

CURRENT STABLE COMMUNITY EDITION | VERIFIED 4 AUGUST 2026

Visual Studio Community 2026

Visual Cheat Sheet

What each screen, symbol and command does — with verified Windows shortcuts and safe workflows.

VISUAL COMMAND INDEX



READ THIS FIRST This is the full Visual Studio Community IDE for Windows — not Visual Studio Code. Visual Studio Community 2026 does not run natively on macOS. On a Mac, use supported Windows 11 in a virtual/cloud machine, or choose Visual Studio Code and a different cheat sheet.

What it is and what it does

Question	Precise answer
What is it?	A full integrated development environment (IDE): editor, project system, compiler/build tools, debugger, tests, profiler, Git and extensions in one application.
What does it build?	Windows, web, cloud, game, mobile and other applications. The available languages and project types depend on the workloads installed.
Is it free?	Yes for individual developers. Organisation use has licence limits: see the current Community licence terms before deploying it to a workplace.
What is not included automatically?	Every SDK, compiler, template and tool. Select the workloads required for the type of software you intend to build.

Five habits that prevent expensive mistakes

1. **Confirm the product:** The purple Visual Studio IDE and blue Visual Studio Code are different programs with different menus and shortcuts.
2. **Build before debugging:** Fix compiler errors first; a debugger cannot run an application that did not build successfully.
3. **Use a Git branch:** Keep each feature or fix separate, review the diff, then commit a small coherent change.
4. **Read the selected output source:** Error List summarises issues; Output usually contains the detailed build, debug or Git log.
5. **Check the current context:** Commands and shortcuts can change with the active window, language profile, project type, workload and keyboard mapping.

SYMBOL NOTE The large symbols in this guide are simplified recognition aids. Microsoft may restyle, recolour or move the actual controls between themes and updates; use the printed control name and menu path as the final check.

02 Start window and project containers

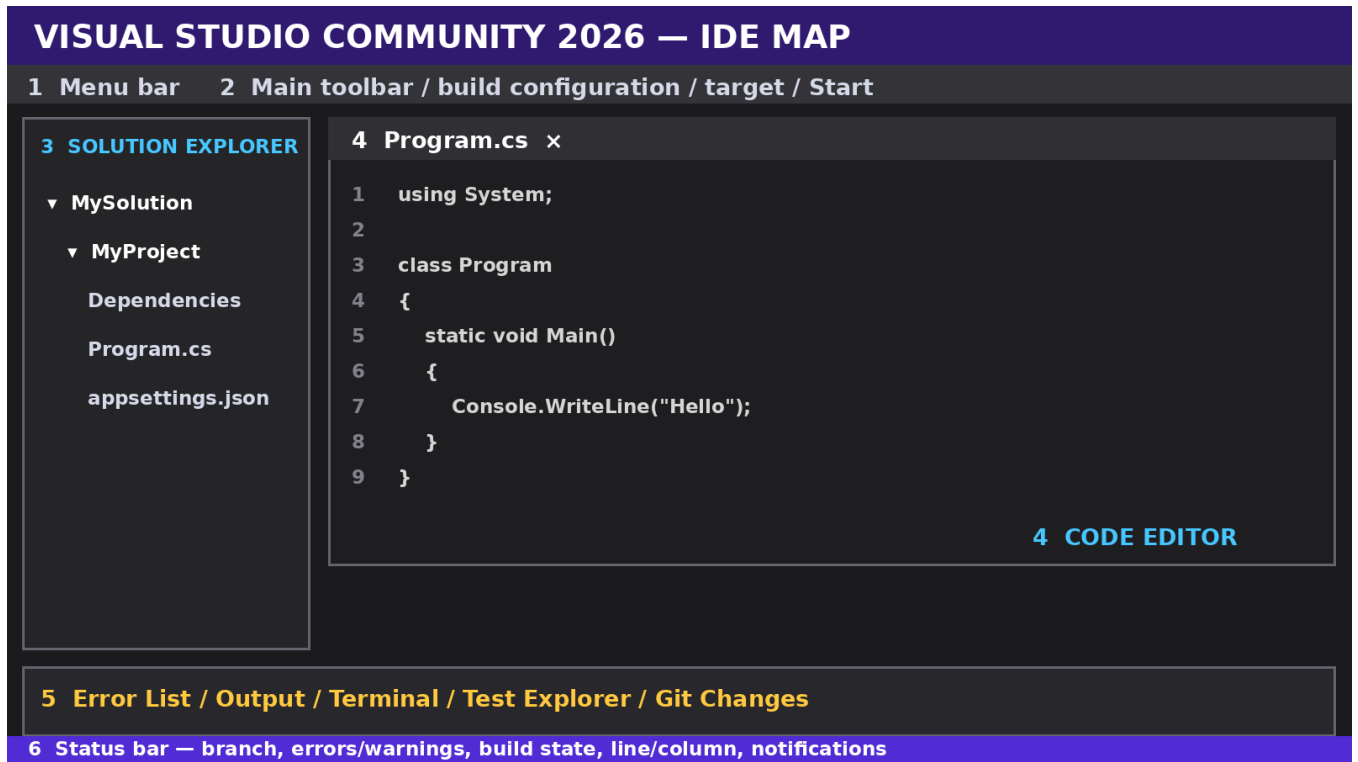
Choose how the code enters Visual Studio, then understand which container you are changing.

Symbol	Choice	May look like	What it does	Use it for
START WINDOW				
	Create a new project	window with a plus	Opens the template chooser, then creates a new project with starter files and settings.	Beginning a new application, library, service, test project or other supported project type.
	Clone a repository	branch/network symbol	Downloads a remote Git repository to a local folder and opens it in Git context.	Working on an existing GitHub, Azure DevOps or other Git repository.
	Open a project or solution	folder marked SLN	Loads an existing project/solution and its configured build system.	Opening a .sln/.slnx solution or a supported project file.
	Open a local folder	plain folder	Opens source directly without requiring a Visual Studio solution or project.	Folder-based work such as CMake, scripts or an existing code tree.
	Continue without code	empty editor window	Opens the IDE with no project loaded.	Changing settings, joining a session or attaching the debugger to a process.
CONTAINERS				
	Solution	stacked SLN pages	Top-level workspace that can organise one or more related projects.	Building, debugging and managing several projects together.
	Project	boxed P	Contains source, resources, references and build settings for one output or logical unit.	An app, library, test assembly or service.
	Source file	document	An individual code, configuration, markup or resource file.	Editing a specific piece of the project.

TEMPLATE VS WORKLOAD A template is the starter structure for one project type. A workload is an installer bundle that provides groups of templates, SDKs and tools for a development area.

03 The IDE screen

The editor is central; tool windows can be docked, floated, pinned or auto-hidden around it.



Area	What it does	Important detail
1 Menu bar	Opens every installed command category.	Available commands change with the active context and installed workloads.
2 Main toolbar	Shows configuration, target and common build/debug controls.	Confirm the startup target and Debug/Release choice before running.
3 Solution Explorer	Shows solution, projects, dependencies, folders and files.	Right-clicking different nodes produces different commands.
4 Code Editor	Edits code and displays IntelliSense, diagnostics and debugger markers.	A coloured squiggle is analysis; confirm build errors in Error List/Output.
5 Tool window area	Hosts Output, Error List, Terminal, Test Explorer, Git Changes and debugger windows.	Use the View, Debug, Test and Git menus to reopen missing windows.
6 Status bar	Shows current branch, errors/warnings, task state, line/column and notifications.	Items vary by project, source-control state and installed features.

LAYOUT RECOVERY If windows are lost or badly arranged, use Window > Reset Window Layout. This resets docking, not your project files.

04 Tool windows and panels

These windows explain the project, show problems or provide specialised controls. Open them from their named menu paths.

Symbol	Window	May look like	What it does	Open / use
PROJECT AND DESIGN				
	Solution Explorer	file tree	Displays and manages solutions, projects, dependencies, folders and files.	View > Solution Explorer Ctrl+Alt+L
	Properties	three sliders	Shows editable properties for the currently selected object, file or design item.	View > Properties Window F4
	Toolbox	tool case	Lists controls/components that can be dragged onto supported designers.	View > Toolbox Ctrl+Alt+X; contents depend on the active designer.
RESULTS AND COMMAND SURFACES				
	Error List	warning triangle	Aggregates errors, warnings and messages from builds and live analysis.	View > Error List Ctrl+, E
	Output	window with >_	Shows detailed text from the selected source, such as Build, Debug or Git.	View > Output Ctrl+Alt+O
	Terminal	window with >_	Runs a command-line shell inside Visual Studio.	View > Terminal; the shell/profile depends on your Windows setup.
	Test Explorer	laboratory flask	Discovers, groups, runs and debugs tests from supported test adapters.	Test > Test Explorer Ctrl+E, T
	Git Changes	branch with changes	Shows the working tree, commit message, included/excluded changes and push/pull actions.	Git > Git Changes Ctrl+0, G
	Status bar	coloured bottom strip	Shows compact state indicators and context-dependent actions.	Bottom of IDE; click an indicator only after reading its label/tooltip.

PIN AND CLOSE The pin toggles between docked and auto-hidden. The X closes only that tool window. Reopen it from the appropriate menu; neither action deletes project content.

05 Core file and IDE controls

Use project-aware commands for project content; use plain New File only for an unassociated file.

Symbol	Command	May look like	What it does	Menu / shortcut
CREATE AND OPEN				
	New Project	window with plus	Creates a project from a template.	File > New > Project Ctrl+Shift+N
	New Item	page with plus	Adds a new template-based item to the selected project.	Project > Add New Item Ctrl+Shift+A
	Add Existing Item	arrow into page	Adds an existing file to the selected project; exact behaviour depends on project type.	Project > Add Existing Item Shift+Alt+A
	Open	folder	Opens a file, project, solution or folder through the File menu.	File > Open; Ctrl+O file, Ctrl+Shift+O project/solution
SAVE AND REVERSE				
	Save	single disk	Saves the selected/current document or item.	Ctrl+S
	Save All	stacked disks	Saves every modified document/item in the current IDE session.	Ctrl+Shift+S
	Undo	left curved arrow	Reverses the most recent supported editor/designer action.	Ctrl+Z; history is usually document/context specific.
	Redo	right curved arrow	Reapplies the most recently undone supported action.	Ctrl+Y
FIND AND CONFIGURE				
	Search / Feature Search	magnifying glass	Finds files, symbols, commands, settings or features according to the search surface used.	Ctrl+T or Ctrl+, for Go to All; Ctrl+Q for Feature Search.
	Settings	gear	Opens IDE configuration such as editor, keyboard, themes, build and source control.	Tools > Options

SAVE IS NOT SOURCE CONTROL Saving writes the working file to disk. A Git commit records a local versioned snapshot. A push sends commits to a remote repository. These are three separate actions.

06 Solution Explorer

Select the correct node before using Add, Properties, Build or Set as Startup Project.

Symbol	Node/action	May look like	What it does	Use carefully
HIERARCHY				
	Solution	stacked SLN pages	Organises the projects loaded together.	Right-click for solution-wide build, add and configuration commands.
	Project	boxed P	Defines one buildable/deployable unit and its settings.	Right-click to build, add items, manage packages or set startup.
	Dependencies	connected package	Groups project references, framework/SDK references, packages and other dependencies.	Expand it to inspect what the project relies on.
	Source file	document	Opens in the editor or an appropriate designer.	Double-click to open; F2 renames a selected tree item when supported.
TOOLBAR AND CONTEXT ACTIONS				
	Solution Explorer	tree window	Searches/manages the current hierarchy and offers view/filter controls.	Ctrl+Alt+L
	Properties	sliders	Shows the selected item's immediate properties.	F4; project Property Pages can also be opened by right-clicking the project.
	New Item	page plus	Creates a new project item from an installed template.	Select the intended project/folder first.
	Add Existing Item	arrow into page	Includes an existing file through the project system.	Do not confuse adding a file with adding a project reference.

Four different ways to add something

Need	Use	Result
Create a new code/configuration file	Project > Add New Item	A file generated from a selected item template.
Include an existing file	Project > Add Existing Item	The project includes or links the selected file according to project type/options.
Use another project in this solution	Add > Project Reference	A build/reference dependency on the selected project.
Use a published library	Manage NuGet Packages	A package reference plus restore metadata in the project.

STARTUP PROJECT F5 launches the configured startup target. If the wrong program runs, right-click the intended executable project and choose Set as Startup Project, or configure multiple startup projects in solution properties.

07 Code Editor: write, navigate and refactor

Editor features are language-service dependent; not every language/project offers every command.

Symbol	Feature	May look like	What it does	Shortcut / caution
WRITE AND FIX				
	IntelliSense	completion list	Suggests symbols, members, parameters and documentation as you type.	Ctrl+Space completes a word; Ctrl+J lists members.
	Quick Actions	light bulb	Offers context-aware fixes and refactorings for the caret/selection.	Ctrl+. then inspect the preview before applying.
	Rename	tag marked Aa	Renames a supported symbol and updates discovered references.	Ctrl+R, Ctrl+R; review the preview/scope.
	Format	braces with sparkle	Applies the language formatter to a document or selection.	Ctrl+K, Ctrl+D document; Ctrl+K, Ctrl+F selection.
	Comment	double slash	Comments or uncomments selected lines using the language's syntax.	Ctrl+K, Ctrl+C / Ctrl+K, Ctrl+U; Ctrl+J toggles line comments in current releases.
	Code folding	minus boxes	Collapses or expands structural code regions without deleting code.	Ctrl+M, Ctrl+M toggles the current outlining region.
UNDERSTAND CODE				
	Go to Definition	arrow into code page	Opens the declaring definition when the language service can resolve it.	F12
	Peek Definition	small code panel	Shows the definition inline without leaving the current file.	Alt+F12
	Find All References	linked nodes	Lists discovered uses of a symbol across the loaded scope.	Shift+F12

CHORDS A shortcut written Ctrl+K, Ctrl+D is sequential: press Ctrl+K, release K, then press Ctrl+D. It is not one four-key press.

08 Build, configuration and run

A build turns source and project settings into outputs. Running starts the selected startup target.

Symbol	Control	May look like	What it does	Menu / caution
CONFIGURATIONS				
	Debug configuration	DBG selector	A named property set normally configured for easier debugging and lower optimisation.	Actual settings are defined by the project; Debug is a name, not a guarantee.
	Release configuration	REL selector	A named property set normally configured for deployment/performance.	Release builds can still contain symbols; inspect project properties.
BUILD COMMANDS				
	Build Solution	hammer	Builds the solution for the selected configuration, compiling required/out-of-date targets.	Build > Build Solution Ctrl+Shift+B
	Rebuild Solution	hammer with circular arrow	Cleans and then builds the selected solution targets from scratch.	Use when stale outputs are suspected; it is slower than Build.
	Clean Solution	eraser	Removes generated build outputs for the selected configuration/targets.	It does not delete source files or Git history.
RUN COMMANDS				
	Start Debugging	green play with bug	Builds if required, starts the startup target and attaches the debugger.	Debug > Start Debugging F5
	Start Without Debugging	outlined green play	Builds if required and launches without attaching the debugger.	Debug > Start Without Debugging Ctrl+F5
	Stop Debugging	red square	Ends the current debug session and normally terminates the debugged target.	Shift+F5

Stage	If it fails	Where to look
Restore	Required packages/SDKs cannot be obtained.	Output > Package Manager / build output; package sources and network.
Compile/build	Source or project configuration is invalid.	Error List summary, then Output > Build for full details.
Launch	Executable/target, permissions or environment is wrong.	Debug output, launch profile, startup project and target selector.
Run	A runtime exception or logic error occurs.	Exception Helper, debugger windows, logs and tests.

09 Debugger controls and symbols

Breakpoints suspend before the marked statement executes; the yellow instruction pointer marks the next statement.

Symbol	Control	May look like	What it does	Shortcut / note
SESSION CONTROLS				
	Start Debugging	green play with bug	Starts or continues under the debugger.	F5
	Pause / Break All	yellow pause bars	Requests a break in the running debugged processes so state can be inspected.	Ctrl+Alt+Break
	Stop Debugging	red square	Stops the debug session.	Shift+F5
	Restart	circular arrow	Stops and starts the current debug target again.	Ctrl+Shift+F5
BREAK AND STEP				
	Breakpoint	solid red circle	Suspends execution just before the marked statement runs, when the breakpoint is bound and enabled.	F9 toggles; appearance/status can vary, so check the tooltip or Breakpoints window.
	Current statement	yellow arrow	Marks the next statement the debugger intends to execute while paused.	It is not an error marker.
	Step Over	curved arrow over dot	Executes the current statement; called functions run without stepping through them.	F10
	Step Into	down arrow to dot	Executes the current statement and enters a called function when debuggable.	F11
	Step Out	up arrow from dot	Runs until the current function returns and pauses in its caller.	Shift+F11
	Run to Cursor	arrow to vertical line	Runs to the selected code location without creating a persistent breakpoint.	Ctrl+F10

SAFE DEBUG LOOP Set one breakpoint (F9) > start (F5) > inspect values > step over (F10) or into (F11) > continue (F5) > stop (Shift+F5). Change one thing, then repeat.

10 Debugger windows

Many debugger windows are useful only while attached; variable evaluation generally requires break mode.

Symbol	Window	May look like	What it does	Open / shortcut
VARIABLES AND EXPRESSIONS				
	Locals	window showing x=42	Displays local variables in the current scope/frame.	Debug > Windows > Locals Ctrl+Alt+V, L
	Watch	eye in a window	Evaluates expressions that you choose and keeps them visible while stepping.	Debug > Windows > Watch > Watch 1 Ctrl+Alt+W, 1
	Immediate	window showing ?=	Evaluates expressions or statements in the current debug context when supported.	Debug > Windows > Immediate Ctrl+Alt+I
EXECUTION STRUCTURE				
	Call Stack	stacked frames	Shows the chain of active function/method calls for the selected thread.	Debug > Windows > Call Stack Ctrl+Alt+C
	Breakpoints window	red dots and rows	Enables, disables, deletes and configures breakpoints, conditions and hit settings.	Debug > Windows > Breakpoints Ctrl+Alt+B
	Exception Settings	warning triangle	Controls which exception types make the debugger break when thrown or unhandled.	Debug > Windows > Exception Settings Ctrl+Alt+E
	Modules	connected package	Lists loaded binaries/modules and their symbol-loading status.	Debug > Windows > Modules Ctrl+Alt+U
	Threads	branch/network	Shows application threads and lets you select or inspect them.	Debug > Windows > Threads Ctrl+Alt+H

When a value looks wrong

Check	Why
Selected stack frame	Locals and expression evaluation use the current frame/context.
Optimised code	Optimisation can move/remove variables or change stepping behaviour.
Loaded symbols	Without matching symbols/source, breakpoints and source-level stepping may not bind correctly.
Side effects	Some property getters or expressions execute code when evaluated; do not assume inspection is passive.

11 Errors, tests and diagnostics

Use the summary to locate the problem, then the detailed log or debugger to understand it.

Symbol	Tool	May look like	What it does	Use / caution
FIND THE CAUSE				
	Error List	red warning triangle	Summarises compiler/build and live-analysis errors, warnings and messages.	Filter by source/project carefully; an IntelliSense item is not automatically a confirmed build failure.
	Output	text console window	Shows detailed ordered text from Build, Debug, Test, Git and other providers.	Choose the correct source from the Output dropdown before reading.
	Exception Settings	yellow warning triangle	Controls debugger breaking behaviour for exception categories.	Do not simply suppress an exception you have not understood.
	Profiler	gauge	Collects performance/diagnostic data using the selected tool and launch mode.	Debug > Performance Profiler Alt+F2; available tools depend on project and installed components.
TEST CODE				
	Test Explorer	flask	Discovers and displays supported tests with outcome, duration and grouping/filtering.	Ctrl+E, T
	Run tests	flask with play	Runs tests without debugger attachment.	Ctrl+R, A runs all; Ctrl+R, T runs tests in context.
	Debug tests	flask with red dot	Runs tests with the debugger so breakpoints can be hit.	Ctrl+R, Ctrl+A all; Ctrl+R, Ctrl+T in context.

Signal	Meaning	Correct response
Compiler/build error	The build did not produce the required output.	Open the first relevant error, then inspect Output > Build.
Warning	A potential problem; it may not block a build unless warnings are treated as errors.	Understand it; do not dismiss warnings by default.
Runtime exception	The program failed while executing a path.	Read exception type/message, stack trace and inner exception; inspect state.
Failed test	Actual result or execution did not meet the test.	Open the test result/output; reproduce and debug the smallest failing test.
Analyzer suggestion	A rule-based code-quality/style finding with configured severity.	Use Quick Actions only after reviewing the proposed change.

12 Git source control

Saving changes a file; committing records locally; pushing sends commits to a remote repository.

Symbol	Action	May look like	What it does	Use / caution
REPOSITORY AND LOCAL WORK				
	Clone a repository	branch/network with arrow	Creates a local copy of a remote Git repository.	Start window > Clone a repository.
	Git Changes	branch with plus/minus	Shows modified/untracked files and the commit surface for the working tree.	Git > Git Changes Ctrl+0, G
	Git Repository	branch/network with arrows	Shows branches and commit history for local and remote refs.	Git/View menu or status bar; Manage Branches.
	Branch	forked line	Keeps one line of work separate from another until integrated.	Create a branch for each focused feature/fix Ctrl+0, N opens branches.
	Commit	branch with tick	Records a local snapshot of the selected/included changes with a message.	Review every diff; use a specific message.
REMOTE AND INTEGRATION				
	Fetch	two horizontal arrows	Downloads updated remote references without integrating them into the current branch/worktree.	Safe way to see what changed remotely.
	Pull	down arrow	Fetches, then integrates remote changes into the current branch according to Git configuration.	Commit/stash local work first when necessary; resolve conflicts deliberately.
	Push	up arrow	Sends local commits to the configured remote branch.	A push is not the same as saving or committing.
	Merge	two lines joining	Combines histories/changes from branches.	Build and test after the merge.
	Merge conflict	cross over branch	Git cannot automatically reconcile overlapping changes.	Choose/edit the correct final content, mark resolved, then complete the merge/rebase.

SAFE GIT LOOP Fetch/pull with a clean or safely stashed working tree > create/switch branch > edit > build/test > review diff > commit > pull/reconcile if needed > push.

13 Workloads, packages and extensions

These install different kinds of capability. A missing command or template often means the required component is not installed.

Symbol	Feature	May look like	What it does	Menu / caution
VISUAL STUDIO INSTALLATION				
	Workloads	four boxes with tick	Installer bundles for development areas, such as .NET desktop, ASP.NET/web or Desktop development with C++.	Tools > Get Tools and Features > Workloads > select > Modify.
	Extensions	puzzle/plus symbol	Optional add-ins that extend the IDE.	Extensions > Manage Extensions; assess publisher, permissions, support and update status.
	Settings	gear	Configures installed IDE behaviour without installing a new workload.	Tools > Options; search the options tree.
PROJECT DEPENDENCIES				
	NuGet package	cube marked N	A versioned dependency obtained from configured package sources and referenced by a project.	Right-click project > Manage NuGet Packages; review licence and vulnerabilities.
	Dependencies	connected package	Solution Explorer grouping for packages, frameworks, SDKs and project references.	Expand to identify the exact reference and version/state.
	Project	boxed P	Can reference another compatible project in the solution.	Use Add Project Reference when source projects should build together.
AI INSIDE VISUAL STUDIO				
	GitHub Copilot	robot face	Microsoft's in-IDE AI assistance surface for code/chat/agent features in supported current installs.	Availability depends on sign-in, plan, update, workload and organisation policy; it is separate from this ChatGPT chat.

When a command, template or designer is missing

1. **Check context:** Select the correct project/file/designer; many commands appear only for a relevant context.
2. **Search the command:** Use Feature Search (Ctrl+Q) and note the exact command name.
3. **Check the workload:** Tools > Get Tools and Features, then confirm the required workload/individual component.
4. **Update safely:** Community is supported on the latest Stable release; save/commit work and update through Visual Studio Installer.
5. **Check compatibility:** Project target, SDK version, architecture and extension version can all limit availability.

14 Windows shortcuts: create, save and edit

Defaults below use Visual Studio's General profile. Keyboard layout, language, active context and custom mappings can change them.

Files and projects

Command	Default shortcut	What it does / exact note
New file	Ctrl+N	Creates a new unassociated file.
New project	Ctrl+Shift+N	Opens the new-project template dialog.
Add new item	Ctrl+Shift+A	Adds a template-based item to the selected project.
Add existing item	Shift+Alt+A	Adds an existing file through the selected project.
Open file	Ctrl+O	Opens an individual file.
Open project/solution	Ctrl+Shift+O	Opens a supported project or solution.
Save selected item	Ctrl+S	Saves the active/selected document or item.
Save all	Ctrl+Shift+S	Saves all modified documents/items.
Close document	Ctrl+F4	Closes the active document tab. Ctrl+W is not assigned to this by default in the Visual C# profile.
Exit Visual Studio	Alt+F4	Closes the application after any save prompts.

Editing and finding

Command	Default shortcut	What it does / exact note
Cut / Copy / Paste	Ctrl+X / Ctrl+C / Ctrl+V	Standard clipboard operations in the active context.
Undo / Redo	Ctrl+Z / Ctrl+Y	Reverses or reapplies supported actions in the active context.
Find / Replace	Ctrl+F / Ctrl+H	Searches or replaces in the current scope/document surface.
Find / Replace in files	Ctrl+Shift+F / Ctrl+Shift+H	Searches or replaces across the chosen file scope; review replace scope carefully.
Find next / previous	F3 / Shift+F3	Moves through matches.
Complete word	Ctrl+Space	Invokes completion for the current language service.
List members	Ctrl+J	Shows members/completion list when supported.
Parameter info	Ctrl+Shift+Space	Shows parameter information for the current call when supported.
Quick Actions	Ctrl+.	Opens context fixes/refactorings.
Toggle line comment	Ctrl+/	Toggles line comments in current Visual Studio releases; language/keyboard layout can affect it.
Comment / uncomment	Ctrl+K, Ctrl+C / Ctrl+K, Ctrl+U	Sequential chords for the selection.
Format document / selection	Ctrl+K, Ctrl+D / Ctrl+K, Ctrl+F	Runs the language formatter.

VERIFY ANY SHORTCUT Tools > Options > Environment > Keyboard > search the command ID/name. This is the authoritative answer for your installed profile and keyboard.

15 Windows shortcuts: navigate, refactor and open windows

Letter O and number 0 are called out where confusion is common. Commas indicate sequential chords.

Navigate and refactor

Command	Default shortcut	What it does / exact note
Go to All	Ctrl+T or Ctrl+,	Searches files, types, members and symbols according to filters/context.
Go to line	Ctrl+G	Moves the caret to a line number.
Go to Definition	F12	Opens the resolved declaration/definition.
Peek Definition	Alt+F12	Shows the resolved definition inline.
Find All References	Shift+F12	Lists discovered references.
Navigate backward / forward	Ctrl+- / Ctrl+Shift+-	Moves through code navigation history.
Next / previous document	Ctrl+Tab / Ctrl+Shift+Tab	Shows document navigation and moves between open documents.
Rename symbol	Ctrl+R, Ctrl+R	Runs supported semantic rename.
Extract method	Ctrl+R, Ctrl+M	Runs supported extract-method refactoring.
Toggle outlining region	Ctrl+M, Ctrl+M	Collapses or expands the current code region.
Collapse to definitions	Ctrl+M, Ctrl+O	Collapses bodies to definition-level outlines.
Next / previous error location	F8 / Shift+F8	Moves through Error List or Output locations.

Open common windows

Command	Default shortcut	What it does / exact note
Solution Explorer	Ctrl+Alt+L	Opens/focuses Solution Explorer.
Properties Window	F4	Opens/focuses Properties for the selection.
Error List	Ctrl+I, E	Sequential chord: Ctrl+backslash, then E.
Output	Ctrl+Alt+O	O is the letter O.
Toolbox	Ctrl+Alt+X	Opens Toolbox; contents depend on active designer.
Test Explorer	Ctrl+E, T	Sequential chord.
Full screen	Shift+Alt+Enter	Toggles full-screen IDE view.
Feature Search	Ctrl+Q	Searches commands, settings and features. Current releases also support Ctrl+Shift+P as a familiar alternative.

16 Windows shortcuts: build, debug, test and Git

Function keys may require Fn or Fn Lock on some keyboards.

Build and debug

Command	Default shortcut	What it does / exact note
Build Solution	Ctrl+Shift+B	Builds the selected solution configuration.
Start / Continue	F5	Starts debugging, or continues when paused.
Start without debugging	Ctrl+F5	Launches without debugger attachment.
Stop debugging	Shift+F5	Ends the debug session.
Restart	Ctrl+Shift+F5	Restarts the current debug target.
Toggle breakpoint	F9	Adds/removes a breakpoint at the current line when supported.
Delete all breakpoints	Ctrl+Shift+F9	Deletes every breakpoint — use carefully.
Step Over / Into / Out	F10 / F11 / Shift+F11	Controls statement-by-statement execution.
Run to Cursor	Ctrl+F10	Runs to the selected location without a persistent breakpoint.
Break All	Ctrl+Alt+Break	Requests a pause in running debugged processes.
Attach to Process	Ctrl+Alt+P	Opens the process attachment dialog; choose code type/process carefully.
QuickWatch	Shift+F9 or Ctrl+Alt+Q	Inspects one expression in break mode.
Locals / Watch 1	Ctrl+Alt+V, L / Ctrl+Alt+W, 1	Sequential debugger-window chords.
Call Stack / Immediate	Ctrl+Alt+C / Ctrl+Alt+I	Opens the named debugger window.
Breakpoints / Exceptions	Ctrl+Alt+B / Ctrl+Alt+E	Opens breakpoint or exception settings windows.

Tests and Git

Command	Default shortcut	What it does / exact note
Run all tests	Ctrl+R, A	Sequential chord; runs all discovered tests.
Debug all tests	Ctrl+R, Ctrl+A	Sequential chord; runs discovered tests with debugger.
Run tests in context	Ctrl+R, T	Runs tests relevant to the current selection/context.
Debug tests in context	Ctrl+R, Ctrl+T	Debugs tests relevant to the current context.
Git Changes	Ctrl+0, G	First key is number zero; opens Git Changes.
Git Branches	Ctrl+0, N	First key is number zero; opens branches.
Git Commits	Ctrl+0, O	First key is number zero; final key is letter O; opens commits/history.

17 Compatibility, ChatGPT help and verification

Use this page when the screen, shortcut or available feature does not match the guide.

Platform and edition facts

Fact	Precise position as checked 4 August 2026
Current product	Visual Studio Community 2026 on the latest Stable channel. Community is supported only on the latest Visual Studio release.
Supported desktop OS	Visual Studio 2026 supports 64-bit Windows 11 (including supported ARM64 editions) and listed Windows Server editions. It is not a native macOS application.
Mac	Visual Studio for Mac was retired on 31 August 2024. Microsoft recommends Visual Studio on Windows or Visual Studio Code on Mac; Windows 11 VM/cloud options are possible, with ARM64 workload limitations.
Community licence	Individuals may use Community for free or paid apps. Organisation use is restricted; verify the current licence terms, especially for enterprise or more than five non-enterprise users.
Feature availability	Varies by workload, project type, target, architecture, extension, account/licence, organisation policy and update.

How this ChatGPT/Codex can help

Symbol	Surface	May look like	What help is possible	What to provide
WITH PROJECT FILES OR A REPOSITORY				
	GitHub Copilot	robot face	For clarity: the in-IDE GitHub Copilot product is separate from this ChatGPT/Codex conversation.	Use its own sign-in/licence and Visual Studio controls.
	Code Editor	code window	This ChatGPT/Codex can explain, create or edit code and project files that you attach or make available in its workspace.	Provide the relevant repository/files and state the exact desired result.
	Error List	warning triangle	It can diagnose compiler/runtime/test errors from copied text, logs, screenshots and project context.	Include the first complete error, Output details, language/project type and what changed.
	Test Explorer	flask	It can write or revise tests and run them when the provided working environment supports the required tools.	It cannot assume your local Windows installation or private dependencies are available.

DIRECT CONTROL LIMIT This chat cannot see or click your local Visual Studio window, install workloads on your PC, or read private project files unless you explicitly share/connect the relevant files or workflow. A screenshot plus copied Output/Error List text is the fastest diagnostic hand-off.

Official Microsoft verification links

- [Visual Studio Community — features and licence summary](#)
- [Visual Studio 2026 release history](#)
- [Visual Studio 2026 system requirements](#)
- [Tour the Visual Studio IDE](#)
- [Default Visual Studio keyboard shortcuts](#)
- [Solution Explorer](#)
- [Debugger code navigation](#)
- [Git in Visual Studio](#)
- [Modify workloads and components](#)
- [Visual Studio for Mac retirement](#)

FINAL CHECK If a command differs on your machine, use Tools > Options > Environment > Keyboard and Feature Search. The installed command mapping and the selected context override any printed cheat sheet.