Claude Code
Features Guide

Part 2 of the Claude Code Series

Claude Code Plan Mode: Think Before You Build

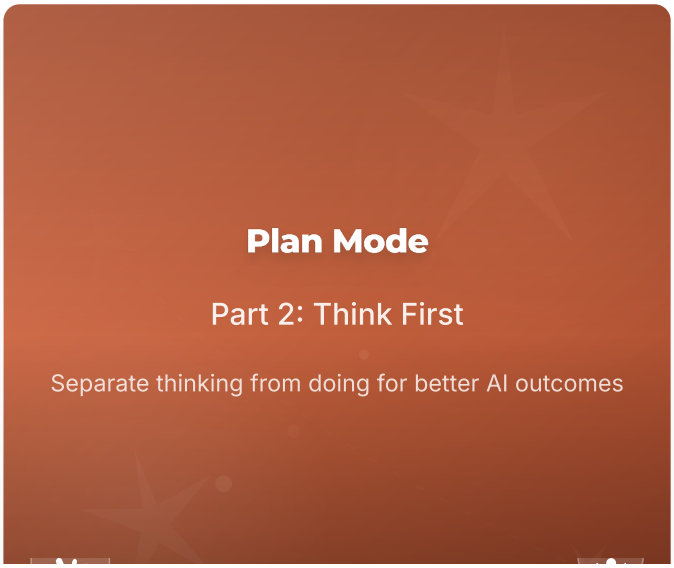
Claude Code Plan Mode separates thinking from execution. Learn how to use Shift+Tab to plan changes before making them - with real agency workflow examples.

9 February 2026 10 min read

Every developer has experienced it. You ask an AI assistant to make a change, and before you can blink, it has edited seven files, refactored a module you did not ask about, and introduced a subtle bug three layers deep.

The problem is not intelligence. The problem is impulse. Most AI coding tools jump straight from understanding to execution, skipping the part where a human would pause, think, and plan.

Claude Code's Plan Mode fixes this. It forces the AI to stop, explore, and present its plan before touching a single line of code. And after integrating it into our agency workflows, we think it might be the most underrated feature in the entire update.



What Is Plan Mode?

Plan Mode is a read-only exploration state in Claude Code. When activated, the AI can read files, search the codebase, analyse dependencies, and reason about architecture - but it cannot write, edit, or execute anything.

Once it has formed a plan, it presents that plan to you in full. You review it, ask questions, request changes, and only when you approve does the AI exit Plan Mode and begin implementation.

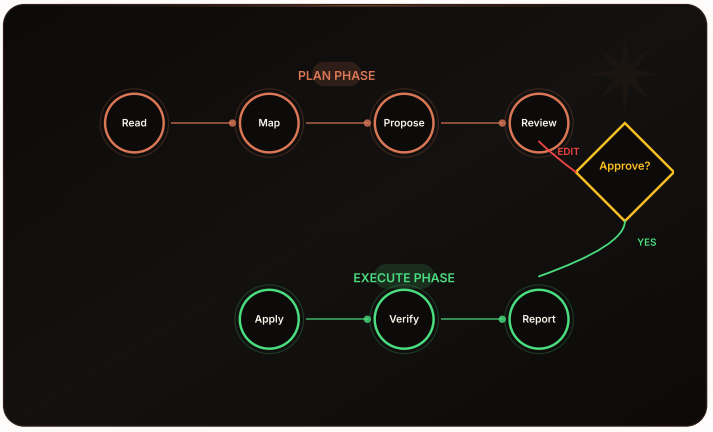
How to Activate It

There are three ways to enter Plan Mode:

1. **Shift+Tab twice** - The quickest method. Press Shift+Tab twice in the Claude Code interface to toggle Plan Mode on.
2. **The /plan command** - Type `` `/plan` `` followed by your request.
3. **Auto Plan Mode** - A setting that forces Claude Code to always start in Plan Mode before making changes. This is the "maximum safety" option.

When Plan Mode is active, you will see a visual indicator in the interface. The AI's responses will describe what it intends to do rather than doing it.

The Plan → Execute Workflow



Why Separating Thinking from Execution Matters

Consider the difference between an architect and a builder.

An architect surveys the site, studies the requirements, considers structural constraints, and produces detailed plans.

Only after the plans are reviewed and approved does the builder start pouring concrete.

Nobody wants a builder who shows up and starts pouring concrete based on a verbal description of what you want. Yet that is exactly how most AI coding tools operate - they hear your request and immediately start building.

Plan Mode gives Claude Code an architect phase. And that separation has three practical benefits:

1. You Catch Mistakes Before They Happen

When the AI presents its plan, you can spot problems before any code changes. Maybe it is planning to modify a shared utility that other parts of the application depend on. Maybe it is taking an approach that will create technical debt. Maybe it has misunderstood the requirement entirely.

In all these cases, it is infinitely cheaper to correct the plan than to undo the implementation.

2. You Stay in Control

Plan Mode keeps you in the decision-making seat. The AI does the research and thinking, but you approve the direction. This is especially important for:

- Production codebases where a bad change has real consequences
- Unfamiliar codebases where you need to understand the impact before committing
- Client projects where you need to explain and justify the approach

3. The AI Produces Better Work

This might seem counterintuitive, but constraining the AI to think first often produces better results than letting it act immediately. When Claude Code explores the codebase in Plan Mode, it builds a more complete understanding of dependencies, patterns, and constraints. The implementation that follows is more informed and less likely to break things.

Plan Mode: Think, Then Build

Shift+Tab activates Plan Mode. Claude analyses your request, maps dependencies, identifies affected files, and proposes an approach — without making any changes until you approve.

When to Use Plan Mode

Plan Mode adds a step to the workflow, so it is not appropriate for everything. Here is when we use it and when we skip it.

Use Plan Mode For:

New feature development - When you are adding something that touches multiple files or introduces new patterns. You want to see the AI's proposed architecture before it starts scaffolding.

Multi-file changes - Any change that spans more than two or three files benefits from a plan. The AI should explain which files it will modify, in what order, and why.

Architectural decisions - If a change involves choosing between approaches (REST vs GraphQL, SQL vs NoSQL, monolith vs microservice), Plan Mode lets the AI present the tradeoffs before committing to a direction.

Unclear requirements - When you are not entirely sure what you want, Plan Mode turns Claude Code into a thinking partner. It will ask clarifying questions and propose options rather than guessing and building.

Code review preparation - Before submitting a PR, you can use Plan Mode to have the AI review the changes and suggest improvements without automatically applying them.

Unfamiliar codebases - When working with code you did not write, Plan Mode lets the AI map out the relevant files, patterns, and dependencies before proposing changes.

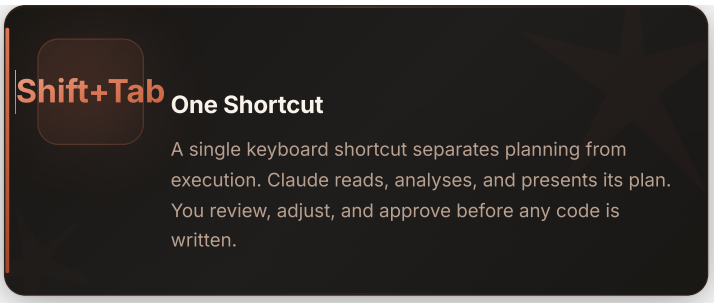
Skip Plan Mode For:

Typo fixes - If you are fixing a misspelled variable name, just let the AI do it.

Single-line changes - Adding an import statement or updating a version number does not need a planning phase.

Well-defined, isolated tasks - If the change is small, self-contained, and you know exactly what needs to happen, Plan Mode adds unnecessary friction.

Exploration and questions - If you are just asking the AI to explain code or answer questions (not make changes), Plan Mode is redundant since the AI is already in read-only mode.



Real Example: Planning Our Agent Teams Audit

Before running the Agent Teams case study we described in our Agent Teams guide, we used Plan Mode to scope the workstreams.

We activated Plan Mode and asked Claude Code to analyse the Mr Splash Plumbing project files - the site configurations, available data sources, audit frameworks, and known issues. Then we asked it to propose how to divide the work across a team of specialised agents.

The AI explored the project structure, reviewed the available Google Search Console data, checked the content across suburb pages, and mapped the technical stack. Then it presented a plan:

> *""Based on the project data, I recommend four workstreams:*
>
> 1. *GSC indexation analysis (data available for 7 active sites)*
> 2. *Location page content audit (43 suburb pages across the network)*
> 3. *Backlink and keyword opportunity mapping (Ahrefs export available)*
> 4. *Technical WordPress audit and task prioritisation (12 known pending items)*
>
> *Each workstream can run independently. The GSC analysis and technical audit may surface overlapping issues, which will serve as natural validation.""*

That last line turned out to be prophetic - both agents did find the same WordPress noindex issue independently.

The point is: Plan Mode gave us confidence that the team structure was sound before we spent tokens running four agents in parallel. We adjusted the briefs based on the plan, added specific data file references for each agent, and launched the team knowing the workstreams were properly scoped.

When to Use Plan Mode

Multi-file refactor	Always
New feature build	Always
Architecture changes	Always
Bug fix (single file)	Optional
Quick edit / typo	Skip

Rule of thumb: if the task touches more than 2 files, use Plan Mode.

Auto Plan Mode: Maximum Safety for Sensitive Projects

For projects where every change needs oversight, Claude Code offers Auto Plan Mode. When enabled, every request automatically starts in Plan Mode. The AI will never make changes without presenting a plan first.

We use Auto Plan Mode for:

- **Client production environments** - Where an unintended change could affect a live website
- **Security-sensitive work** - Where we need to review every file access and modification
- **Training and onboarding** - New team members can see exactly what the AI proposes before it executes, which builds understanding and trust

Auto Plan Mode can be toggled on and off in Claude Code's settings. It applies globally, so every interaction starts with planning.

Plan Mode in an Agency Context

Beyond individual development work, Plan Mode has transformed how we collaborate with clients at Jordan James Media.

Client Transparency

When a client asks "what changes are you making to my website?", we can show them the exact plan before anything happens. Plan Mode output serves as natural documentation of proposed changes.

For example, before running a batch of SEO optimisations, we activate Plan Mode and generate the full list of proposed title tag changes, meta description updates, and content modifications. The client reviews and approves. Only then do we execute.

This level of transparency builds trust and eliminates the "what did you change?" conversations that slow down project delivery.

Quality Assurance

Our internal workflow now mandates Plan Mode for any change that affects:

- Client-facing pages
- Database schemas
- API endpoints
- Authentication or permissions

The plan serves as a lightweight review step that catches issues before they reach formal code review. It is not a replacement for proper QA, but it is a remarkably effective first filter.

Pairing Plan Mode with Checkpoints

Plan Mode works beautifully alongside Claude Code's Checkpoints feature. The workflow looks like this:

1. **Plan Mode** - AI proposes changes
2. **You review** and approve the plan
3. **Checkpoint created** automatically before execution begins
4. **AI implements** the approved plan
5. **If something goes wrong** - restore from the checkpoint

This gives you a safety net at both the planning stage (catch it before it happens) and the execution stage (undo it if

something slips through).

Tips for Getting the Most from Plan Mode

After months of daily use, here are the practices that have made the biggest difference for us.

Be Specific About What You Want in the Plan

Instead of "plan how to add authentication", try "plan how to add Google OAuth to the Next.js app, including which files you'll modify, what packages you'll add, and how it integrates with the existing user model."

The more specific your planning request, the more useful the plan will be.

Ask Follow-Up Questions Before Approving

Plan Mode is a conversation, not a one-shot output. If the plan mentions modifying a file you care about, ask why. If it proposes an approach you are unfamiliar with, ask for alternatives. The AI will refine its plan based on your feedback.

Use Plan Mode to Learn Unfamiliar Code

When joining a new project, activate Plan Mode and ask the AI to explain the architecture. Since it can read files but not change them, it is a safe way to explore and understand a codebase without any risk of accidental modifications.

Keep Plans for Documentation

The plans Claude Code generates are useful artefacts. We save significant plans alongside our project documentation as

records of architectural decisions and their rationale.

Without vs With Plan Mode

Key Takeaways

1. **Plan Mode separates thinking from execution** - The AI explores and proposes before changing anything
2. **Shift+Tab twice** to activate, or use the `/plan` command
3. **Use it for complex changes** - New features, multi-file edits, architectural decisions, and unclear requirements
4. **Skip it for simple fixes** - Typos, single-line changes, and well-defined small tasks
5. **Auto Plan Mode** forces every interaction to start with a plan - ideal for production and client work
6. **We used it to scope our Agent Teams case study** - Planning the workstreams before launching parallel agents saved tokens and improved results
7. **Pairs naturally with Checkpoints** - Plan before execution, checkpoint before changes, restore if needed

---

PRO TIP

Auto Plan Mode for Sensitive Projects

Enable auto-plan in your project settings and Claude will always plan first for that project — even if you forget to hit Shift+Tab. Perfect for production codebases where an accidental change could be costly.

Ready to Work Smarter with AI-Powered Planning?

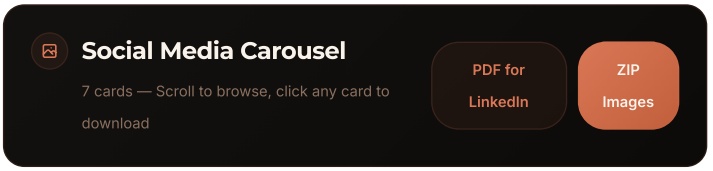
Jordan James Media uses Claude Code's advanced features to deliver more transparent, more reliable results for Australian businesses. If you are curious about what AI-first workflows look like in practice, let us show you.

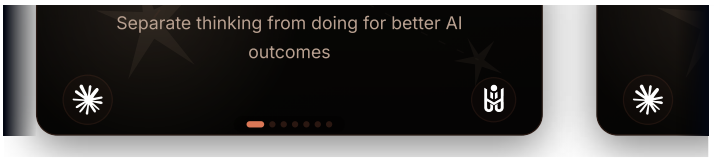
Talk to Us About AI-Powered Marketing

Related Reading:

- Claude Code's Biggest Update: 7 Features That Changed How We Build
- How to Use Claude Code Agent Teams: The Complete Guide
- Claude Code Checkpoints: Never Lose Your Work Again

The Safety Loop





Share This Article
Spread the knowledge

Twitter LinkedIn Copy

Build With Claude Code

We use Claude Code to ship faster, smarter, and at scale. Let us build your next project.

Get Your Free AI Strategy → App Development

Claude Code
8 parts

View All

Quick Links

→ Series Home
→ App Development
→ Get In Touch

Claude Code Stats

Model	Opus 4.6
Features	7
Series Parts	8

Agency Tested
Real projects, real results

Ready to outperform your competition?

Get your free AI strategy consultation today.

Start Your Free Consultation →

Jordan James Media
GROWTH STUDIO

Australia's AI-first digital marketing agency. We build intelligent systems that drive exponential growth.

info@jordanjamesmedia.com

0475 897 737

Sydney, Australia

SERVICES

- SEO Services
- Google Ads
- Website Design
- Local SEO
- Content Marketing
- App Development

COMPANY

- About Us
- Blog
- Case Studies
- Pricing
- Tools
- Contact

LOCATIONS

- Sydney
- Melbourne
- Brisbane

Perth

Adelaide

All Locations

All Locations →

Privacy Policy

Terms of Service

Portal Login

Built with Claude 4.6 Opus