MCZ · MCP SERVERS

Android Studio bridge manual

Android Studio bridge 0.2.0 · 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 Android Studio bridge 0.2.0 lets Claude work in Android Studio: Gradle sync and builds (following the log while they run), parsed build errors, run configurations, the debugger (Java/Kotlin and native LLDB), devices with adb and logcat, and open editors. Several Claude sessions can share it, each claiming its own project. New to the MCP servers? Read Getting started first.

How it works

- An Android Studio **plugin** (AsMcp) runs a small MCP server inside Android Studio, on this computer only, on port 8790 (a second Android Studio takes the next free port, up to 8799).
- Claude starts the **bridge** for each session. It runs on Android Studio's own Java, answers the tool list itself (so the tools are there even before Android Studio starts), and forwards each call to Android Studio. It can also start Android Studio (`launch_studio`).
- Calls carry a private token from `%APPDATA%\AsMcp\token.txt`, so other programs cannot drive Android Studio.
- The plugin is written in plain Java against Android Studio's own APIs, with no third-party code.

Requirements

- Windows 11, 64-bit.
- Android Studio 2026.1 (Quail) or later. The bridge uses Android Studio's built-in Java; nothing else to install.
- No administrator rights.

Install and uninstall

Close Android Studio, then run `AsMcpSetup-0.2.0.exe`. It installs for your Windows account only (setup files in `%LOCALAPPDATA%\AsMcp\setup`):

- the plugin into Android Studio's plugins folder (`%APPDATA%\Google\AndroidStudio<version>\plugins\AsMcp`),
- a token in `%APPDATA%\AsMcp\token.txt` (kept if it exists),
- the bridge into `%LOCALAPPDATA%\AsMcp\bridge\AsMcp-0.2.0.jar`,

- the MCZ MCP Manager, if needed.

Setup finds Android Studio in `Program Files\Android\Android Studio` or `%LOCALAPPDATA%\Programs\Android Studio` (or use `/STUDIO="<folder>"`), and waits while Android Studio is running. With **Add to Claude Desktop** ticked it registers the server as `android-studio`:

```
"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"]
}
```

Then start Android Studio and restart Claude Desktop. Android Studio's log then shows `AsMcp 0.2.0 listening on http://127.0.0.1:8790/mcp`. Setup's own log is `%LOCALAPPDATA%\AsMcp\setup\install.log`.

Silent install: `AsMcpSetup-0.2.0.exe /VERYSILENT /TASKS=claude [/STUDIO="<Android Studio folder>"]` (fails if Android Studio is running).

Uninstall from **Settings > Apps** (close Android Studio first): it removes the plugin and the `android-studio` entry. The token and the bridge folder are left (delete them by hand if you want). Licence keys stay in `%LOCALAPPDATA%\MCZ\products`.

For Claude Code:

```
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"
```

First use

Open your project in Android Studio (or ask Claude to start it), then ask:

- "*Claim the QuestApp project, run a Gradle sync and tell me the errors.*"
- "*Build assembleDebug and show me what failed.*"
- "*Which devices does adb see? Run the app configuration on the phone and show me the logcat for my package.*"
- "*Set a breakpoint at MainActivity.kt line 42, start debugging and show the locals when it stops.*"

Claude claims the project first (see below). The 30-day trial starts when you install the server on a computer.

Tools

36 tools.

Group	Tools
Android Studio	`studio_info`, `instances`, `launch_studio`

Group	Tools
Projects	`claim_project`, `release_project`, `open_project`, `close_project`
Gradle	`sync`, `build`, `build_wait`, `build_status`, `build_cancel`
Errors and build log	`get_build_log`, `get_errors`
Run and debug	`run_configurations`, `run`, `run_status`, `stop`, `debug_state`, `debug_control`, `debug_wait`, `get_locals`, `evaluate`
Breakpoints	`breakpoints`
Devices	`devices`, `logcat`, `adb`
Editors and actions	`documents`, `save_all`, `open_file`, `refresh_files`, `actions`
Logs	`read_log`, `server_log`
Licence	`license_status`, `license_install`

Notes:

- `studio_info` comes first: version, port, open projects and who has claimed them, running builds and debug sessions.
- `build` runs Gradle tasks (default `assembleDebug`) and waits about 50 seconds; Claude follows longer builds with `build_wait`.
- `get_errors` parses Kotlin, javac, CMake/clang and AAPT errors and Gradle's "What went wrong".
- `debug_control` pauses, resumes, steps over, into or out, or stops; `evaluate` runs Java/Kotlin or LLDB expressions.
- `breakpoints` lists, adds (with a condition), removes and clears line breakpoints, including native C/C++ ones.
- `logcat` filters by package, tag, level or text; `adb` runs any adb command.
- `actions` searches for and runs any Android Studio action by its id.

Several sessions at once

- **One server per Android Studio**, on ports 8790 to 8799. Every project tool takes a **project** argument (part of the project's name or path, or a port), and the call goes to the Android Studio that has that project open.
- **Claim before working.** `claim_project` returns a lease, which Claude passes on every call for that project. Another session gets *"In use: project 'X' is claimed by another session (owner, since ..., last call ...)"*.
- A lease ends with `release_project`, after 30 minutes without calls, or when Android Studio restarts. `release_project force=true` frees another session's lease; use it only when you know that session is finished.
- No lease is needed for `studio_info`, `instances`, `claim_project`, `release_project`, `open_project`, `server_log`, `launch_studio`, the device tools (`devices`, `logcat`, `adb`), `read_log`, `actions` and the licence tools.

Settings

Variable	Default	Meaning
`ASMCP_PORT`	8790	First port Android Studio's server tries.
`ASMCP_TOKEN`	`token.txt`	Use this token instead of the file.
`ASMCP_STUDIO`	from `--studio`	The Android Studio folder the bridge and scripts use.

Android Studio reads the token when it starts; the bridge reads it on every call, so the token is never written into Claude's configuration.

Licence

30 days free, then a key. `license_status` and `license_install` always work (also when Android Studio is closed). From a prompt:

```
"<Android Studio>\jbr\bin\java.exe" --enable-native-access=ALL-UNNAMED -cp
"%LOCALAPPDATA%\AsMcp\bridge\AsMcp-0.2.0.jar" asmcp.StdioBridge --license
```

shows the state (exit code 0 when the tools work, 2 when not); add a key (or the path of a file holding it) after `--license` to install it. See Getting started and the licence terms.

Troubleshooting

- **"Android Studio is not running with the AsMcp plugin"**: start Android Studio (or ask Claude to `launch_studio`). If it is running, check **Settings > Plugins** that AsMcp is installed and enabled.
- **Check the connection** (with Android Studio open): `"<Android Studio>\jbr\bin\java.exe" --enable-native-access=ALL-UNNAMED -cp "%LOCALAPPDATA%\AsMcp\bridge\AsMcp-0.2.0.jar" asmcp.StdioBridge --selftest` ends with `SELFTEST OK` or `SELFTEST FAILED`.
- **HTTP 403 / token mismatch**: the token changed after Android Studio started. Restart Android Studio.
- **"In use"**: another session has claimed the project; release it there, or wait for its lease to expire.
- **Long builds or waits**: Claude Desktop gives a tool call about a minute, so the bridge limits waits to 55 seconds and Claude calls `build_wait` or `debug_wait` again. (`adb` is not limited: keep adb commands short.)
- **Installing an update fails**: Android Studio locks the plugin and a running Claude locks the bridge jar. Close both, then run setup again.
- **The server is not listed in Claude**: see Troubleshooting in Getting started.

Logs

- Android Studio's `idea.log` (**Help > Show Log in Explorer**), category **AsMcp**; Claude can read it with `read_log` and AsMcp's own recent activity with `server_log`.
- Bridge messages go to standard error, in Claude's MCP log (`%APPDATA%\Claude\logs\mcp-server-android-studio.log`).
- Setup: `%LOCALAPPDATA%\AsMcp\setup\install.log`.