

← Back to Series



Claude Code
Features Guide

Part 0 of the Claude Code Series

Claude Code's Biggest Update Yet: 7 Features That Changed How We Build Websites

We used Claude Code's new Agent Teams to audit 20+ sites in parallel. Here are the 7 features every developer and agency needs to know about in 2026.

📅 9 February 2026 ⌚ 14 min read

Last week, we used four AI agents running simultaneously to audit a local plumbing franchise with 20+ microsites across Sydney and Melbourne. One agent analysed Google Search Console indexing data. Another optimised location pages. A third ran deep SEO research. The fourth handled dev and content planning.

In under 15 minutes, they produced seven deliverables -- working in parallel, coordinating autonomously, and catching things that would have taken a human team days to find.

They found 300 pages blocked from Google by a single WordPress checkbox. They uncovered 97% duplicate content across 43 suburb pages. They identified 11 keywords sitting one position away from page one.

That is not a hypothetical. That happened on a Tuesday afternoon, and it fundamentally changed how we think about AI-assisted development at Jordan James Media.

The tool that made it possible? Claude Code -- specifically, a wave of updates from Anthropic throughout late 2025 and into 2026 that have turned it from a clever coding assistant into something closer to a digital workforce.

This post breaks down the seven features that matter most. We have written a dedicated deep-dive for each one, and we will link to all of them below. But first, here is the big picture.

What's in This Series

This is the hub post for our seven-part series on Claude Code's most impactful features. Each article goes deep on a single capability, with real examples from our agency workflow:

1. **Agent Teams: Running Multiple AI Agents in Parallel** -- The headline feature. Multiple Claude instances working simultaneously on different parts of a project.
2. **Plan Mode: Think First, Code Second** -- How structured planning before execution leads to dramatically better results.
3. **Checkpoints and Rewind: Fearless Development** -- Automatic save states that let you experiment without risk.
4. **Custom Subagents: Specialist AI on Demand** -- Creating purpose-built agents with their own tools, prompts, and permissions.
5. **The Skills System: Teaching Claude Your Workflows** -- Portable, reusable instructions that make Claude smarter about your specific processes.
6. **MCP Integration: Connecting Claude to Everything** -- The protocol that lets Claude talk to your databases, APIs, and external services.
7. **Background Tasks: AI That Works While You Work** -- Long-running autonomous processes that free you to focus elsewhere.

Whether you are a developer, an agency owner, or a business trying to understand what AI tools can actually do in 2026, this series will give you a practical, no-fluff look at each feature.

Claude Code: 7 Features

Complete Feature Guide

From assistant to autonomous AI workforce



Why This Update Matters

Claude Code is a terminal-based AI coding tool from Anthropic. It sits in your development environment, understands your entire codebase, and helps you write, debug, and ship code through natural language commands.

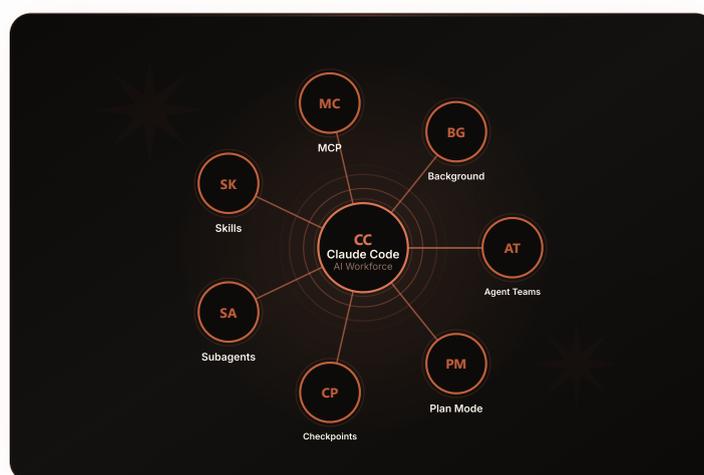
What changed in the past six months is the scale and ambition of what it can do.

In September 2025, Anthropic released Claude Code 2.0 alongside Claude Sonnet 4.5. That update introduced checkpoints, parallel subagents, a VS Code extension, and automation hooks. Then in early 2026, Claude Opus 4.6 arrived with Agent Teams -- the ability to run multiple independent Claude instances that coordinate through a shared task list and direct messaging.

Combined, these features represent a shift from "AI assistant that helps you code" to "AI workforce that can run parts of your business."

For agencies like ours, the implications are significant. We manage websites, run SEO campaigns, build dashboards, and produce content for multiple clients simultaneously. Having AI that can genuinely work in parallel -- not just autocomplete faster, but actually run four independent workstreams at once -- changes the economics of what a small team can deliver.

The Claude Code Ecosystem



Feature 1: Agent Teams

Deep dive: Claude Code Agent Teams: Running Multiple AI Agents in Parallel

Agent Teams is the feature that gets the most attention, and deservedly so. It lets you spin up multiple Claude Code instances, each with its own context window, and have them work on different parts of a project simultaneously.

Each agent operates independently. They can communicate with each other through direct messages. They share a task list where work items can be created, claimed, and completed. A lead agent coordinates the overall effort while team members handle their assigned pieces.

This is not theoretical. Anthropic themselves used 16 parallel agents to build a 100,000-line C compiler in Rust -- one capable of compiling the Linux kernel across x86, ARM, and RISC-V architectures.

How We Use It

At Jordan James Media, Agent Teams transformed our audit workflow. For the plumbing franchise project mentioned above, we configured a team with four specialist agents:

- **Agent 1 (GSC Analyst):** Pulled indexing data across 20+ microsites, identifying coverage issues and blocked pages.
- **Agent 2 (Location Optimiser):** Reviewed all 43 suburb-specific landing pages for content uniqueness, meta data, and schema markup.
- **Agent 3 (SEO Researcher):** Ran keyword gap analysis and competitor benchmarking across all target suburbs.
- **Agent 4 (Dev Planner):** Compiled findings into actionable development tickets with priority scoring.

All four ran concurrently. The lead agent collected their outputs, cross-referenced findings, and produced a unified report. Total time from kickoff to deliverables: under 15 minutes.

The strongest use cases we have found are research and review tasks where multiple angles need investigating, multi-module feature development where agents can each own separate components, and debugging sessions where competing hypotheses can be tested in parallel.

15 min

7 Deliverables

Four Claude Code agents produced 7 deliverables in under 15 minutes — that's AI-powered operations, not just AI-assisted coding.

Feature 2: Plan Mode

Deep dive: Claude Code Plan Mode: Think First, Code Second

Plan Mode separates thinking from doing. When activated, Claude analyses your request and codebase, then presents a detailed plan of action -- without making any actual changes until you approve.

This matters more than it sounds. The most common failure pattern with AI coding tools is not that they write bad code. It is that they start writing before they fully understand the problem. Plan Mode fixes that by creating an explicit step where Claude asks clarifying questions, maps out dependencies, and proposes an approach you can review and refine.

How We Use It

We use Plan Mode at the start of every complex task. Before refactoring a client's site architecture, before building a new dashboard feature, before running a multi-site audit -- Claude first produces a plan we can inspect.

The plan-then-execute cycle keeps us in control. We catch misunderstandings before they become wasted work. And because the plan is editable, we can steer Claude's approach before a single line of code gets written.

For agencies managing multiple client projects, this is essential. You cannot afford to let an AI agent go off on its own interpretation of a brief. Plan Mode builds in the review step that keeps quality consistent.



Agent Teams

Run 4+ AI agents in parallel, each with its own context, tools, and objectives. They coordinate without stepping on each other.

Feature 3: Checkpoints and Rewind

Deep dive: Checkpoints and Rewind: Fearless Development with Claude Code

Checkpoints automatically save your code state before every change Claude makes. If something goes wrong, you can instantly rewind to any previous point by pressing Escape twice or running the `~/rewind` command.

This sounds simple, and that is exactly why it is so powerful. Checkpoints remove the fear of experimentation. You can ask Claude to try ambitious refactors, test risky approaches, or explore architectural alternatives -- knowing that you are always one keystroke away from undoing everything.



Three Restore Options

When you rewind, you get three choices:

1. **Conversation only:** Go back to a previous prompt while keeping file changes intact.
2. **Code only:** Revert file changes while keeping the conversation history.
3. **Both:** Restore everything to a prior point in the session.

How We Use It

Checkpoints changed our willingness to experiment. Before this feature, we were cautious about asking Claude to attempt large-scale changes -- the cost of cleaning up a bad result was too high. Now, we regularly ask Claude to try two or three different approaches to the same problem, rewind between each attempt, and compare results.

It complements Git rather than replacing it. Think of checkpoints as session-level undo and Git as permanent history.



Feature 4: Custom Subagents

Deep dive: Custom Subagents: Building Specialist AI for Your Workflow

Custom subagents let you create purpose-built AI agents with their own system prompts, tool access, and permissions. When Claude encounters a task that matches a subagent's description, it delegates to that specialist, which works independently and returns results.

This is where Claude Code starts feeling less like a tool and more like an extensible platform. You are not limited to what Claude can do out of the box -- you can build specialised agents for your specific workflows.

How We Use It

We have built subagents for several recurring tasks:

- **SEO Auditor:** A subagent with access to our audit frameworks and checklists that can run standardised site audits.
- **Content Reviewer:** A subagent that checks blog drafts against our style guide, keyword targets, and SEO requirements.
- **Code Reviewer:** A subagent focused specifically on security, performance, and accessibility in client website code.

Each subagent has a focused prompt and limited tool access, which means it stays on task and produces consistent output. The main Claude instance handles orchestration -- deciding when to delegate and how to combine results.

7 GAME-CHANGING FEATURES

- 1 Agent Teams
- 2 Plan Mode
- 3 Checkpoints
- 4 Subagents
- 5 Skills System
- 6 MCP Integration
- 7 Background Tasks



Feature 5: The Skills System

Deep dive: [Claude Code Skills System: Teaching AI Your Processes](#)

Skills are portable, reusable sets of instructions that extend what Claude can do. They work across Claude Code, the Claude web interface, and the API. Think of them as recipes that teach Claude how to follow your specific processes.

The clever part is progressive disclosure. Claude first scans skill metadata -- descriptions and summaries -- to identify which skills are relevant to the current task. If a skill matches, Claude loads the full instructions. This keeps things efficient: Claude is not reading every skill file on every request, only the ones it needs.

How We Use It

Our agency has built skills for:

- **Client onboarding:** A step-by-step process for setting up new client projects, including folder structures, credential storage, and initial audits.
- **Blog publishing:** Our content production workflow from brief to draft to review to deployment.
- **Site launch checklist:** Everything that needs to happen before a client website goes live.

Skills mean that any Claude instance -- whether it is the main agent or a subagent or a teammate in an Agent Team -- can follow our agency's established processes without us re-explaining them each time.

The image shows a dark-themed comparison card with two sections. The top section, titled 'BEFORE AGENT TEAMS' with a minus sign icon, lists: One agent, one task at a time; Sequential execution; Hours per project; Manual coordination. The bottom section, titled 'WITH AGENT TEAMS' with a plus sign icon, lists: 4+ agents working simultaneously; Parallel execution; Minutes per project; Automatic coordination.

Feature 6: MCP Integration

Deep dive: [Claude Code MCP Integration: Connecting AI to Your Stack](#)

The Model Context Protocol (MCP) is the standard that lets Claude connect to external data sources and services. Where skills teach Claude what to do, MCP gives Claude access to the data it needs to do it.

Through MCP, Claude can talk to databases, APIs, content management systems, cloud platforms, and more -- all within a standardised, secure framework. MCP servers can dynamically update their available tools without requiring reconnection, which means your workflows adapt in real time.

How We Use It

MCP is the backbone of our dashboard system. We have MCP connections to:

- **Cloudflare:** Managing DNS, Workers, KV stores, and R2 buckets for client sites.
- **GitHub:** Accessing repositories, pull requests, and deployment pipelines.
- **Supabase:** Reading and writing to our agent management database.
- **Netlify:** Interacting with our serverless functions and blob storage.

When an agent needs to check a client's DNS configuration or deploy a Netlify function, it does not need to leave Claude Code. MCP provides the bridge, and Claude handles the rest.

PRO TIP



Plan Mode: Think Before You Build

Shift+Tab activates Plan Mode. Claude analyses your request, maps dependencies, and proposes an approach — without making changes until you approve. This alone prevents the #1 failure pattern with AI tools.

Feature 7: Background Tasks

Deep dive: Claude Code Background Tasks: AI That Works While You Work

Background tasks let Claude run long-running processes without blocking your current workflow. You can kick off a task -- a large codebase analysis, a comprehensive audit, a test suite -- and continue working on other things while it runs.

Before launching a background task, Claude Code prompts for any tool permissions the agent will need, ensuring it has approvals upfront. This is important: you are not giving carte blanche to an unsupervised process. You are pre-authorising specific capabilities and then letting the agent work within those boundaries.

How We Use It

Background tasks are perfect for the jobs that are important but not urgent -- the ones that used to sit in a backlog because no one had a spare hour. We use them for:

- **Codebase health checks:** Running security and performance analysis across all client repositories.
- **Content audits:** Scanning site content for outdated information, broken links, and SEO decay.

- **Data processing:** Transforming and analysing large datasets without tying up our primary Claude session.

The key benefit is not just time savings -- it is that work which previously did not happen at all now happens routinely.



Checkpoints: Your Safety Net

Automatic snapshots before every AI edit. Press Escape twice to rewind. Try ambitious refactors knowing you can always roll back.

The Bigger Picture: What This Means for Agencies and Businesses

These seven features do not exist in isolation. The real power is in how they combine.

Consider our plumbing franchise audit. We used **Agent Teams** to run four agents in parallel. Each agent followed a **Skill** that defined our audit methodology. They connected to client data through **MCP** integrations. The lead agent used **Plan Mode** to structure the overall approach before any work began. **Checkpoints** protected us from any missteps. **Custom Subagents** handled specialist tasks within each agent's workflow. And the entire process could have been kicked off as a **Background Task** while we worked on something else.

That is not seven separate features. That is an integrated system for getting work done at a scale and speed that was not possible 12 months ago.

For Agencies

If you run a digital marketing or web development agency, these tools change what a small team can promise and deliver. Audits that took days take minutes. Multi-site deployments that required careful sequential coordination can run in parallel. Repetitive tasks that ate into your team's creative time are handled autonomously.

For Businesses

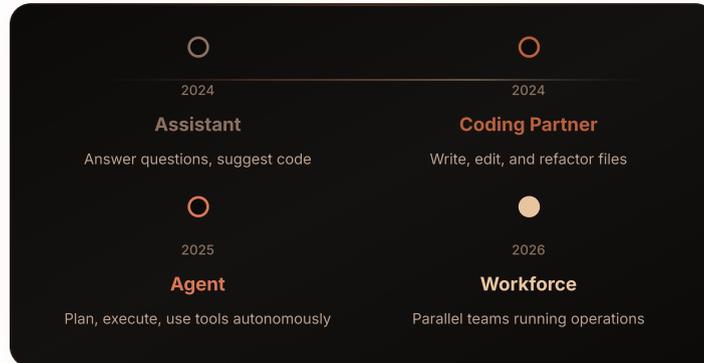
If you are a business evaluating AI tools for your development workflow, Claude Code represents a meaningful step beyond code autocomplete. This is task-level automation -- not just suggesting the next line of code, but running entire workstreams with minimal supervision.

For Developers

If you are a developer who has been sceptical about AI coding tools, the checkpoint and plan mode features

specifically address the "trust" problem. You are never committed to an AI's output. You can always review the plan, always rewind the changes, and always maintain control.

The Evolution of Claude Code



Key Takeaways

- **Agent Teams** let you run multiple Claude instances simultaneously, coordinating through shared task lists and direct messaging. This is the single biggest productivity multiplier.
- **Plan Mode** separates thinking from execution, catching misunderstandings before they become wasted work.
- **Checkpoints** remove the risk of experimentation by letting you rewind to any previous state instantly.
- **Custom Subagents** let you build specialist agents tailored to your specific workflows and quality standards.
- **The Skills System** makes your processes portable and reusable across all Claude instances.
- **MCP Integration** connects Claude to your external tools, databases, and services through a standardised protocol.
- **Background Tasks** enable long-running autonomous work that frees your team to focus on higher-value activities.
- Combined, these features represent a shift from AI-assisted coding to AI-powered operations.

“

The most important shift isn't any single feature — it's that Claude Code now feels less like a tool and more like a team.

— Jordan James Media

Ready to See What Claude Code Can Do for Your Business?

At Jordan James Media, we have been building with Claude Code since its early days. These features are not just tools we write about -- they are how we deliver results for our clients every day.

If you are an Australian business or agency looking to integrate AI into your development and marketing workflows, we would love to chat about what is possible.

[Get in touch with Jordan James Media](#) to discuss how AI-first development can work for your team.

Build With Claude Code

Let us show you what's possible

[Get Your Free AI Strategy](#)



Related Reading

Dive deeper into each feature with our dedicated guides:

- [Claude Code Agent Teams: Running Multiple AI Agents in Parallel](#)
- [Claude Code Plan Mode: Think First, Code Second](#)
- [Checkpoints and Rewind: Fearless Development with Claude Code](#)
- [Custom Subagents: Building Specialist AI for Your Workflow](#)
- [Claude Code Skills System: Teaching AI Your Processes](#)
- [Claude Code MCP Integration: Connecting AI to Your Stack](#)
- [Claude Code Background Tasks: AI That Works While You Work](#)



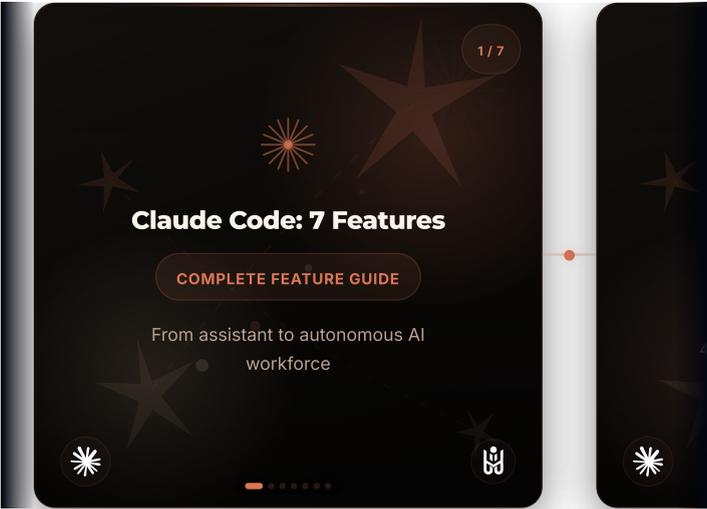
Key Takeaway



- 1 Agent Teams let you parallelise work across multiple AI agents
- 2 Plan Mode separates thinking from execution for safer outcomes
- 3 Checkpoints provide instant rollback — experiment without fear
- 4 Custom Subagents let you build specialist AI workers
- 5 Skills codify your team's expertise into reusable instructions
- 6 MCP connects Claude to your entire tech stack
- 7 Background Tasks run autonomously while you do other work

 **Social Media Carousel**
7 cards — Scroll to browse, click any card to download

[PDF for LinkedIn](#) [ZIP Images](#)



 **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

Part 0

Part 1

Part 2

Part 3

Part 4

Part 5

Part 6

Part 7

[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 →](#)



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 →

