscode/pull/187781)
  + Fix missing terminal find widget shorcuts in tooltip [PR #187787](https://github.com/microsoft/vscode/pull/187787)
* [@lkct (Rickey K. Liang)](https://github.com/lkct): Revert Python docstring color [PR #184938](https://github.com/microsoft/vscode/pull/184938)
* [@m-ou-se (Mara Bos)](https://github.com/m-ou-se): Accept a trailing colon in --goto format. [PR #186174](https://github.com/microsoft/vscode/pull/186174)
* [@mwerschy (Malte Werschy)](https://github.com/mwerschy): Implement indent guide colorization options. [PR #178112](https://github.com/microsoft/vscode/pull/178112)
* [@rielAsh24 (Ashriel)](https://github.com/rielAsh24): Fix #182014: ISO Date format [PR #185587](https://github.com/microsoft/vscode/pull/185587)
* [@ssigwart (Stephen Sigwart)](https://github.com/ssigwart): Fix single line if/else and loop indentation for Typescript and PHP [PR #136577](https://github.com/microsoft/vscode/pull/136577)
* [@SvanT (Svante Boberg)](https://github.com/SvanT): Dispose TerminalWordLinkDetector listeners [PR #187083](https://github.com/microsoft/vscode/pull/187083)
* [@weartist (Hans)](https://github.com/weartist): fix #187223 [PR #188144](https://github.com/microsoft/vscode/pull/188144)
* [@yiliang114 (易良)](https://github.com/yiliang114)
  + fix: typos [PR #186607](https://github.com/microsoft/vscode/pull/186607)
  + fix: Close #187788, recovery tree view state [PR #187902](https://github.com/microsoft/vscode/pull/187902)
* [@yshaojun](https://github.com/yshaojun)
  + fix: color hints may display twice(#175476) [PR #186926](https://github.com/microsoft/vscode/pull/186926)
  + fix: prevent duplicate reason(#186206) [PR #187985](https://github.com/microsoft/vscode/pull/187985)
  + fix: modified editor width(#175397) [PR #188012](https://github.com/microsoft/vscode/pull/188012)

Contributions to vscode-extension-samples:

* [@dsanders11 (David Sanders)](https://github.com/dsanders11): Fix EACCES error code name [PR #847](https://github.com/microsoft/vscode-extension-samples/pull/847)

Contributions to vscode-hexeditor:

* [@lorsanta](https://github.com/lorsanta)
  + fix: scroll to the top when hit home key [PR #448](https://github.com/microsoft/vscode-hexeditor/pull/448)
  + Fix: the selection count is updated when switching between tab groups. [PR #449](https://github.com/microsoft/vscode-hexeditor/pull/449)

Contributions to vscode-js-debug:

* [@zardoy (Vitaly)](https://github.com/zardoy): fix member expression completions [PR #1758](https://github.com/microsoft/vscode-js-debug/pull/1758)

Contributions to vscode-pull-request-github:

* [@mgyucht (Miles Yucht)](https://github.com/mgyucht): Correctly iterate backwards through diffs across files [PR #5036](https://github.com/microsoft/vscode-pull-request-github/pull/5036)

Contributions to vscode-vsce:

* [@drewbrokke (Drew Brokke)](https://github.com/drewbrokke): (#868) publish option --skip-duplicate doesn't consider target platforms [PR #873](https://github.com/microsoft/vscode-vsce/pull/873)

Contributions to inno-updater:

* [@panekj (Jakub Panek)](https://github.com/panekj): feat: replace unmaintained winapi with windows-sys [PR #18](https://github.com/microsoft/inno-updater/pull/18)

Contributions to language-server-protocol:

* [@c-claeys (Cristopher Claeys)](https://github.com/c-claeys): Add multirange formatting via textDocument/rangesFormatting to spec [PR #1759](https://github.com/microsoft/language-server-protocol/pull/1759)
* [@gundermanc (Christian Gunderman)](https://github.com/gundermanc): Semantic Tokens LSIF specification. [PR #1754](https://github.com/microsoft/language-server-protocol/pull/1754)
* [@semanticart (Jeffrey Chupp)](https://github.com/semanticart): Fix typo [PR #1774](https://github.com/microsoft/language-server-protocol/pull/1774)

Contributions to monaco-editor:

* [@claylibrarymarket](https://github.com/claylibrarymarket): Fix Twig's plain text class expression [PR #4063](https://github.com/microsoft/monaco-editor/pull/4063)
* [@FossPrime (Ray Foss)](https://github.com/FossPrime): Use new GitHub pages workflow [PR #4000](https://github.com/microsoft/monaco-editor/pull/4000)
* [@leandrocp (Leandro Pereira)](https://github.com/leandrocp): Elixir - Add support for multi-letter uppercase sigils [PR #4041](https://github.com/microsoft/monaco-editor/pull/4041)
* [@philippleidig (PhilippLe)](https://github.com/philippleidig): Add TwinCAT file support for structured text (st) language [PR #3315](https://github.com/microsoft/monaco-editor/pull/3315)
* [@remcohaszing (Remco Haszing)](https://github.com/remcohaszing)
  + Add mdx language [PR #3096](https://github.com/microsoft/monaco-editor/pull/3096)
  + Export custom TypeScript worker variables [PR #3488](https://github.com/microsoft/monaco-editor/pull/3488)
  + Document some basic concepts [PR #4087](https://github.com/microsoft/monaco-editor/pull/4087)