MCZ · MCP SERVERS

Getting started with the MCZ MCP servers

Git MCP, Visual Studio bridge, Blender bridge, PC Hardware MCP, Unreal bridge, Android Studio bridge · 15 November 2026 · https://mcz.software/Mcp

Contents

1. What an MCP server is
2. How the six servers work
3. Free trial and licences
4. What the installers do to Claude's configuration
5. Editing claude_desktop_config.json by hand
6. Claude Code
7. Troubleshooting

This guide covers what the six MCZ MCP servers are, how they connect to Claude, licences, the Claude Desktop configuration file, and what to do when something does not work. Each server also has its own manual: Git MCP, Visual Studio bridge, Blender bridge, PC Hardware MCP, Unreal bridge and Android Studio bridge.

What an MCP server is

MCP (the Model Context Protocol) is the way an AI assistant such as Claude uses tools on your computer. An MCP server is a small program that offers Claude a list of tools, for example "build the solution", "list the objects in this Blender scene" or "show the git status of this folder". Claude decides when to call a tool, the server does the work on your PC, and the result goes back to Claude. Nothing is sent anywhere else: the servers run locally and talk only to Claude and to programs on the same computer.

Claude asks before it uses a tool for the first time in a conversation (depending on your Claude settings), so you stay in control of what it does.

How the six servers work

Claude starts each server as a small local program and talks to it over standard input and output ("stdio"). What happens next depends on the server:

Server	What it talks to
Git MCP	Runs the `git` command line on your PC, in the repository folder Claude names.
Visual Studio bridge	A Visual Studio extension inside each open Visual Studio window.
Blender bridge	A Blender add-on inside each running Blender.
Unreal bridge	An editor plugin inside each running Unreal Editor.
Android Studio bridge	A plugin inside Android Studio.
PC Hardware MCP	Reads hardware information itself, or through an optional local sensor service that only reads.

The four **bridges** (Visual Studio, Blender, Unreal, Android Studio) control an editor you have open. Several Claude sessions can use a bridge at the same time: each session is routed to its own editor window (or project), so two chats never type into the same editor. How a session finds its window differs a little per editor; the server manuals explain it. The editor must be running with its extension, add-on or plugin loaded (most bridges can also start the editor for you).

Free trial and licences

Every server works fully for **30 days** on each computer, from the day you install it there. Each server has its own trial. (Installing it again, or installing another server, does not restart a trial. A server copied onto a computer without its installer starts its trial at its first tool call instead.) After that it needs a licence key, whether you use it at home or at work: there is no free personal use of the MCP servers.

- **Personal licence**: one computer, no expiry, for your own non-commercial use. Not for a computer that belongs to an organisation or is joined to a Windows domain or Microsoft Entra ID.
- **Commercial licence**: one computer, one year, renews automatically until you cancel. Needed for any work use and on any domain- or Entra-joined computer. After the last day the server keeps working for 30 more days, then stops until a renewed key is installed.
- **Bundle**: one key for all six servers on one computer, personal or commercial.

Each key is tied to the computer's **Machine ID**, the same ID that MCZ shows. Prices and terms: mcz.software/Mcp and the MCP servers licence terms.

Checking and installing a licence

Every server has two tools that keep working after the trial:

- `license_status` shows whether the server is licensed or in its trial (and the days left), this computer's Machine ID and the link to buy a licence. Ask Claude: "*What is the licence status of the Visual Studio bridge?*"
- `license_install` installs a key. Paste the key (it starts with `MCZ1-`) or the whole licence e-mail: "*Install this licence key for the Blender bridge: MCZ1-...*"

When the trial has ended, every other tool answers with a short message that says so and gives the link to buy.

The MCZ MCP Manager

Every server's installer also installs the **MCZ MCP Manager**, unless a copy of the same or a newer version is already there. Everything goes in one Start menu folder, **MCZ MCP**: *MCZ MCP Manager*, *MCP servers getting started* and a manual for each server you install. It shows, in one window:

- this computer's Machine ID (with a **Copy** button),
- each server: whether it is installed, its version, and its licence state or trial days left,
- a **Quick licence** button per server, which opens the buy page on mcz.software with the product and your Machine ID already filled in,
- a box to paste a key (or the licence e-mail) and **Install key**. A bundle key is installed once for every server it covers,
- a **Register with Claude** button per installed server, and **Register all with Claude** (see below).

Opening the manager never starts a trial.

Register with Claude

If a server has gone missing from Claude, for example after `claude_desktop_config.json` was edited by hand and broken, **Register with Claude** (version 1.0.3 and later of the manager) puts the server's entry back, the same entry its installer writes. It updates every Claude Desktop configuration file it finds (the normal one and the Microsoft Store copy), and creates that file if Claude Desktop is installed and has none. It also adds the server to Claude Code's settings file (`~\.claude.json`), but only when that file already exists: it is never created, so if Claude Code has not been run yet that file is skipped. Before changing any file it saves a backup next to it, named `<file>.bak-<date>-<time>`.

- A valid file keeps everything else exactly as it was, comments included.
- A broken `claude_desktop_config.json` is rebuilt with every other server and setting that could be recovered from it, plus the MCZ servers. If nothing could be recovered, it holds only the MCZ servers; the original is still in the backup.
- Claude Code's `~\.claude.json` holds a lot of other settings, so if it is broken the manager asks first: **Yes** rebuilds it from what could be recovered, **No** leaves it exactly as it is.

A window then lists each file with what happened (registered, already OK, repaired with the backup's path, skipped or failed and why). Restart Claude Desktop afterwards: quit it from the notification-area icon (see *Restarting Claude Desktop*) and start it again. For support, the same is available from a command prompt: `"%LOCALAPPDATA%\Programs\MCZ MCP Manager\McpManager.exe" --register all` (or a server's code such as `gitmcp`; add `--yes` to allow rebuilding a broken `~\.claude.json`).

From a command prompt

Each server can also show and install licences from a command prompt with `--license`:

```
<server program> --license                (shows the state; exit code 0 = tools work, 2 = not)
<server program> --license MCZ1-XXXXX-...  (installs a key, or the path of a file holding it; 0 =
installed, 1 = failed)
```

The program name for each server is in its manual. Keys and trial records are kept in `%LOCALAPPDATA%\MCZ\products` (and `%ProgramData%\MCZ\products` if that folder exists), shared by all the servers and the manager.

What the installers do to Claude's configuration

Claude Desktop reads the list of MCP servers from a file called `claude_desktop_config.json`. When you tick the *Add to Claude Desktop* option, each installer:

- adds (or updates) **its own entry** under `mcpServers`, and leaves every other server and setting as it was,
- updates every copy of the file it finds (the normal Claude Desktop and the Microsoft Store version keep it in different places, see below), and creates the file if there is none,
- first saves a backup next to it, named `claude_desktop_config.json.bak-<date>-<time>`, then writes the new file in one step so Claude never reads half a file.

Uninstalling a server removes its entry the same way. After installing or uninstalling, **quit Claude Desktop completely and start it again** (see below), because Claude reads the file only when it starts.

Editing claude_desktop_config.json by hand

You only need this if you want to change an entry yourself (for example the Unreal project a session opens), or to repair the file.

Where it is

- **Easiest:** in Claude Desktop open **Settings > Developer > Edit Config**. This opens the folder with the right file for your copy of Claude.
- Claude Desktop from claude.ai: `%APPDATA%\Claude\claude_desktop_config.json` (type `%APPDATA%\Claude` in the File Explorer address bar).
- Claude Desktop from the Microsoft Store: `%LOCALAPPDATA%\Packages\Claude_<id>\LocalCache\Roaming\Claude\claude_desktop_config.json` (`<id>` is a code that differs per PC; look in `%LOCALAPPDATA%\Packages` for the folder starting with `Claude_`).

Edit it with Notepad or any text editor. Close Claude Desktop first, or restart it afterwards.

The structure

The file is JSON: one object with an `mcpServers` object inside, which has one entry per server. Each entry has a `command` (the program to start) and, for some servers, `args` (the words passed to it). Other settings in the file stay where they are.

This is a complete example with all six servers, exactly as the installers write them. Replace `YourName` with your Windows user name, and check the versions and paths against your own file (your Android Studio, Unreal project and installed versions may differ):

```
{
  "mcpServers": {
    "git": {
      "command": "C:\\Users\\YourName\\AppData\\Roaming\\GitMcp\\server\\GitMcp.Server.exe",
      "args": ["--stdio"]
    },
    "visualstudio": {
      "command": "C:\\Users\\YourName\\AppData\\Roaming\\VsMcp\\VsMcpBridge.exe",
      "args": []
    },
    "blender-csharp": {
      "command": "C:\\Users\\YourName\\AppData\\Roaming\\BlenderMcp\\server\\BlenderMcp.Server.exe"
    },
    "pc-hardware": {
      "command":
"C:\\Users\\YourName\\AppData\\Roaming\\PCHardwareMcp\\server\\PCHardwareMcp.Server.exe"
    },
    "unreal-csharp": {
      "command": "C:\\Users\\YourName\\AppData\\Roaming\\UnrealMcp\\server\\UnrealMcp.Server.exe",
      "args": ["--port", "9878", "--project", "C:\\Projects\\MyGame\\MyGame.uproject"]
    },
    "android-studio": {
      "command": "C:\\Program Files\\Android\\Android Studio\\jbr\\bin\\java.exe",
      "args": ["-Xmx64m", "--enable-native-access=ALL-UNNAMED", "-cp",
               "C:\\Users\\YourName\\AppData\\Local\\AsMcp\\bridge\\AsMcp-0.2.0.jar",
               "asmcp.StdioBridge", "--studio", "C:\\Program Files\\Android\\Android Studio"]
    }
  }
}
```

The names on the left (`git`, `visualstudio`, `blender-csharp`, `pc-hardware`, `unreal-csharp`, `android-studio`) are the names Claude shows for the servers; the installers use exactly these.

JSON rules that trip people up

- Every name and every text value is in **double quotes** (`"`), never single quotes.
- Entries are separated by **commas**, and there is **no comma after the last entry** in an object or list.
- In Windows paths every backslash is **doubled**: `C:\\Users\\YourName\\...`. A single backslash breaks the file. Forward slashes (`C:/Users/YourName/...`) also work.
- Braces `{ }` and brackets `[ ]` must match.
- Keep a copy of the file before you edit it (the installers' `.bak-...` files are copies too).

Checking the file

Open PowerShell and run (for the normal Claude Desktop):

```
Get-Content "$env:APPDATA\Claude\claude_desktop_config.json" -Raw | ConvertFrom-Json | Out-Null
```

No output means the JSON is valid. An error message gives the line and position of the mistake. To list the servers in it:

```
(Get-Content "$env:APPDATA\Claude\claude_desktop_config.json" -Raw | ConvertFrom-
Json).mcpServers.PSObject.Properties.Name
```

Restarting Claude Desktop

Closing the Claude window does not quit it: it keeps running in the notification area (next to the clock). Right-click the Claude icon there and choose **Quit**, then start Claude again. In a new conversation, open the tools menu (the connectors / tools button in the message box) to check that the servers are listed and switched on.

Claude Code

Claude Code (the command-line and IDE version of Claude) keeps its own list of MCP servers. Add a server for your user account (all folders) with `claude mcp add --scope user`, using the same program and arguments as the Claude Desktop entry:

```
claude mcp add --scope user git -- "%APPDATA%\GitMcp\server\GitMcp.Server.exe" --stdio
claude mcp add --scope user visualstudio -- "%APPDATA%\VsMcp\VsMcpBridge.exe"
claude mcp add --scope user blender-csharp -- "%APPDATA%\BlenderMcp\server\BlenderMcp.Server.exe"
claude mcp add --scope user pc-hardware -- "%APPDATA%\PCHardwareMcp\server\PCHardwareMcp.Server.exe"
claude mcp add --scope user unreal-csharp -- "%APPDATA%\UnrealMcp\server\UnrealMcp.Server.exe" --project
"C:\Projects\MyGame\MyGame.uproject"
claude mcp add --scope user android-studio -- "C:\Program Files\Android\Android Studio\jbr\bin\java.exe"
-Xmx64m --enable-native-access=ALL-UNNAMED -cp "%LOCALAPPDATA%\AsMcp\bridge\AsMcp-0.2.0.jar"
asmcp.StdioBridge --studio "C:\Program Files\Android\Android Studio"
```

Run these in a Command Prompt (in PowerShell write `$env:APPDATA` instead of `%APPDATA%`). `claude mcp list` shows what is registered. The Visual Studio bridge's installer also adds its entry to Claude Code's settings by itself when Claude Code is installed. The Visual Studio and Blender bridges route a session by its working folder, so a Claude Code session started in a project folder goes to that project's window automatically; the Unreal and Android Studio bridges route by the project named in each call.

Troubleshooting

Claude says a server failed to start, or it is missing from the tools list. Check that the program in the `command` line exists (copy the path into File Explorer). Run it once from a command prompt with `--license`: if it prints the licence state, the program itself is fine. Make sure the configuration file is valid JSON (see *Checking the file*), then quit and restart Claude Desktop. Claude Desktop's own MCP logs are in `%APPDATA%\Claude\logs` (files named `mcp-server-<name>.log`).

"The configuration file is invalid" or no servers appear at all. One mistake in the JSON stops Claude reading the whole file. The quickest fix is **Register all with Claude** in the MCZ MCP Manager (see *Register with Claude*): it backs the file up, rebuilds it with every server it can recover and adds the MCZ servers back; then restart Claude Desktop. To fix it by hand instead, run the PowerShell check above and fix the line it reports, or copy back the most recent `.bak-...` file.

The tool answers that the free trial has ended. Buy a licence with the link in the message (or **Quick licence** in the MCZ MCP Manager), then install the key with `license_install`, the manager, or `--license`. A computer joined to a domain or Entra ID needs a commercial key; a personal key is refused there.

A bridge says the editor is not running. Open the editor (Visual Studio, Blender, Unreal Editor or Android Studio) with its extension, add-on or plugin enabled, or ask Claude to start it (`vs_session open`, `blender_session open`, `launch_editor` or `launch_studio`). Check the editor's own MCP status as described in that server's manual.

"Already in use" or "In use". Another Claude session is already working in that editor window or project. Use a different window, or ask that session to release it (the session tools in each bridge's manual).

A call timed out. Claude Desktop gives a tool call about a minute. Long builds and waits are split into steps: the server returns after about 50 seconds and Claude calls the matching `..._wait` tool to carry on.

More help: support@mcz.software.