Workflow Learning Guide — Slash Commands Explained

> ***Last Updated:*** *February 10, 2026*
> ***Audience:*** *Internal — founders, team members, and collaborators*
> ***Purpose:*** *Understand what each workflow does, why it matters, and what skills it builds*

What Are Slash Commands?

Slash commands are **automation workflows** that turn multi-step processes into a single instruction. Instead of manually creating folders, moving files, and formatting documents, you type something like `/new-client` and the system handles the rest.

Think of them as your **standard operating procedures (SOPs) turned into one-click automation**. They ensure consistency, save time, and eliminate human error — the same principles behind why franchises succeed (every McDonald's makes a Big Mac the same way).

Why This Matters for Your Growth

Learning these workflows teaches you:

- **Systems thinking** — how to break complex processes into repeatable steps
- **Automation mindset** — identifying tasks that should be standardized vs. improvised
- **Professional operations** — running a business with the rigor of a larger firm
- **Technical literacy** — understanding file structures, version control, and dev tooling

---

The Workflows

📦 `/consolidate` — Daily Workspace Consolidation

What it does: End-of-day cleanup that audits your files, creates a daily summary of everything accomplished, archives old trackers, updates the dashboard, generates PDFs, and commits changes.

When to use it: At the end of any work session where you created, modified, or deleted files.

Skills it builds:

- **Documentation discipline** — capturing what was done while it's fresh
- **File lifecycle management** — knowing when to archive vs. delete vs. keep
- **Clean workspace habits** — entropy naturally increases; this fights it
- **Audit trail thinking** — creating records that prove what happened and when

Key concept — "Archive, don't delete": Old files get archived with redirects to newer versions rather than deleted outright. This preserves history while keeping the workspace lean. In a legal or compliance context, this practice is called *record retention*.

🧹 `/cleanup` — Workspace & Cache Cleanup

What it does: Removes empty scaffold folders, temporary files, and clears browser/tool caches that accumulate over time.

When to use it: Weekly, or whenever the workspace feels cluttered. Also useful before taking a backup or snapshot.

Skills it builds:

- **Storage hygiene** — understanding that digital workspaces accumulate cruft
- **Cache awareness** — knowing what browser caches are and why they grow
- **Proactive maintenance** — fixing small messes before they become big ones

Key concept — "Digital housekeeping": Empty folders and temp files are like leaving empty boxes in your office. They don't take up much space individually, but they clutter your view and make it harder to find what matters.

📁 `/new-client` — Client Folder Scaffold

What it does: Creates the complete folder structure for a new client — Sessions, Website & Strategy, Media, Action Items, and Meeting Notes — each with their own Markdown/PDF subfolders.

When to use it: Every time you onboard a new client.

Skills it builds:

- **Information architecture** — designing folder structures that scale
- **Onboarding process design** — standardizing how new relationships begin
- **Template thinking** — creating reusable structures instead of reinventing each time
- **Client management** — organizing all client materials in one predictable location

Key concept — "Structure before content": Setting up the right containers before filling them means you never have to reorganize later. This is the same principle behind database schema design — define the shape of your data before you start collecting it.

📋 `/new-session` — Session Scaffold

What it does: Creates a session folder with pre-filled invoice, service agreement, and receipt documents from templates. You provide the client name, session type, date, rate, and deliverables — it fills in everything else.

When to use it: Before each paid client session (photo shoots, consultations, website reviews, etc.).

Skills it builds:

- **Professional invoicing** — generating consistent billing documents
- **Contract basics** — understanding service agreements and scope definitions
- **Variable templating** — seeing how placeholders (`{{CLIENT_NAME}}`) get replaced with real data
- **Business operations** — having paperwork ready *before* the session, not scrambling after

Key concept — "Template-driven operations": Every Fortune 500 company uses templates for repetitive documents. The difference between amateur and professional isn't creativity — it's consistency. Templates free your creativity for the work that actually matters.

📄 `/generate-pdfs` — Markdown → PDF Converter

What it does: Batch-converts `.md` files to styled `.pdf` files, placing them in sibling `PDF/` folders. Can target one folder or sweep the entire workspace.

When to use it: After creating or updating markdown documents that need to be shared or printed.

Skills it builds:

- **Document delivery** — understanding that different audiences need different formats
- **Build tooling** — using CLI tools (`md-to-pdf`) to transform files

- **File organization patterns** — the Markdown/PDF sibling folder convention
- **Batch processing** — thinking in terms of "do this to all files" instead of one at a time

Key concept — "Source and artifact": The markdown file is your *source of truth* (editable, version-controlled). The PDF is an *artifact* (shareable, printable, but never edited directly). This mirrors how software development works: you edit code, then *build* the product.

🚀 `/build-site` — Client Website Generator

What it does: Generates a complete, deployment-ready website from a client's site brief, brand kit, and industry playbook. Supports three build lanes: Vite (fast static sites), Next.js (React apps), and hybrid.

When to use it: When building a website for a client from scratch.

Skills it builds:

- **Web development fundamentals** — HTML, CSS, JavaScript structure
- **Build systems** — understanding Vite, npm, and development servers
- **Brand-to-code translation** — turning brand guidelines into CSS variables and design tokens
- **Industry marketing** — using playbooks that encode best practices for different business types
- **Full-stack thinking** — seeing how design decisions flow from brief → brand kit → code → live site

Key concept — "Playbook-driven development": Instead of starting with a blank page, you start with proven patterns for your client's industry. A restaurant site needs different things than a nonprofit site. Playbooks encode that knowledge so every site starts strong.

🗂️ `/organize-files` — File Organizer

What it does: Organizes files into folder structures — creates directories, moves files by type, renames to match naming conventions, and verifies the result.

When to use it: When files are dumped into a flat folder and need to be sorted, or when importing files from an external source.

Skills it builds:

- **File management** — moving, renaming, and organizing at scale
- **Naming conventions** — understanding `snake_case`, date prefixes, and type suffixes
- **Directory design** — creating logical hierarchies for different content types
- **PowerShell/CLI proficiency** — getting comfortable with command-line file operations

Key concept — "Convention over configuration": When everyone follows the same naming rules, you never have to ask "where did I put that?" or "what is this file?" Good naming is self-documenting.

🔍 `/search` — Workspace Search

What it does: Searches across all `.md` and `.txt` files in the workspace for any keyword or phrase. Returns file names, line numbers, and matching context.

When to use it: When you know something exists but can't remember where, or when you want to find all references to a topic across the workspace.

Skills it builds:

- **Information retrieval** — finding needles in haystacks efficiently
- **Cross-referencing** — discovering connections between documents

- **Grep/search fundamentals** — the same tool (`ripgrep` , `Select-String`) used by professional developers daily

Key concept — "Searchability is a feature": If you can't find it, it doesn't exist. This is why consistent naming, clear titles, and plain text formats (Markdown) matter — they're all searchable. Binary formats like `.docx` or `.psd` are opaque to search.

💾 `/snapshot` — Backup Creator

What it does: Creates timestamped backups of Dashboard HTML files and key workspace files before major edits. Like a save point in a video game.

When to use it: Before making big changes — redesigning the dashboard, restructuring folders, or updating templates.

Skills it builds:

- **Backup discipline** — protecting your work before risky changes
- **Version management** — understanding that you need rollback capability
- **Risk mitigation** — thinking about "what if this goes wrong?" before it does
- **Recovery planning** — knowing how to restore from a backup

Key concept — "Measure twice, cut once": Snapshots are cheap insurance. A 30-second backup can save hours of re-work. This is the manual equivalent of Git branching — creating a safe copy before experimenting.

Quick Reference Table

Command	Purpose	Frequency	Skill Level
`/consolidate`	End-of-day cleanup & summary	Daily	Beginner
`/cleanup`	Remove cache & empty folders	Weekly	Beginner
`/new-client`	Set up client folders	Per client	Beginner
`/new-session`	Create session documents	Per session	Beginner
`/generate-pdfs`	Convert MD → PDF	As needed	Beginner
`/build-site`	Generate client website	Per project	Intermediate
`/organize-files`	Sort & rename files	As needed	Beginner
`/search`	Find anything in workspace	As needed	Beginner
`/snapshot`	Pre-edit backup	Before big changes	Beginner

The Bigger Picture

These workflows aren't just convenience tools — they represent a **professional operating system** for your business. Each one encodes a best practice:

1. **Consistency** → Every client gets the same folder structure, every invoice looks the same
2. **Traceability** → Daily summaries, backups, and PDFs create an audit trail

3. **Scalability** → Adding a 10th client is as easy as the 1st — just run `/new-client`
4. **Professionalism** → Deliverables arrive in PDF format with proper naming
5. **Resilience** → Backups mean one mistake doesn't undo weeks of work

Master these patterns and you'll think like an operations manager, not just a creative. That's the difference between a freelancer and a *business*.