PART TWO

PART TWO · A FIELD GUIDE FOR DIRECTORY OPERATORS

When Claude Stops Recommending and Starts Doing

Working with the Brilliant Directories MCP connection.

Courtesy of Gerald Griffith

The Vending Club · thevendingclub.net

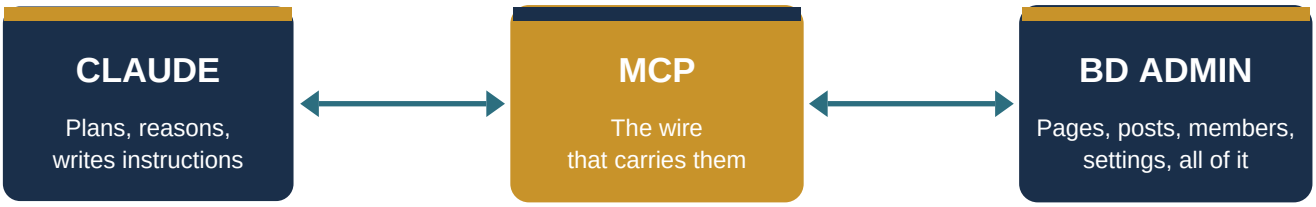
Companion to Part One — The Three-Document Foundation.

MCP IN PLAIN ENGLISH

The wire that lets Claude work directly with your site

In Part One, you built a strategic foundation — three documents that together make every later prompt sharper. Up to that point, Claude has only been *describing* what to do. The next step turns Claude into an active builder.

MCP stands for Model Context Protocol. The technical details aren't important. What matters is this: **MCP is the wire that lets Claude talk to your BD site directly.** When the wire is connected, Claude can both read your site's data and make changes to it. Without the wire, Claude can advise you brilliantly, but you have to do every click yourself.



Setup is its own topic — both Anthropic and Brilliant Directories publish documentation that walks through it. It's a one-time configuration, and once it's in place the connection persists across your working sessions. What this guide covers is what becomes possible after the wire is in place.

WHAT CHANGES

Before and after the connection

The shift isn't subtle. The same kind of plan that previously took weeks of admin work to execute can be scoped, instructed, and executed in a single working session. Your role doesn't disappear — it relocates from clicking to deciding.

WITHOUT MCP	WITH MCP
You are the executor	Claude is the executor
Each task = a manual click	Each task = a single instruction
Claude is a consultant	You are the editor
Scope limited by your time	Scope limited by what you verify
Plans live in a doc	Plans live and run in a doc

The reason this matters is leverage. The strategic foundation you built in Part One is the *what*. MCP is the *how*. Without both, the project stalls — you either have a plan with no execution capacity, or execution capacity with no clear plan. Together, they compound.

WHAT MCP UNLOCKS

Five categories of capability

Once connected, Claude has access to dozens of tools that map to BD's API. You don't need to memorize them. You describe what you want; Claude picks the right tool. Those tools cluster into five practical categories.

1	**INSPECT**	Read what's currently configured — members, posts, pages, post types, custom fields, memberships, settings. Useful for verification, audits, and "what's currently set?" checks before you change anything.
2	**CREATE**	Build new content — pages (including SEO landing pages with custom slugs and search-result types), blog posts, listings, leads, custom fields, post types, memberships. The most leverage-producing category.
3	**CONFIGURE**	Adjust structure — modify existing post types, update custom field options, change membership tier capabilities, update site settings, edit category trees.
4	**MANAGE**	Operate on records — update member profiles, modify lead lifecycle states, edit page content, change member tiers, update lead assignments.
5	**COORDINATE**	Site-wide operations — cache refreshes, redirects, sitemap actions, search-related changes, content propagation.

IN PRACTICE

What this looks like in real work

Three illustrative examples drawn from a real BD platform build. The specifics are illustrative; the patterns generalize to any directory.

BUILDING A LAYERED SEO CONTENT SYSTEM AT SCALE

An operator running a national directory wants state-specific SEO landing pages — one per state — that pull pre-filtered listings from the directory, carry custom hero copy with local references, and route keyword searches through state-aware URL parameters. Done by hand, that's a long manual build for each state: admin form, copy, configure, save, verify, repeat. Coordinated through MCP, the operator defines one canonical template, then instructs Claude to build the same pattern across the top seven states using a portable CSS naming convention. Result: seven state pages live, each fully configured, each verified — completed in a single working session.

BACKFILLING CONTENT DEPTH ON A TIGHT SCHEDULE

The audit identified the platform as light on educational content. The plan called for a series of pillar articles on operator pain points — locations, profitability, payment systems, business-sale due diligence. Done by hand, the workflow is write, format, copy into admin, configure, publish — a long loop per post. Coordinated through MCP, the directive becomes a single instruction: draft and publish a defined set of pillar articles as drafts in the blog. They arrive ready for review with correct titles, slugs, categories, and bodies. The operator's role shifts from producer to editor — which is where the judgment actually matters.

IN PRACTICE — CONTINUED

Standing up a new feature module

LAUNCHING AN ENTIRELY NEW MARKETPLACE MODULE

The implementation plan called for a marketplace feature — a structured submission board with custom fields, a multi-stage lifecycle for listings, notification rules, and seeded starter content. Conventionally, this rolls out over weeks: building the post type, configuring fields one by one, setting up workflows, writing seed posts, testing the submission flow. Coordinated through MCP, the structure is created, fields configured, lifecycle defined, and starter content seeded in a single working session. The operator's role narrows to approving structure and providing seeding judgment, not building each piece.

What these three have in common

Each is a task that, done by hand, would take days or weeks of focused admin work. None of them required writing code. None of them required deep technical understanding of BD's database or API. What each required was a clear strategic intent — which is exactly what the three foundation documents in Part One produce. The combination is what makes the leverage real.

Notice also what the operator's role becomes in each case. Not absent. Not passive. **Editorial.** Reviewing what was built, catching what's wrong, deciding what comes next. The connection doesn't replace your judgment — it frees your judgment from the mechanical work that previously consumed it.

HONEST LIMITATIONS

What MCP can't (or shouldn't) do

A guide that overpromises is worse than no guide. A few things the MCP connection doesn't solve, with brief reasons why — so you go in with realistic expectations.

- **Server-level caching.** Some changes don't appear on the live site until BD's underlying server cache clears, which Claude can't force. File a support ticket for cache-related propagation issues.
- **Hardcoded UI elements.** Header/footer buttons, certain widget caches, and a handful of global elements live outside what the API exposes. These need a BD support ticket.
- **Rich-text editor quirks.** BD's body-content editor strips certain HTML — custom inputs and JavaScript injected directly into page bodies can get wiped if you later edit through the visual editor. Workarounds exist (specific injection fields are reliable) but they're learned-the-hard-way knowledge.
- **Business decisions.** Claude won't decide your pricing tiers, verification standards, or moderation policy. You decide; Claude implements. This is a feature, not a limitation.
- **Verification.** The MCP layer occasionally reports success when a change didn't fully persist, especially around cache layers. Always spot-check on the live site before declaring a task done.

These limitations don't undercut the connection's value — they're the boundaries of where it adds value most. Knowing them up front keeps frustration low.

WORKING WELL

Patterns that produce good outcomes

A handful of habits, learned from real BD-build work, that consistently produced good outcomes. None of these are rules — they're worth adopting because they make the connection easier to live with.

Treat Claude as a high-trust executor

The most productive sessions started with broad directives — "execute Phase 1, items 1–10, and report back when done" — rather than micro-instructions. Claude works best with the strategic context already documented and a clear destination. Approve broad batches; intervene when something looks wrong; let Claude flag issues proactively rather than waiting to be asked.

Maintain an Outstanding Issues Tracker

A single document in your project listing BD-support-blocked items, pending operator-action items, and recently resolved items. Claude can read and update it every session. It prevents items from getting lost when conversations get long, and it creates a clean handoff point at the end of each working session.

Keep conversations focused

Long conversations get less effective over time as the context window fills. A good rhythm: one chat per strategic conversation, one chat per implementation phase, one chat per substantial day of execution. When a session ends, ask Claude to summarize what was done so the next session can pick up cleanly.

Verify before declaring done

After Claude reports completing a task, spot-check it. Open the live site in an incognito window, view source on the page that was supposedly fixed, confirm the change is actually there. Trust but verify, especially in the first week.

When stuck, ask "what would make this easier?"

If a task is dragging — too many sub-decisions, too much friction, too many "actually, maybe we should..." moments — that's usually a signal that the strategic context is incomplete. Stop. Ask Claude what additional context document would make the task straightforward. Sometimes the answer is a pricing-tier matrix, an editorial style guide, or a roles-and-permissions document. Write that document, then come back. It almost always unblocks.

A MENTAL MODEL

MCP, in one sentence

Without MCP, your AI assistant is brilliant but bottlenecked at your hands.

With MCP, the bottleneck moves from your hands to your judgment — which is where it should have been all along. Your job becomes deciding what's right; Claude's job becomes making it true.

That's it. That's the whole shift.

The strategic foundation tells you what to do. The connection makes doing it possible. Both halves matter.

Courtesy of Gerald Griffith · The Vending Club · thevendingclub.net

Pairs with Part One — The Three-Document Foundation.