

COMPLETE REFERENCE



ACOS Reference

The definitive guide to every command, agent, skill, and workflow in the Agentic Creator OS.

FRANKX | v7.0 | 2026

01

SYSTEM

System Architecture

ACOS v7.0 is built on a layered architecture: **CLAUDE.md** defines the system, **skill-rules.json** manages auto-activation, **commands/** contain slash command definitions, and **hooks/** automate lifecycle events.

46

CORE
COMMANDS

37

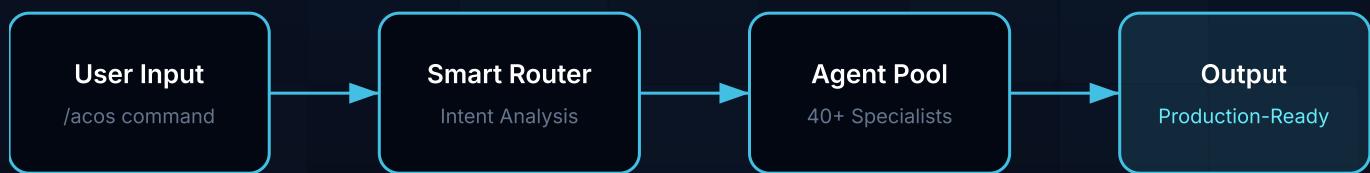
ACOS
COMMANDS

22

AUTO RULES

20+

HOOKS



Every command routes through intelligent classification to the optimal agent

02

COMMANDS

Command Reference: Core

COMMAND	CATEGORY	DESCRIPTION
/acos	Router	Universal auto-router — classifies intent and routes to optimal agent
/frankx-ai-build	Dev	Full build session with quality gates and testing
/frankx-ai-deploy	DevOps	Two-repo production deployment to frankx.ai
/frankx-ai-blog	Content	Create SEO-optimized blog post with schema markup
/frankx-ai-seo	SEO	Site-wide SEO analysis and optimization
/frankx-ai-components	UI	React component development with Tailwind
/frankx-ai-content-pipeline	Content	End-to-end content: research → write → SEO → publish
/frankx-ai-products	Product	Digital product creation and launch workflow
/frankx-ai-prompts	AI	Prompt library management and generation
/frankx-ai-analytics	Data	Site analytics and performance monitoring
/frankx-ai-daily	Ops	Daily operations and task management

03

COMMANDS

Command Reference: Agentic

COMMAND	CATEGORY	DESCRIPTION
/ultraworld	Creative	Arcanean swarm world-building — 7 creative agents
/acos-swarm	Multi-Agent	Swarm coordination for parallel agent execution
/acos-flow	Workflow	Workflow orchestration and pipeline management
/acos-agents	Agents	Agent lifecycle: spawn, monitor, coordinate
/acos-memory	Memory	Persistent memory and cross-session learning
/acos-checkpoint	State	Session state checkpointing and recovery
/agentic-jujutsu	Learning	Trajectory extraction and pattern learning

Advanced Orchestration

COMMAND	CATEGORY	DESCRIPTION
/claude-flow-swarm	Internal	Low-level swarm spawning and control
/claude-flow-memory	Internal	Neural memory persistence layer
/claude-flow-sparc	Internal	SPARC methodology execution
/claude-flow-agents	Internal	Agent type registry and capabilities

"83 internal claude-flow commands were absorbed into ACOS v7.0 — never surfaced to users, always working behind the scenes."

Agent Reference



Technical Architect

Skills: architecture-patterns, ai-agents-architect, api-design, oracle-adk
Triggers: architecture, system design, backend, API



Music Producer

Skills: suno-ai-mastery, suno-prompt-architect
Triggers: suno, music, song, audio



Content Engine

Skills: frankx-brand, seo-content-writer, schema-markup
Triggers: article, blog, content, post



SEO Intelligence

Skills: seo-fundamentals, keyword-strategist, schema-markup
Triggers: seo, keywords, rankings



Frontend Designer

Skills: vercel-react, web-design, shadcn-ui, tailwind, accessibility
Triggers: component, design, ui, ux



DevOps Engineer

Skills: vercel-deployment, docker, github-actions
Triggers: deploy, build, ci, pipeline

05

SKILLS

Auto-Activation Rules

The `skill-rules.json` system uses 22 rules to automatically activate skills based on file patterns, keywords, and task context. 5 core skills are always active.

Always Active (Core 5)

`frankx-brand` Brand voice & guidelines

`clean-code` Code quality standards

`next-best-practices` Next.js patterns

`typescript-docs` TypeScript standards

`vercel-react` React optimization

Sample Activation Rules

TRIGGER	PATTERN	SKILLS ACTIVATED
File: <code>*.tsx</code>	Component files	vercel-react, tailwind-css, shadcn-ui
File: <code>*.mdx</code>	Blog content	seo-content-writer, schema-markup
Keyword: "deploy"	Deployment tasks	vercel-deployment, github-actions
Keyword: "suno"	Music creation	suno-ai-mastery, suno-prompt-architect
Keyword: "agent"	Agent design	ai-agents-architect, autonomous-agents
Context: Arcanea	Creative gates	Gate-specific skills (10 gates mapped)

06

AUTOMATION

Hooks & Lifecycle Events

ACOS hooks into 7 lifecycle events with 20+ automation hooks, executing shell scripts and Node.js hooks automatically during session and task lifecycles.

SessionStart

Loads context, initializes learning trajectories, activates skill profiles, sets output style.

UserPromptSubmit

Classifies intent, suggests skills, activates brand guidelines, logs context.

PreToolUse

Validates tool permissions, applies security policies, manages file access.

PostToolUse

Logs tool usage, updates trajectory data, triggers quality checks.



Hook Execution Log

[SessionStart] Loading context...

- resume hook: ✓ ACOS Session Context loaded
- learning hook: ✓ 23 trajectories, 22 patterns
- output-style hook: ✓ explanatory mode

[UserPromptSubmit] Classifying intent...

- skill-suggest hook: ✓ 3 skills recommended
- brand hook: ✓ frankx-brand activated
- ✓ All hooks executed in 0.8s

Memory & Learning

ACOS maintains persistent memory across sessions and uses **Agentic Jujutsu** to extract successful patterns from past interactions, continuously improving performance.

Memory Layers

MEMORY.md

Auto-loaded system prompt.
Architecture decisions, patterns,
workarounds. Kept under 200 lines.

Topic Files

Detailed notes organized by topic:
debugging.md, patterns.md, blog-
design-patterns.md.

MCP Memory

Knowledge graph via MCP server
— entities, relations, observations
across sessions.

Agentic Jujutsu

Trajectory Tracking

Every session records tool
sequences, outcomes, and success
rates.

Pattern Extraction

Successful tool sequences are
extracted and stored for reuse.

Self-Improvement

Patterns inform future routing
decisions, improving over time.

08

CONFIG

Configuration Reference

File Structure

● ● ● ACOS Directory Structure

```
.claude/
├── commands/ # 46+ slash command definitions
├── agents/ # Agent profile definitions
├── skills/ # Local skill rules & profiles
├── hooks/ # 4 shell + 4 JS hooks
├── context/ # Decision framework, identity
├── planning/ # Task plans and strategies
└── settings.json # Permissions and config

CLAUDE.md # System definition – the brain
skill-rules.json # 22 auto-activation rules
```

Key Configuration Files

FILE	PURPOSE	FORMAT
CLAUDE.md	Master system prompt — brand, agents, routing	Markdown
skill-rules.json	Auto-activation rules and skill profiles	JSON
settings.local.json	Tool permissions and allowed commands	JSON
.frankx/brand.md	Brand voice and design guidelines	Markdown
MEMORY.md	Persistent auto-memory (< 200 lines)	Markdown

REFERENCE V7.0

Agentic Creator OS

630+ skills. 40+ agents. 130+ commands.
One operating system for everything you create.

frankx.ai | github.com/frankxai/agentic-creator-os

Build what matters.