

Connect An AI Assistant To DjMirror Pulse

MCP

Use the Model Context Protocol (MCP) to let a compatible AI assistant read and control DjMirror Pulse. For example, you can ask it to show the current artist, change a visual style, or trigger a lighting scene. You stay in charge: review requests carefully before allowing...

DjMirror Pulse User Manual Series

Source: docs_user_manuals\MCP_USER_GUIDE.md

Revision 2026.10 • Release 1.0.743

Table of Contents

- Chapter 1: How The Connection Works
- Chapter 2: Choose How To Connect
- Chapter 3: Recommended Setup: Claude Desktop With Stdio
- Chapter 4: Alternative: Connect With Local HTTP
- Chapter 5: What The Assistant Can Do
- Chapter 6: Try It Safely
- Chapter 7: Authentication And Troubleshooting
- Chapter 8: Before A Live Show

This guide is for the supported **Windows and macOS desktop app**. The MCP feature requires an active **Professional or Enterprise** license. Standard and Trial editions do not include the MCP server.

Chapter 1: How The Connection Works

There are three parts:

1. **DjMirror Pulse desktop app** — the show must be open, with its HTTP server running.
2. **DjMirror Pulse MCP server** — the companion program `DjMirrorPulseMcp` translates MCP requests into calls to the app. It is shipped with the desktop app.
3. **MCP-compatible AI client** — for example, Claude Desktop. The AI client connects to the companion program.

The AI client does not connect directly to the lights or visuals. It asks the MCP server to perform an action, and the MCP server passes that request to the running DjMirror Pulse app.

Chapter 2: Choose How To Connect

Connection	Best for	What to enter in the AI client
STDIO (recommended for Claude Desktop)	AI client and DjMirror Pulse on the same computer	The path to <code>DjMirrorPulseMcp</code>
HTTP	A local AI client that supports Streamable HTTP	<code>http://127.0.0.1:8800/mcp</code>

Both methods connect locally by default; they do not make the server available over the internet.

 **Note**

These steps do not configure ChatGPT by name. The HTTP address above is a loopback address available only on the computer running DjMirror Pulse. A hosted AI service cannot reach your computer's `127.0.0.1`. Use ChatGPT only if the specific ChatGPT client and plan you use support a compatible local MCP connection; do not expose the server to the internet to work around this limitation.

Chapter 3: Recommended Setup: Claude Desktop With Stdio

STDIO is the simplest option when Claude Desktop and DjMirror Pulse are installed on the **same computer**. Claude starts the MCP companion program directly; you do not need to open a terminal or configure an HTTP connection in Claude.

Step 1 — Start DjMirror Pulse

1. Open DjMirror Pulse and activate a Professional or Enterprise license. To check it, open **Settings** → **Help** → **Active License** and look for a tick beside **MCP server**, or ask the app directly:

```
curl.exe http://127.0.0.1:8080/api/v1/capabilities
```

The answer contains `"mcpServer": true` under `license.features` when your edition includes MCP.

2. Make sure the app's HTTP server is enabled. It is enabled by default; if you changed the setting, start the app with `--server`.
3. Keep DjMirror Pulse open while using the AI connection.

Step 2 — Find the MCP program

Find the companion program shipped with DjMirror Pulse:

- **Windows:** `DjMirrorPulseMcp.exe`, in the DjMirror Pulse application folder.
- **macOS:** `DjMirrorPulseMcp`, inside the DjMirror Pulse app bundle, alongside the main executable.

Use its full path in the next step. If you cannot find it, repair or reinstall DjMirror Pulse rather than downloading a separate copy.

Step 3 — Add it to Claude Desktop

Open Claude Desktop's `claude_desktop_config.json` file and add the `djmirror` entry inside its existing `mcpServers` object. The usual locations are:

- **Windows:** `%APPDATA%\Claude\claude_desktop_config.json`
- **macOS:** `~/Library/Application Support/Claude/claude_desktop_config.json`

Replace the example executable path with the full path to the MCP program on your computer.

Windows example:

```
{
  "mcpServers": {
    "djmirror": {
      "command": "C:\\Program Files\\DjMirror Pulse\\DjMirrorPulseMcp.exe",
      "args": []
    }
  }
}
```

macOS example:

```
{
  "mcpServers": {
    "djmirror": {
      "command": "/Applications/DjMirror Pulse.app/Contents/MacOS/DjMirrorPulseMcp",
      "args": []
    }
  }
}
```

If the file already contains other MCP servers, keep them. Add only the `"djmirror"` entry, using a comma between entries as required by JSON. Do not paste a second top-level `mcpServers` object.

Step 4 — Restart and check the connection

Quit and reopen Claude Desktop so it reads the updated configuration. Leave DjMirror Pulse running. Ask Claude a read-only question first, such as:

“Use DjMirror Pulse to tell me which artist and visual style are currently selected. Do not change anything.”

If Claude reports that the server is unavailable, see [Authentication and troubleshooting](#).

Chapter 4: Alternative: Connect With Local HTTP

Use this method only if your AI client supports the **MCP Streamable HTTP** transport.

1. In DjMirror Pulse's `config.json`, set `"mcp": true`, then restart the app. This asks the app to start the MCP companion automatically. If the key already exists, change its value rather than adding a duplicate.
2. In the AI client's MCP settings, add the endpoint:

```
http://127.0.0.1:8800/mcp
```

3. Restart or reconnect the AI client, then try the read-only check from Step 4 above.

The MCP HTTP listener binds to the local computer by default. `127.0.0.1` and `localhost` mean “this computer”; they are not addresses another computer or an internet-hosted AI can use. Do not change the listener to a network-facing address unless you understand and have configured the required security controls.

Chapter 5: What The Assistant Can Do

The assistant works through **tools**. This version of DjMirror Pulse offers **199 tools**, grouped here by area. You never call them by name: describe what you want in plain language and the assistant picks the right tool.

Area	Tools	What the assistant can do	Example request
State and configuration	9	Read the live configuration and capabilities; change global or sectional settings; read and set the visualization	"Show me a summary of the current configuration."
Layers	37	Show, hide and reorder layers; read and change each layer's settings (image, avatar, video, artist card, vinyl, marquee, waveforms, lighting...)	"Hide the marquee text layer."
Artists	13	List and select artists; add, edit, delete and reorder them; read the artist layer	"Select the artist <i>Closer</i> ."
Styles	12	List, read, apply, create, edit and delete saved styles	"Apply the style <i>Chill</i> to the live show."
Marquee messages	9	List, add, edit, delete and reorder the scrolling messages	"Add a marquee message: <i>Last orders at 23:45</i> ."
Media and playlists	18	List images, videos, avatars and logos and their playlists; read and change which playlist is active	"List my video playlists."
Stock search and downloads	7	Search Pixabay and Pexels for images, videos and avatars and download the ones you pick	"Search for 5 neon crowd photos but do not download yet."
DMX	23	Read and change fixtures, scenes and channels; trigger scenes, blackout and macros; control silence behavior; export or import a DMX snapshot	"Check the DMX status without changing any fixture."
Schedule	12	List, add, edit, delete, reorder, import, export and run schedule entries	"List the schedule entries."
Timecode	20	Read timecode status and sources; manage cues; import or export them; jump or locate	"List the timecode cues."
Run an action now	1	<code>execute_action</code> fires any of the 14 schedule and controller actions immediately	"Switch to the next video in the playlist."
Screen recording	8	Take a snapshot, start or stop a recording, download either	"Take a snapshot of the output."
Spout	13	Status, start and stop, configuration, senders and multi-output layouts	"Is the Spout output running?"
NDI	7	Status, start and stop the NDI bridge, its configuration and saved configs	"Is the NDI output running?"
Ableton Link and beat clock	8	Read Link status and configuration, mark the downbeat, set or clear beat anchors	"What is the Link tempo?"
Automatic effects	2	Read and change the automatic effects and their rules	"Which auto effects are on?"

The numbers are the tools available in this build, grouped by name (a few tools could fit two areas); new versions may add more. The complete list, with every name and parameter, is in [docs tech/docs mcp/README MCP](#).

Note

Some areas need the matching edition: Link, Timecode, NDI and Spout are Professional or Enterprise features (multi-output layouts are Enterprise). See [License & Activation](#).

Check the connection without an AI client

The MCP companion answers on the local HTTP address when it is running. These two read-only requests prove it works and tell you how many tools it offers (PowerShell: use `curl.exe`):

```
curl.exe -s -X POST http://127.0.0.1:8800/mcp -H "Content-Type: application/json" `
  -d '{"jsonrpc":"2.0","id":1,"method":"initialize","params":{"protocolVersion":"2024-11-0
```

A working server answers with its name and version:

```
{"id":1,"jsonrpc":"2.0","result":{"capabilities":{"tools":{}},"protocolVersion":"2024-11-0
```

Then ask for the tool list with `"method":"tools/list"`; the answer contains one entry for each tool, with its description and parameters. If neither request answers, the companion is not running — see [Authentication and troubleshooting](#).

Chapter 6: Try It Safely

Start with read-only requests, then try a small change when you are ready:

Read-only checks

- "List the available visual styles. Do not change the current style."
- "Tell me which artist is selected."
- "Check the DMX status without changing any fixtures."

Changes — rehearse these before a show

- "Switch to the visual style named `<style name>`."
- "Show the layer named `<layer name>`."
- "Apply the DMX scene named `<scene name>`."

The assistant may ask you to clarify names or options. Check the requested action before approving it, especially for DMX blackout, strobe, fog, haze, or laser actions. Keep the keyboard or a hardware controller available as a manual override.

Chapter 7: Authentication And Troubleshooting

API key

If DjMirror Pulse has an API key configured, the MCP server must use the same key to call the app.

- When the app starts the MCP server automatically (`"mcp": true`), it passes the app's configured API key to the child process.
- If you start the MCP server yourself from Claude or another client, configure that same key for the MCP process in `config_mcp.json` under `auth.apiToken`, or provide it through the MCP process's environment.
- Do not paste API keys into a prompt or share them with an AI chat.

For a local single-computer setup without an API key, the default configuration is ready to use. For advanced configuration, see [docs_mcp/README MCP](#).

Message or symptom	Likely cause	What to do
MCP server is not listed or available in Claude	Invalid executable path, invalid JSON, or Claude has not restarted	Check the path and JSON commas/braces; fully quit and reopen Claude Desktop
<code>APP_NOT_RUNNING</code>	DjMirror Pulse is closed, or its HTTP server is disabled	Open the app and enable/start its HTTP server
<code>UNAUTHORIZED</code>	API keys do not match	Set the same key in DjMirror Pulse and the MCP process
MCP feature unavailable	Standard or Trial edition	Activate a Professional or Enterprise license
HTTP connection refused	MCP companion is not running, or the port differs from <code>8800</code>	Start the companion; check <code>config_mcp.json</code> and any port override; test with the <code>initialize</code> request in \$5
The assistant says a tool does not exist	The client cached an old tool list	Restart the AI client so it reads the tool list again
A request is refused for a Professional feature	The edition does not include it	Check Settings → Help → Active License (see License & Activation)
AI suggests an unexpected change	The request was ambiguous	Ask it to describe the exact action first; approve changes only after checking them

Chapter 8: Before A Live Show

1. Confirm the desktop app and MCP connection are both running.
2. Test a read-only request, then rehearse the specific changes you plan to use.
3. Confirm your API key is not exposed in prompts, logs you share, or an unsecured network service.
4. Keep a human operator and physical controls available throughout the show.

For the complete tool list and advanced settings, see [docs tech/docs mcp/README MCP](#).