Gemini Context Extension

This extension provides three essential tools for managing your Gemini CLI usage with comprehensive model support.

Available Tools

1. track_context_usage

Analyzes your context window usage for any Gemini model:
- Total tokens used vs available
- Breakdown by component (system, tools, MCP servers, extensions, context files)
- Model-specific context window information
- Optimization recommendations

Parameters:
- `mode`: Output detail level - `compact`, `standard` (default), or `detailed`
- `model`: Model to analyze - `gemini-2.5-flash` (default), `gemini-2.5-pro`, `gemini-2.5-flash-lite`, `gemini-2.0-flash-exp`, `gemini-1.5-pro`, `gemini-1.5-flash`

Example usage:
- “How much context am I using?”
- “Show me detailed context usage for Gemini 2.5 Pro”
- “Analyze my context window for Gemini 1.5 Flash”

2. estimate_api_cost

Estimates API costs based on your current context with accurate pricing for all models:
- Per-request cost calculation with input/output breakdown
- Cost comparison across ALL Gemini models
- Savings analysis showing potential cost reductions
- Budget planning for multiple requests
- Smart recommendations for model selection

Parameters:
- `model`: Model to estimate costs for - `gemini-2.5-flash` (default), `gemini-2.5-pro`, `gemini-2.5-flash-lite`, `gemini-2.0-flash-exp`, `gemini-1.5-pro`, `gemini-1.5-flash`
- `requestCount`: Number of requests to estimate (default: 1)

Example usage:
- “What are my API costs?”
- “Estimate costs for Gemini 2.5 Pro with 100 requests”
- “Compare costs between models”

3. compare_gemini_models

Comprehensive comparison of ALL available Gemini models:
- Complete model information (names, descriptions, context windows)
- Pricing for each model (input/output token costs)
- Cost calculations for your current context

- Sorted by cost efficiency (cheapest first)
- Easy comparison to find the best model for your needs

Example usage:
- “Compare all Gemini models”
- “Show me a table of model pricing”
- “Which model is most cost-effective for my usage?”

Supported Models

Latest Generation (2.5 Series)

- **Gemini 2.5 Pro**: Most capable for complex reasoning and coding ($1.25-$2.50/M input, $10-$15/M output, 1M context)
- **Gemini 2.5 Flash**: Balanced speed and performance ($0.30/M input, $2.50/M output, 1M context)
- **Gemini 2.5 Flash-Lite**: Most cost-effective for high-volume tasks ($0.10/M input, $0.40/M output, 1M context)

Previous Generation

- **Gemini 2.0 Flash (Experimental)**: Experimental multimodal model ($0.10/M input, $0.40/M output, 1M context)
- **Gemini 1.5 Pro**: High-context model with 2M token window ($1.25-$2.50/M input, $5-$10/M output, 2M context)
- **Gemini 1.5 Flash**: Cost-efficient with long context support ($0.075-$0.15/M input, $0.30-$0.60/M output, 1M context)

Best Practices

When working with Gemini models:
1. **Start with comparison**: Use `compare_gemini_models` to understand all options
2. **Monitor context**: Use `track_context_usage` regularly to monitor token usage
3. **Optimize costs**: Review recommendations from `estimate_api_cost` for savings opportunities
4. **Choose wisely**:
- Use Flash-Lite for high-volume, cost-sensitive tasks
- Use Flash for balanced speed and performance
- Use Pro for complex reasoning and coding tasks
- Use 1.5 Pro when you need 2M token context window
5. **Clean up**: Consider disabling unused MCP servers to reduce context

Tips

- Context files (GEMINI.md) are discovered hierarchically from your working directory up to project root
- MCP servers contribute significantly to context size (~5k tokens per server)
- Extensions with context files add to your baseline token usage
- Pricing has tiered rates for some models based on prompt size (check model comparison for details)
- The extension automatically calculates costs using the correct pricing tier for your usage