MCZ · MCP SERVERS

Visual Studio bridge manual

Visual Studio bridge 1.9.0 · 15 November 2026 · https://mcz.software/Mcp

Contents

1. How it works
2. Requirements
3. Install and uninstall
4. First use
5. Tools
6. Several sessions at once
7. Settings
8. Licence
9. Troubleshooting
10. Logs

The Visual Studio bridge 1.9.0 lets Claude build, test and debug in Visual Studio: it builds and follows the build log while it compiles, reads the Error List and Output panes, runs tests, drives the debugger and checks for unsaved documents. Several Claude sessions can use it at once, each in its own Visual Studio window. New to the MCP servers? Read Getting started first.

How it works

- A Visual Studio **extension** runs a small MCP server inside every Visual Studio window (ports 8765 and up, one per window, up to 25 windows).
- Claude starts the **bridge** (`VsMcpBridge.exe`) for each session. All sessions share one bridge service on `127.0.0.1:8764`, which knows every open window and every session, and routes each session's calls to its window.
- Everything stays on this computer. Calls between the bridge and Visual Studio carry a private token from `%APPDATA%\VsMcp\token.txt`, so other programs cannot drive Visual Studio.

Requirements

- Windows, 64-bit.
- Visual Studio 2022 or 2026 (Community, Professional or Enterprise; preview builds too).
- .NET Framework 4.8 (part of Windows).
- No administrator rights.

Install and uninstall

Close Visual Studio, then run `VsMcp-Setup-1.9.0.exe`. It installs for your Windows account only:

- the extension into **every** Visual Studio 2022/2026 on the PC,
- the bridge and its files into `%APPDATA%\VsMcp`,
- a token in `%APPDATA%\VsMcp\token.txt` (an existing one is kept),
- the MCZ MCP Manager, if needed.

Setup waits (Retry / Cancel) while Visual Studio is open. With **Add to Claude** ticked it registers the server as `visualstudio` in Claude Desktop, and in Claude Code's settings (`%USERPROFILE%\.claude.json`) if Claude Code is installed:

```
"visualstudio": {
  "command": "C:\\Users\\YourName\\AppData\\Roaming\\VsMcp\\VsMcpBridge.exe",
  "args": []
}
```

Restart Claude Desktop and Visual Studio afterwards. The setup log is `%APPDATA%\VsMcp\setup.log`.

Uninstall from **Settings > Apps** (close Visual Studio first): it removes the extension from every Visual Studio, the `visualstudio` entry (if it points at the bridge) and the token. Licence keys stay in `%LOCALAPPDATA%\MCZ\products`.

For Claude Code by hand:

```
claude mcp add --scope user visualstudio -- "%APPDATA%\VsMcp\VsMcpBridge.exe"
```

First use

Open the solution in Visual Studio (or let the bridge open it), then ask Claude:

- "*Build the solution and tell me the errors.*"
- "*Run the unit tests in MyApp.Tests and show me the failures.*"
- "*Set a breakpoint at OrderService.cs line 120, start debugging and show me the locals when it stops.*"
- "*Are there unsaved files in Visual Studio? Save them.*"

The 30-day trial starts when you install the server on a computer.

Tools

37 tools as Claude sees them.

Group	Tools
Build	`build`, `build_wait`, `build_status`, `build_cancel`, `get_build_log`
Errors and output	`get_errors`, `get_output`, `read_log`
Tests	`run_tests`, `test_status`
Debugger	`debug_state`, `debug_start`, `debug_continue`, `debug_break`, `debug_stop`, `debug_attach`, `debug_step`, `debug_wait`, `get_locals`, `threads`, `select_frame`, `evaluate`, `exception_settings`
Breakpoints	`breakpoints`
Solution and settings	`solution_info`, `open_solution`, `close_solution`, `set_configuration`, `set_startup_project`, `execute_command`, `instances`

Group	Tools
Documents	`documents`, `open_file`, `save_all`
Session	`vs_session`
Licence	`license_status`, `license_install`

Notes:

- `build` builds, rebuilds or cleans the solution or one project and waits about 50 seconds; if the build is still running, Claude follows it with `build_wait` and `get_build_log` (which can continue from a line number).
- `get_output` reads any Output pane (with no pane named it lists them).
- `run_tests` builds and runs the test projects (MSTest, xUnit, NUnit, C++), optionally one project or a filter; `test_status` follows, waits for or cancels the run.
- `debug_wait` waits until the debugger stops, runs or ends, or until a line appears in the output.
- `execute_command` runs any Visual Studio command by name.
- `instances` lists the other open Visual Studio windows.

Several sessions at once

The bridge decides which window a session uses from the session's **folder**:

1. `VSMCP_WORKDIR`, if set;
2. otherwise the folders Claude reports for the session (Claude Code sends its working folder);
3. otherwise the folder the bridge was started in.

It then picks the window whose solution is in (or up to three folders below) that folder. If no window has it, it looks for a `.sln` or `.slnx` in the folder (three levels up, two down) and **starts Visual Studio** on it (the newest installed version), waiting up to four minutes. If several solutions are found it names them so you can choose.

- **One session per solution.** A second session asking for the same solution gets *"Already in use: solution 'X' is routed to another Claude session (owner, folder, since ...)"*. A session's claim ends when it closes, or after 30 minutes without calls.
- A `.sln` and a `.slnx` with the same name count as one solution.
- Claude Desktop sessions have no project folder of their own: use `vs_session` to choose a window, or set `VSMCP_WORKDIR`.

`vs_session` actions:

Action	What it does
`status` (default)	Shows the windows, the sessions and which window this session uses, and why.
`select`	Use the window with this solution (`solution=`).
`open`	Open this solution in a new Visual Studio window and use it.
`auto`	Go back to choosing by folder.
`release`	Let go of the window, so another session can use it.

Action	What it does
`close`	Close a window the bridge started (optionally saving with `save=true`).

Settings

Environment variables (set them in the `env` section of the Claude configuration entry, or system-wide):

Variable	Default	Meaning
`VSMCP_WORKDIR`	(none)	Folders to route by, separated by `;`.
`VSMCP_OWNER`	"Claude session in (pid N)"	The name other sessions see in "Already in use".
`VSMCP_DEVENV`	newest Visual Studio	The `devenv.exe` the bridge starts.
`VSMCP_PORT`	8765	First port of the Visual Studio windows.
`VSMCP_BRIDGE_PORT`	8764	Port of the bridge service.
`VSMCP_TOKEN`	`token.txt`	Use this token instead of the file.

The bridge service stops by itself after 10 minutes without sessions; the next session starts it again.

Licence

30 days free from the first tool call on a computer, then a key. `license_status` and `license_install` always work. From a prompt: `VsMcpBridge.exe --license` shows the state (exit code 0 when the tools work, 2 when not); `VsMcpBridge.exe --license MCZ1-...` installs a key. See Getting started and the licence terms.

Troubleshooting

- **What does the bridge see?** Run `VsMcpBridge.exe --list` in the project folder: it shows the open windows, the sessions and the window this folder would use.
- **Visual Studio does not answer**: check the **Claude MCP** pane in Visual Studio's Output window; it shows the address the extension listens on. If it is missing, the extension is not installed or enabled in that Visual Studio (Extensions > Manage Extensions).
- **HTTP 403 / wrong token**: the token changed after Visual Studio started. Restart Visual Studio.
- **"Already in use"**: another session holds that solution. Close it there, or `vs_session release` in that session. A session started in a folder *above* the solution can take it first; release it there.
- **A build takes longer than a minute**: normal. Claude carries on with `build_wait`.
- **The server is not listed in Claude**: see Troubleshooting in Getting started.

Logs

- Bridge service: `%LOCALAPPDATA%\VsMcp\bridge.log` (kept under 2 MB; the previous one is `bridge.log.old`). With the Microsoft Store version of Claude Desktop the file is under `%LOCALAPPDATA%\Packages\Claude_<id>\LocalCache\Local\VsMcp\`.
- Per-session bridge messages go to standard error, which Claude keeps in its MCP log (`%APPDATA%\Claude\logs\mcp-server-visualstudio.log`).

- Inside Visual Studio: Output window, pane **Claude MCP**.
- Setup: `%APPDATA%\VsMcp\setup.log`.