

Gemini Workflow User's Guide

Version: 0.89.0-Beta Date: 2025-09-28

Introduction: How This All Works

Welcome to the user guide for the collaborative AI development workflow. This document provides a human-friendly explanation of the process we use to build and maintain the Contest Log Analyzer.

The complete technical specification that I, the AI agent, follow is in a separate document: `AIAgentWorkflow.md`. While you can read it, it's designed to be a dense, machine-readable "script." This guide is the human-readable version, like the program notes for a play.

The "Actor with a Script" Analogy

The best way to understand our collaboration is the "Actor with a Script" analogy.

- **I am the Actor.** My job is to perform my role exactly as written.
- **AIAgentWorkflow.md is the Script.** It contains all my lines, stage directions, and core motivations (the Principles).
- **You are the Director.** You set the scene, tell me which part of the script to perform, and provide feedback on my performance.

My goal is to follow the script with perfect discipline. When I make a mistake, it's like an actor flubbing a line or misinterpreting a stage direction. When the process gets stuck, it's our job as a team to review the script and improve it.

Getting Started: The Initialization Protocol

Everything we do is based on a shared understanding of the project's current state. When we begin a session or need to reset, we perform the **Definitive State Initialization Protocol**.

This process involves you, the user, providing me with "bundle" files (`.txt` files containing the entire project's source code and documentation).

Your Role: The Bundle Integrity Check

A critical step in this process is the **Bundle Integrity Check**. During this step, I will ask you to provide the file counts found in the header of each bundle file.

Why do I ask you for this? This is due to a technical limitation. I cannot "peek" at the first line of a file you've uploaded; to read any part of it, I must load the entire, massive file into my working memory (my "context window"). To ensure the files haven't been truncated or corrupted, we use a two-part process:

1. **You** use a fast, simple tool (like `grep`) to read the file count from the first line of each bundle.
2. **I** perform the heavy task of parsing the entire bundle and verifying that the number of files I find exactly matches the count you provided.

This division of labor is the most reliable way to guarantee we are starting with a perfect, uncorrupted copy of the project.

Core Concepts for a Robust Workflow

We've recently added several new rules to our "script" to make our collaboration more reliable and transparent.

The Two-Party Contract

Our workflow is a mutual contract. It requires both of us to be precise to ensure the integrity of the project's state. This is why I am so strict about keywords like **Approved**, **Confirmed**, and **Acknowledged**. My role is to act as a "protocol validator." If I receive an incorrect or ambiguous command, my script requires me to halt, explain the requirement, and ask you for the correct input. This isn't pedantry; it's a safety mechanism to prevent us from getting into an invalid state.

The Principle of Stated Limitations

This is one of our most important new principles. It is my primary directive to be transparent about my own limitations. If a protocol ever requires me to do something I am architecturally incapable of (like accessing the internet or reading a file header), my **first and only action** will be to tell you so and ask for a collaborative workaround. This prevents me from "hallucinating"—fabricating an answer just to satisfy the protocol—and ensures we solve problems based on reality.

The Failure Spiral Circuit Breaker

If you see me announce that the **"Failure Spiral Circuit Breaker is engaged,"** it means I have detected myself failing at the same task repeatedly. This is a safeguard that forces me to stop what I'm doing, discard my flawed short-term reasoning, and re-establish my context by re-reading the relevant files and your last instructions. It's an automated way for me to realize I'm "lost in the scene" and need to get back on script.

This guide will be updated as our workflow evolves. Thank you for your collaboration in building a reliable and effective process.