Context Windows and Usage

Why long chats slow Claude down, and what it costs:

Don't Let Chats Get Too Long

Every chat has a **context window**, which is basically Claude's working memory for that conversation. It holds everything: your messages, Claude's replies, the documents you've attached, and anything pulled in from connectors. Once a chat starts approaching that limit, **Claude becomes very slow** and can lose track of details from earlier in the conversation.

The fix is simple: start a new chat. Before you do, ask Claude to "summarize this chat so I can pick it up in a new one" and paste that summary in. Anything you use over and over belongs in a Project, so you're not reattaching it every time.

Don't Count, Just Check Your Usage Page

You don't need to count tokens or understand the numbers below. **Just check your usage page** in Claude's settings to see how much you've used. If it's climbing fast, long chats and big attachments are usually the reason.

How the Models Compare

Model	Speed	Thinking	Context window	Price (per million tokens)
Fable 5.1	Slower	Adaptive (always on)	1M tokens	$10 input / $50 output
Opus 5	Moderate	Adaptive	1M tokens	$5 input / $25 output
Sonnet 5	Fast	Adaptive	1M tokens	$2 input / $10 output
Haiku 4.5	Fastest	Extended only	200K tokens	$1 input / $5 output

Why I say no to Haiku: adaptive thinking is needed. Adaptive thinking lets Claude decide on its own how hard to think based on how tricky your request is. Haiku doesn't have it, and its context window is also five times smaller, so it hits that slowdown point much sooner.

Why Output Costs More Than Input

- **Input** is everything Claude reads: your message, the entire chat so far, and every document you've attached.
- **Output** is what Claude writes back. It costs about **5x more** per token because writing a response is the heavy lift.
- **Even a 2 sentence message isn't small.** Every time you hit send, Claude rereads the whole conversation and all your attachments, so that short message can be thousands of tokens of input.

Quick tip: new topic, new chat. It keeps Claude fast, keeps your usage down, and keeps answers focused on what you're actually working on.