

/sp-using-git-worktrees

Source: `~/ .claude/skills/sp-using-git-worktrees/SKILL.md`

name: using-git-worktrees **description:**
Use when starting feature work that needs isolation from current workspace or before executing implementation plans - creates isolated git worktrees with smart directory selection and safety verification

Using Git Worktrees

Overview

Git worktrees create isolated workspaces sharing the same repository, allowing work on multiple branches simultaneously without switching.

Core principle: Systematic directory selection + safety verification = reliable isolation.

Announce at start: "I'm using the using-git-worktrees skill to set up an isolated workspace."

Directory Selection Process

Follow this priority order:

1. Check Existing Directories

```
# Check in priority order
ls -d .worktrees 2>/dev/null    # Preferred (hidden)
ls -d worktrees 2>/dev/null    # Alternative
```

If found: Use that directory. If both exist, `.worktrees` wins.

2. Check CLAUDE.md

```
grep -i "worktree.*director" CLAUDE.md 2>/dev/null
```

If preference specified: Use it without asking.

3. Ask User

If no directory exists and no CLAUDE.md preference:

```
No worktree directory found. Where should I create worktrees?

1. .worktrees/ (project-local, hidden)
2. ~/.config/superpowers/worktrees/<project-name>/ (global location)

Which would you prefer?
```

Safety Verification

For Project-Local Directories (.worktrees or worktrees)

MUST verify directory is ignored before creating worktree:

```
# Check if directory is ignored (respects local, global, and system gitignore)
git check-ignore -q .worktrees 2>/dev/null || git check-ignore -q worktrees 2>/dev/null
```

If NOT ignored:

Per Jesse's rule "Fix broken things immediately":

1. Add appropriate line to .gitignore
2. Commit the change
3. Proceed with worktree creation

Why critical: Prevents accidentally committing worktree contents to repository.

For Global Directory (~/.config/superpowers/worktrees)

No .gitignore verification needed - outside project entirely.

Creation Steps

1. Detect Project Name

```
project=$(basename "$(git rev-parse --show-toplevel)")
```

2. Create Worktree

```
# Determine full path
case $LOCATION in
  .worktrees|worktrees)
    path="$LOCATION/$BRANCH_NAME"
    ;;
  ~/.config/superpowers/worktrees/*)
    path="$HOME/.config/superpowers/worktrees/$project/$BRANCH_NAME"
    ;;
esac

# Create worktree with new branch
git worktree add "$path" -b "$BRANCH_NAME"
```

```
cd "$path"
```

3. Run Project Setup

Auto-detect and run appropriate setup:

```
# Node.js
if [ -f package.json ]; then npm install; fi

# Rust
if [ -f Cargo.toml ]; then cargo build; fi

# Python
if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
if [ -f pyproject.toml ]; then poetry install; fi

# Go
if [ -f go.mod ]; then go mod download; fi
```

4. Verify Clean Baseline

Run tests to ensure worktree starts clean:

```
# Examples - use project-appropriate command
npm test
cargo test
pytest
go test ./...
```

If tests fail: Report failures, ask whether to proceed or investigate.

If tests pass: Report ready.

5. Report Location

```
Worktree ready at <full-path>
Tests passing (<N> tests, 0 failures)
Ready to implement <feature-name>
```

Quick Reference

Situation	Action
<code>.worktrees/</code> exists	Use it (verify ignored)
<code>worktrees/</code> exists	Use it (verify ignored)
Both exist	Use <code>.worktrees/</code>
Neither exists	Check CLAUDE.md → Ask user
Directory not ignored	Add to <code>.gitignore</code> + commit
Tests fail during baseline	Report failures + ask
No <code>package.json/Cargo.toml</code>	Skip dependency install

Common Mistakes

Skipping ignore verification

- **Problem:** Worktree contents get tracked, pollute git status
- **Fix:** Always use `git check-ignore` before creating project-local worktree

Assuming directory location

- **Problem:** Creates inconsistency, violates project conventions
- **Fix:** Follow priority: existing > CLAUDE.md > ask

Proceeding with failing tests

- **Problem:** Can't distinguish new bugs from pre-existing issues
- **Fix:** Report failures, get explicit permission to proceed

Hardcoding setup commands

- **Problem:** Breaks on projects using different tools
- **Fix:** Auto-detect from project files (`package.json`, etc.)

Example Workflow

You: I'm using the using-git-worktrees skill to set up an isolated workspace.

[Check .worktrees/ - exists]

[Verify ignored - git check-ignore confirms .worktrees/ is ignored]

[Create worktree: git worktree add .worktrees/auth -b feature/auth]

[Run npm install]

[Run npm test - 47 passing]

Worktree ready at /Users/jesse/myproject/.worktrees/auth

Tests passing (47 tests, 0 failures)

Ready to implement auth feature

Red Flags

Never:

- Create worktree without verifying it's ignored (project-local)
- Skip baseline test verification
- Proceed with failing tests without asking
- Assume directory location when ambiguous
- Skip CLAUDE.md check

Always:

- Follow directory priority: existing > CLAUDE.md > ask
- Verify directory is ignored for project-local
- Auto-detect and run project setup
- Verify clean test baseline

Integration

Called by:

- **brainstorming** (Phase 4) - REQUIRED when design is approved and implementation follows
- **subagent-driven-development** - REQUIRED before executing any tasks
- **executing-plans** - REQUIRED before executing any tasks
- Any skill needing isolated workspace

Pairs with:

- **finishing-a-development-branch** - REQUIRED for cleanup after work complete

Revision #5

Created 2026-02-18 08:39:56 UTC by John

Updated 2026-06-21 20:01:02 UTC by John