# Role Context Manager

Plugin Version 1.3.0

Role-based document context manager for Claude Code

## Table of Contents



# Initial Setup Phase

**When to use:** First-time user or new project initialization

## Commands

`/init-org-template [--global|--project]`

**Purpose:** Initialize organizational framework from a template

**What it does:**

- Invokes Template Setup Assistant agent
- Analyzes your project structure
- Presents available templates (software-org, startup-org)
- Applies chosen template to appropriate scope
- Records template tracking for auto-updates

**Flags:** `--global` (apply to ~/.claude/), `--project` (apply to ./.claude/)

`/setup-plugin-hooks [--global|--project]`

**Purpose:** Configure SessionStart hook for automatic validation

**What it configures:**

- Adds SessionStart hook to settings.json
- Sets up automatic validation and update checks
- Creates setup marker file

`/set-role [role-name] [--global|--project]`

**Purpose:** Set your current role to determine which documents load

**Parameters:** role-name (required, e.g., software-engineer, product-manager)

**What it does:**

- Validates role exists at current organizational level
- Updates preferences.json with your role
- Initializes role-specific document references
- Displays documents that will load on next session

`/set-org-level [level] [--global|--project]`

**Purpose:** Explicitly set organizational level

**Levels:**

- **company** - Root level (CTO, CPO, CISO, VP Engineering)
- **system** - Coordination level (Engineering Manager, Platform Engineer)
- **product** - Product management level (Product Manager, QA Manager, UX Designer)
- **project** - Implementation level (Software Engineer, DevOps Engineer, QA Engineer)

**When to use:** Directory structure doesn't match standard patterns, want to override automatic detection

## Agent: Template Setup Assistant

**Template Setup Assistant**

**Invoked by:** `/init-org-template`

**Purpose:** Guide users through template selection and setup

**Capabilities:**

- Analyzes current directory structure
- Loads available templates from plugin's templates/ directory
- Presents template options with recommendations
- Asks clarifying questions about organization size and stage
- Applies selected template with user confirmation
- Guides user to next steps (/set-role)

# Configuration Phase

**When to use:** Setting up or adjusting your role and preferences

## Commands

`/show-role-context`

**Purpose:** Display current role and document loading status

**Shows:**

- Configuration hierarchy (both global and project configs)
- Current organizational level
- Current role (with source scope indicator)
- Documents that will load (✓ exists, ! missing, - excluded)
- Custom additions and removals

`/update-role-docs [+/-]file ... [--global|--project]`

**Purpose:** Customize which documents load for your role

**Syntax:**

- `+path/to/doc.md` - Add a document
- `-/quality-standards.md` - Remove a document (absolute path with /)
- `+new.md -old.md +another.md` - Multiple changes

**Supports:** Relative paths (relative to current directory), Absolute paths (from repository root, start with /)

`/init-role-docs [--reset]`

**Purpose:** Initialize or reset document references to role guide defaults

**Flags:** `--reset` - Reset to defaults, clearing customizations

**When to use:** First time setting up a role, want to reset customizations, role guide has been updated

## Agent: Role Guide Generator

### Role Guide Generator

**Invoked by:** `/create-role-guide [role-name]`

**Purpose:** Create custom role guides following established patterns

**What it creates:**

- Role overview and responsibilities
- Deterministic behaviors (AI MUST follow)
- Agentic opportunities (AI SHOULD suggest)
- Common workflows and example scenarios
- Document references
- Integration with other roles

# Daily Usage Phase

**When to use:** Working on projects with established configuration

## Commands

`/generate-document [type] [--auto]`

**Purpose:** Generate documents from templates using role context

**Supported Document Types:**

- **Technical:** ADR, TDD, API docs, operational runbook
- **Product:** PRD, feature spec, user story, product overview
- **Strategic:** OKRs, roadmap, strategy, vision
- **Standards:** Engineering standards, quality standards, security policy
- **Process:** Contributing guide, development setup, team handbook

**Flags:** `--auto` - Batch mode with minimal interaction

## Agent: Document Generator

**Document Generator**

**Invoked by:** `/generate-document`

**Purpose:** Generate high-quality organizational documents from templates

**Capabilities:**

- Reads user's role and role guide
- Accesses bundled document templates
- Understands organizational context and level
- Asks document-specific questions
- Generates documents with appropriate structure
- Places documents in correct location
- Updates cross-references

# Maintenance Phase

**When to use:** Validating setup, syncing updates, troubleshooting

## Commands

`/validate-setup [flags] [--global|--project]`

**Purpose:** Validate .claude directory structure and configuration

**Flags:**

- `--quick` - Essential checks only
- `--fix` - Auto-fix issues with confirmation
- `--silent` - No output unless issues found (for SessionStart hook)
- `--quiet` - One-line summary only (for SessionStart hook)
- `--summary` - Brief checklist of results

**Checks:**

- Directory structure exists and is complete
- JSON files are valid
- Role guides exist and are populated
- Reference integrity (roles, documents, templates)
- Cross-references between files

`/sync-template [flags] [--global|--project]`

**Purpose:** Synchronize template updates while preserving customizations

**Flags:**

- `--check-only` - Check for updates without applying (for SessionStart hook)
- `--quiet` - Minimal output with check-only
- `--preview` - Show changes without applying
- `--force` - Force update check even if recently checked

**What it does:**

- Compares your template version with registry
- Analyzes differences (new files, updates, conflicts)
- Creates automatic backup before changes
- Performs intelligent three-way merge
- Handles conflicts with user input
- Generates detailed migration report

`/create-role-guide [role-name]`

**Purpose:** Create custom role guides following organizational patterns

Invokes the Role Guide Generator agent to create a comprehensive role guide.

## Agents: Framework Validator & Template Sync

### Framework Validator

**Invoked by:** `/validate-setup`

**Purpose:** Comprehensive validation of .claude directory setup

**Special Modes:**

- **First-Run Mode:** Detects missing .claude, offers initialization
- **Silent Mode (--silent):** No output unless issues found
- **Quiet Mode (--quiet):** One-line summary only
- **Summary Mode (--summary):** Brief checklist

**Validation Checks:** Critical (directory exists, JSON valid), Important (multiple roles, current role valid), Quality (documents exist, no broken references)

### Template Sync

**Invoked by:** `/sync-template`

**Purpose:** Intelligent template synchronization with customization preservation

**Capabilities:**

- Version detection (compares current vs latest)
- Difference analysis (file-level and content-level diffs)
- Change categorization (safe to auto-apply, merge required, conflict, preserve)
- Backup creation (timestamped backup before changes)
- Intelligent merge (three-way merge, additive merge)
- Conflict resolution (presents options to user)
- Update tracking and validation

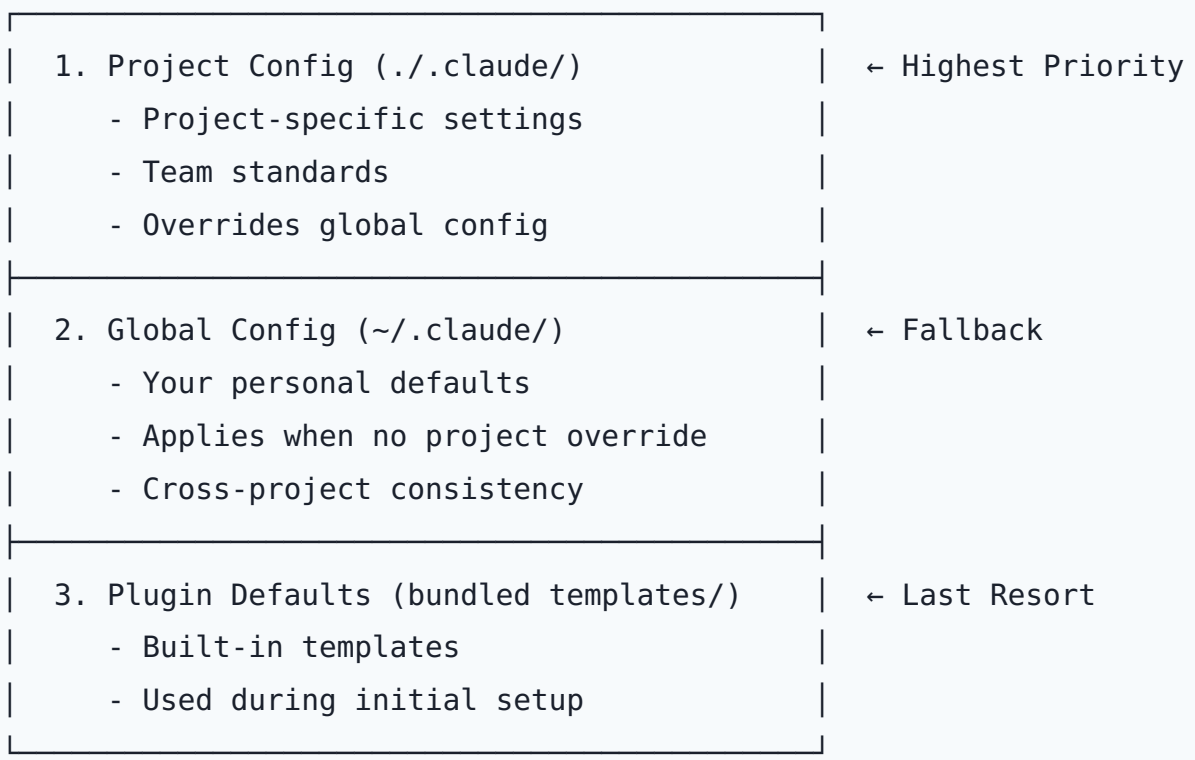
# Understanding Scope: How It Affects Your Reference Files

> **Scope determines where your configuration is stored and which settings take precedence.**

## The Three Scopes

| Scope | Location | Purpose | When to Use |
|---|---|---|---|
| **Global** | ~/.claude/ | Personal defaults that apply across all projects | You want consistent role and document settings everywhere |
| **Project** | ./.claude/ | Project-specific configuration that overrides global defaults | Your team has standardized roles and documents for a project |
| **Auto** | (Dynamic) | Automatically chooses project or global based on context | You want smart defaults without thinking about scope |

# Configuration Hierarchy




**Key Concept:**

Project configuration ALWAYS overrides global configuration when both exist. This allows teams to enforce standards while letting individuals maintain personal preferences elsewhere.

# Quick Reference

## All Slash Commands

| Command | Purpose | Key Flags |
|---|---|---|
| `/init-org-template` | Initialize organizational framework | --global, --project |
| `/setup-plugin-hooks` | Configure SessionStart hook | --global, --project |
| `/set-role` | Set your current role | --global, --project, --scope |
| `/set-org-level` | Set organizational level | --global, --project |
| `/show-role-context` | Display current configuration | (none) |
| `/update-role-docs` | Customize document references | --global, --project |
| `/init-role-docs` | Reset to role guide defaults | --reset |
| `/generate-document` | Generate documents from templates | --auto |
| `/create-role-guide` | Create custom role guide | (none) |
| `/validate-setup` | Validate .claude directory | --quick, --fix, --silent, --quiet, --summary |
| `/sync-template` | Synchronize template updates | --check-only, --quiet, --preview, --force |

## All Agents

| Agent | Invoked By | Purpose |
|---|---|---|
| Template Setup Assistant | /init-org-template | Guide template selection and setup |
| Role Guide Generator | /create-role-guide | Create custom role guides |

| Agent | Invoked By | Purpose |
|---|---|---|
| Document Generator | /generate-document | Generate organizational documents |
| Framework Validator | /validate-setup | Validate .claude directory setup |
| Template Sync | /sync-template | Synchronize template updates |

## Key Configuration Files

| File | Purpose | Contains |
|---|---|---|
| preferences.json | User preferences | Current role, auto_update_templates, applied_template info |
| role-references.json | Team defaults | Default document references per role |
| role-references.local.json | Personal customizations | User-specific document additions/removals (gitignored) |
| organizational-level.json | Org level tracking | Current organizational level (company/system/product/project) |
| settings.json | Hook configuration | SessionStart hook commands |

# Common Patterns

## Pattern 1: Individual Developer (Global Only)

```
/init-org-template --global
/set-role software-engineer --global
# Works everywhere automatically
```

## Pattern 2: Team Project (Project Only)

```
cd team-project
/init-org-template --project
/set-role qa-engineer --project
git add .claude/
git commit -m "Add team configuration"
```

## Pattern 3: Hybrid (Recommended)

```
# Global defaults for personal work
/set-role software-engineer --global

# Override for specific projects
cd special-project
/set-role devops-engineer --project
```

# SessionStart Hook

**Purpose:** Automatic validation and update checks when starting a new session

**Default Configuration (.claude/settings.json):**

```
{
  "hooks": {
    "SessionStart": [
      "/validate-setup --quiet",
      "/sync-template --check-only"
    ]
  }
}
```

### What Happens:

- `/validate-setup --quiet` - Validates setup, shows one-line summary
- `/sync-template --check-only` - Checks for updates (respects auto_update_templates preference)

### Example Outputs:

```
✓ Setup valid
✓ Template up-to-date (software-org v1.0.0)
```

Or:

```
⚠ Setup incomplete - run /init-org-template to initialize
ℹ Template update available (v1.0.0 → v1.1.0). Run /sync-template to update.
```

---

For more information, visit the plugin repository or check the documentation in your .claude/docs/ directory.

Generated with Claude Code • Role Context Manager Plugin v1.3.0