PRO SYSTEM

Building Custom Agents

Design, build, and deploy specialized AI agents that integrate with the ACOS ecosystem.

by Frank Guzman · frankx.ai · v7.0

Anatomy of an ACOS Agent

An agent is a **markdown file** in `.claude/agents/` that defines a specialized persona for Claude Code's Task tool.

Agent Definition File

```
# .claude/agents/my-agent.md
name: Content Optimizer
role: Content quality and SEO specialist
specialty: Analyzing and improving content for search and readability
## Core Capabilities
- SEO analysis (keyword density, meta tags, schema)
- Readability scoring (Flesch-Kincaid, sentence length)
- Voice consistency checking against brand guidelines
- Internal linking opportunities
## Tools Access
- All read/search tools (Glob, Grep, Read)
- WebSearch for competitive analysis
- WebFetch for SERP analysis
## Output Format
Return a structured report with:
1. Current score (1-100)
2. Top 5 improvements
3. Specific line-by-line suggestions
```

Required Fields

- **name** — Display name
- **role** — One-line role description
- **specialty** — Domain expertise

Recommended Fields

- **capabilities** — What it can do
- **tools** — Which tools it uses
- **output format** — Expected deliverable

Agent Design Patterns

Pattern 1: Domain Expert

Single-domain agent with deep knowledge. Best for specific, repeatable tasks.

```
Examples:
- SEO Auditor (checks pages against 50+ criteria)
- Music Producer (Suno prompt engineering + genre mastery)
- Security Reviewer (OWASP top 10 + code analysis)
```

Pattern 2: Pipeline Agent

Multi-step agent that orchestrates a workflow. Takes input, processes through stages, produces artifact.

```
Examples:
- Factory (research → outline → draft → edit → publish)
- Product Launch (spec → build → test → deploy → market)
```

Pattern 3: Reviewer Agent

Analyzes existing work and provides feedback. Read-only, returns structured reports.

```
Examples:
- Code Reviewer (bugs, security, quality issues)
- Content Reviewer (accuracy, voice, SEO compliance)
- Accessibility Auditor (WCAG 2.2 compliance)
```

Pattern 4: Coordinator Agent

Meta-agent that manages other agents. Used for complex tasks requiring multiple specialties.

```
Examples:
```

```
- Starlight Orchestrator (multi-agent coordination)
- Council (assembles domain experts for decisions)
- Master Story Architect (coordinates writing team)
```

Building Your First Agent

Let's build a practical agent step by step: a **Blog Post Optimizer**.

Step 1: Define the Agent

```
# Create the file
$ touch .claude/agents/blog-optimizer.md
```

Step 2: Write the Definition

```
# Blog Post Optimizer
You are the Blog Post Optimizer, a specialized agent that
analyzes MDX blog posts and provides actionable improvements.
## Role
Content quality specialist focused on SEO, readability,
and engagement optimization for technical blog posts.
## Process
1. Read the target MDX file
2. Analyze: word count, heading structure, keyword density
3. Check: meta title/description, schema markup, internal links
4. Score: readability (target: grade 8-10), engagement hooks
5. Report: structured findings with specific line numbers
## Output
Return a markdown report with:
- Overall score (1-100)
- Top 5 high-impact improvements
- Keyword opportunities
- Missing SEO elements
```

Step 3: Register with Skill Rules (Optional)

```
# In .claude/skills/skill-rules.json, add to "agents":
"blog-optimizer": {
  "type": "workflow",
  "activation": "suggest",
  "promptTriggers": {
    "keywords": ["optimize blog", "improve post", "blog seo"],
    "intentPatterns": ["optimize.*blog", "improve.*post"]
  }
}
```

Step 4: Use It

```
# The agent activates when its triggers match, or call directly:
```

```
"Optimize my latest blog post at content/blog/my-post.mdx"
```

Advanced Agent Techniques

Multi-Agent Composition

Create coordinator agents that delegate to specialists:

```
# Content Quality Council
Assemble these agents for a comprehensive review:
1. Blog Optimizer → SEO & readability analysis
2. Brand Voice Checker → Tone consistency
3. Fact Checker → Claim verification
4. Link Auditor → Internal link opportunities
Synthesize all findings into a single action plan
ranked by impact and effort.
```

Context-Aware Agents

Agents that adapt behavior based on project context:

```
## Context Detection
Before starting, check:
- If CLAUDE.md exists, read project conventions
- If package.json exists, detect tech stack
- If .frankx/ exists, apply brand guidelines
- Adapt output format to match project style
```

Self-Improving Agents

Agents that learn from their own trajectory data:

```
## Learning Loop
1. After each task, record: input, approach, output, feedback
2. Use /agentic-jujutsu to extract patterns
3. Update agent definition with successful strategies
4. Add anti-patterns to avoid list
```

Best Practice: Start simple. Build a single-purpose agent first. Get it working well. Then add complexity, composition, or learning capabilities incrementally.

Integrating Agents with ACOS

Triggering from Commands

Reference your agent in a command file to auto-invoke it:

```
# .claude/commands/optimize-blog.md
Use the Blog Optimizer agent to analyze the specified post.
Follow the process defined in .claude/agents/blog-optimizer.md.
After optimization, run /publish to deploy improvements.
```

MCP Integration

Agents can use MCP servers for external capabilities:

```
## Tools
- Use Playwright MCP for visual regression testing
- Use Nano Banana MCP for generating comparison images
- Use Memory MCP to persist findings across sessions
```

Skill Pairing

Pair agents with specific skills for enhanced capabilities:

```
## Required Skills
This agent requires these skills to be active:
- seo-fundamentals (SEO analysis rules)
- schema-markup (structured data validation)
- frankx-brand (brand voice checking)
```

Testing Your Agent

Validation checklist for custom agents:

- Does it produce consistent output format?
- Does it handle edge cases (empty files, missing data)?
- Does it respect read-only vs. write permissions?
- Does it integrate cleanly with existing workflows?
- Is the definition clear enough for Claude to follow reliably?

Need help building agents?

hello@frankx.ai

frankx.ai · github.com/frankxai · @frankxeth