

Sidebar

Programming · VS Code Files and Auto Save Cheat Sheet

The `files.*` settings and their closest relatives: saving, tidying files on save, line endings and encoding, recovery, and which files are shown or watched. Exact ID, what it does, and its default. Sheet 2 in the VS Code series; sheet 1 covers `editor.*`.

AUTO SAVE

`files.autoSave` `off` (default), `afterDelay`, `onFocusChange`, `onWindowChange`

`files.autoSaveDelay` Milliseconds to wait with `afterDelay` (default 1000)

`files.autoSaveWorkspaceFilesOnly` `true`: only auto-save files inside the open workspace (default `false`)

`files.autoSaveWhenNoErrors` `true`: skip auto save while the file has errors (default `false`)

`File > Auto Save` Menu toggle that switches auto save on and off without opening settings

`Ctrl+S / Ctrl+K S` Save the current file / Save All (Windows/Linux)

TIDY UP ON SAVE

`files.trimTrailingWhitespace` Remove spaces at the end of lines on save (default `false`)

`files.insertFinalNewline` Ensure the file ends with a newline (default `false`)

`files.trimFinalNewlines` Remove extra blank lines at the end of the file (default `false`)

`files.trimTrailingWhitespaceInRegexAndStrings` `true` (default): trailing-space trimming also touches regexes and strings

`editor.formatOnSave` Run the formatter when you save (default `false`)

`editor.formatOnSaveMode` `file` (default), `modifications`, `modificationsIfAvailable`: whole file or only changed lines

`editor.codeActionsOnSave` Fixes on save, e.g. `{ "source.fixAll": "explicit" }`

LINE ENDINGS & ENCODING

`files.eol` Default line ending for new files: `"\n"`, `"\r\n"` or `"auto"` (default `auto`)

`files.encoding` Character encoding used when reading and writing files (default `utf8`)

`files.autoGuessEncoding` `true`: try to detect a file's encoding when opening it (default `false`)

`files.candidateGuessEncodings` Encodings to try first when guessing (default none)

`Status bar: CRLF / LF` Click to convert the current file's line endings

`Status bar: UTF-8` Click, then *Reopen with Encoding* or *Save with Encoding*

HOT EXIT & RECOVERY

`files.hotExit` `off`, `onExit` (default), `onExitAndWindowClose`: remember unsaved files between sessions

`files.restoreUndoStack` Restore undo history when a file reopens (default `true`)

`workbench.localHistory.enabled` Keep automatic snapshots of saved files (default `true`)

`workbench.localHistory.maxFileEntries` Snapshots kept per file (default 50)

`workbench.localHistory.maxFileSize` Largest file, in KB, that gets snapshots (default 256)

`Timeline view` Explorer panel listing a file's local history; right-click an entry to compare or restore

HIDING, WATCHING & PROTECTING FILES

- files.exclude** Glob patterns hidden from the Explorer; default hides `.git`, `.svn`, `.hg`, `.jj`, `.DS_Store`, `Thumbs.db`
- files.watcherExclude** Folders VS Code does not watch for changes; the default skips `.git/objects` and similar
- files.watcherInclude** Extra folders to watch (default none)
- files.associations** Map patterns to languages, e.g. `{ "*.cmd": "bat" }` (default none)
- files.readonlyInclude** Glob patterns of files to open read-only
- files.readonlyExclude** Exceptions to **files.readonlyInclude**
- files.enableTrash** `true` (default): deleting from the Explorer moves files to the Recycle Bin

CONFLICTS, TIMEOUTS & DIALOGS

- files.saveConflictResolution** `askUser` (default) or `overwriteFileOnDisk` when the file changed elsewhere
- files.participants.timeout** How long format/fix-on-save may run, in ms (default 60000)
- files.defaultLanguage** Language assigned to new, untitled files (default: none, you choose)
- files.dialog.defaultPath** Folder where Open/Save dialogs start (absolute path)
- files.simpleDialog.enable** `true`: use VS Code's own file dialog instead of the system one (default `false`)

A REASONABLE STARTER SET (MY SUGGESTIONS, NOT DEFAULTS)

- "files.autoSave": "onFocusChange"** Saves when you switch away, and still runs format on save
- "files.trimTrailingWhitespace": true** No invisible trailing spaces in commits
- "files.insertFinalNewline": true** A final newline keeps diffs and Unix tools happy
- "files.trimFinalNewlines": true** No pile of blank lines at the end
- "files.eol": "\n"** LF for new files, common for cross-platform projects
- "files.exclude": { "**/__pycache__": true }** Hide clutter you never open
- [markdown]** Override with `{ "files.trimTrailingWhitespace": false }`, because two trailing spaces mean a line break in Markdown

WHY DIDN'T FORMAT ON SAVE RUN? CHECK WHICH KIND OF SAVE IT WAS

Format on save is skipped for the timed auto save (`"files.autoSave": "afterDelay"`), but it does run when you press Ctrl+S and for the `onFocusChange` and `onWindowChange` auto saves. If you want auto save *and* formatting, use `onFocusChange`. Related: `editor.codeActionsOnSave` values are explicit (only on manual saves), `always`, or `never`.