

# Claude Multi-Agent PM Framework User Guide v4.1.0

## Claude Multi-Agent PM Framework

### User Guide v4.1.0

Complete Developer Documentation

CMPM Framework Team

July 08, 2025

---

## Table of Contents

### Part I: Getting Started

1. [Getting Started & Installation](#)
2. [Architecture & Core Concepts](#)

### Part II: Using CMPM

3. [Claude Slash Commands & Orchestration](#)
4. [Directory Organization & Best Practices](#)

### Part III: Development

5. [Custom Agent Development](#)
6. [Advanced Features & Integration](#)

## Part IV: Support

### 7. [Troubleshooting & FAQ](#)

## Appendices

### A. [Glossary](#) B. [Index](#)

## Preface

The Claude Multi-Agent PM Framework (CMPM) represents a revolutionary approach to software development project management through intelligent agent orchestration. This comprehensive guide provides everything needed to master the framework, from basic installation to advanced customization.

## Who This Guide Is For

This guide is designed for competent developers who want to leverage multi-agent coordination for enhanced productivity. Whether you're a solo developer or part of a larger team, CMPM provides the tools and patterns needed for efficient project management.

## How to Use This Guide

- **Quick Start:** Follow Chapter 1 for immediate setup (30 minutes)
- **Comprehensive Learning:** Read all chapters for complete mastery (4-6 hours)
- **Reference:** Use the glossary and index for quick lookups
- **Troubleshooting:** Refer to Chapter 7 for problem resolution

## Part I: Getting Started

### Chapter 1: Getting Started & Installation

### Chapter 2: Architecture & Core Concepts

## Part II: Using CMPM

# **Chapter 3: Claude Slash Commands & Orchestration**

# **Chapter 4: Directory Organization & Best Practices**

## **Part III: Development**

# **Chapter 5: Custom Agent Development**

# **Chapter 6: Advanced Features & Integration**

## **Part IV: Support**

# **Chapter 7: Troubleshooting & FAQ**

## **Appendices**

# **Appendix A: Glossary**

## **Framework Terms**

**Agent:** An autonomous component in the CMPM framework responsible for specific tasks such as development, testing, or deployment.

**Orchestration:** The coordination and management of multiple agents to accomplish complex project goals.

**Task Tool:** The primary mechanism for delegating work to specialized agents via subprocess creation.

**Memory Integration:** The use of mem0AI for persistent learning and context retention across agent interactions.

**Slash Commands:** Natural language commands beginning with '/' used to interact with the CMPM framework.

## Technical Terms

**ai-trackdown-tools:** CLI toolset for hierarchical project management and ticket tracking.

**Claude Code:** Individual development agent working within supervised subprocesses.

**CMPM:** Claude Multi-Agent PM Framework - the complete project management orchestration system.

**Git Worktree:** Isolated working directories that share repository history, enabling parallel agent work.

**mem0AI:** Memory service providing intelligent context retention and learning capabilities.

**Multi-Agent Architecture:** System design enabling multiple specialized agents to work collaboratively.

**Subprocess Delegation:** Framework pattern for creating isolated agent environments via Task tool.

## Agent Types

**Architect Agent:** Designs system architecture, APIs, and project scaffolding.

**Data Agent:** Manages data processing, storage solutions, and analytics integration.

**Documentation Agent:** Creates and maintains technical documentation and user guides.

**Engineer Agent:** Implements source code, business logic, and feature development.

**Integration Agent:** Handles system integration, API coordination, and service mesh management.

**Operations Agent:** Manages deployment, infrastructure, and monitoring systems.

**Performance Agent:** Optimizes system performance and analyzes bottlenecks.

**QA Agent:** Ensures quality through testing, validation, and quality assurance processes.

**Research Agent:** Investigates technologies, gathers requirements, and provides analysis.

**Security Agent:** Analyzes security vulnerabilities and implements security measures.

# Appendix B: Index

## A

- Agent coordination, 15, 23, 45
- Agent types, 12-18, 67-89
- ai-trackdown-tools, 25, 33, 112
- API development, 78, 95, 134
- Architecture patterns, 34-56

## B

- Best practices, 67-89, 145
- Bug fixing, 156-162

## C

- Claude Code integration, 23, 45
- CMPM installation, 8-14
- Commands, slash, 57-66
- Configuration management, 89-94
- Custom agents, 95-134

## D

- Data management, 78, 112
- Development workflow, 45-56
- Directory structure, 67-89
- Documentation generation, 124, 178

## E

- Error handling, 156-162
- Event-driven architecture, 112, 134

## F

- Framework architecture, 15-34

## G

- Git worktrees, 23, 89

## I

- Installation guide, 8-14
- Integration patterns, 112-134

## M

- mem0AI integration, 135-145
- Memory management, 135-145
- Multi-agent coordination, 15-34

## O

- Orchestration language, 57-66
- Operations management, 78, 145

## P

- Performance optimization, 145-155
- Project structure, 67-89

## Q

- Quality assurance, 78, 95

## S

- Security implementation, 145, 162
- Slash commands, 57-66
- System requirements, 8-12

## T

- Task delegation, 23, 45
- Testing frameworks, 78, 162
- Troubleshooting, 156-178

## U

- User guide navigation, 1-7

# W

- Workflow optimization, 45-56