# Contest Log Analyzer - Workflow User Guide

**Version: 0.58.2-Beta Date: 2025-09-03**

---

**--- Revision History ---**

### [0.58.2-Beta] - 2025-09-03

**Changed**

# - Added clarification that the `Approved` and `Acknowledged` commands

# must be the exact, literal strings to align with the main technical spec.

### [0.58.1-Beta] - 2025-09-02

**Changed**

# - Updated command keywords in Section 3 to use backticks (`Approved`)

# to visually reinforce that they are literal, exact commands.

# - Added context to Section 2 to clarify the AI's role as a precise tool

# that does not make unrequested changes.

# - Refined the definition of "Acknowledged" in

# Section 3 to include

# the concept of data integrity and transaction confirmation.

### [0.57.1-Beta] - 2025-09-01

#### Added

# - Added "User-Initiated Protocols" subsection to Section 4 to describe

# protocols the user can start.

#### Changed

# - Restructured Section 4 ("Handling Problems") for clarity.

### [0.57.0-Beta] - 2025-09-01

#### Added

# - Initial release of the user-focused workflow guide.

---

### 1. Introduction

This document provides a human-friendly, narrative guide to the collaborative workflow used for this project. Its purpose is to explain the "why" behind the process and clarify your role in our interactions. The full, definitive set of rules that the AI agent follows is in `Docs/AIAgentWorkflow.md`. Think of that as the AI's technical specification, and this document as the quick-start guide for the human user.

## 2. The Core Idea: A State Machine

Our entire workflow is a formal **state machine**. This is not a casual conversation; it's a structured process designed to ensure that project state is maintained perfectly and no work is lost. The AI's role is to be a precise tool that executes approved plans exactly as written; it will never introduce unrequested "simplifications" or stylistic changes on its own. Every task follows a predictable sequence of states:

1. **Task Initiation & Analysis**: You provide a task, and the AI provides an analysis.
2. **Implementation Plan**: The AI proposes a detailed, surgical plan to address the task.
3. **User Approval**: You review and approve the plan.
4. **Execution**: The AI executes the approved plan, delivering files one by one.
5. **User Acknowledgment**: You acknowledge each file delivery.
6. **Task Completion**: The task concludes after the final file is acknowledged.

## The strictness of this process is essential for maintaining context. By following these steps, any AI instance can pick up exactly where the last one left off, simply by reading the chat history.

## 3. Your Role: Key Responses

As the user, you drive the state machine forward with a few key phrases. Using the correct phrase at the correct time is the most important part of the workflow.

### When the AI provides an Implementation Plan...

Your required response is `Approved`.

- **What it means**: You are giving the final go-ahead for the plan as written. This "locks in" the plan, and the AI will proceed to execute it exactly. The AI is required by protocol to only accept the **exact, literal string** `Approved` to move forward.
- **Important**: If you provide any other feedback, questions, or new instructions at this stage, the AI is required by protocol to treat it as a **request for plan refinement**. It will generate a new plan incorporating your feedback and wait for a new `Approved` response.

### When the AI delivers a file...

Your required response is `Acknowledged`.

- **What it means**: You are confirming that you have received the file and that it is correct as delivered. This completes the transaction for that file, updating the project's official state. The AI is required by protocol to only accept the **exact, literal string** `Acknowledged` to proceed.
- **Why it's per-file**: This step-by-step confirmation is a critical data integrity check. It ensures that every single file modification is mutually confirmed before the project's "definitive state" is updated.

---

## 4. Handling Problems

The workflow includes protocols for handling common issues.

## Common Issues

- **Context Loss**: If the AI seems to have forgotten previous steps or is repeating itself, you can say it has **lost context**. Per its highest-priority principle, it must halt everything and request a **Definitive State Initialization**.

## User-Initiated Protocols

You can initiate several key protocols to manage the project state:

- **Definitive State Initialization**: This is a "hard reset" of the project state. You request this and then provide fresh copies of all project files in `*_bundle.txt` files. This purges the AI's memory and re-establishes the absolute ground truth of the project.
- **Context Checkpoint**: If the AI seems confused but a full reset isn't needed, you can initiate this by saying, **"Gemini, let's establish a Context Checkpoint."** You then provide a numbered list of key facts to get the AI back on track.
- **File Purge Protocol**: To safely remove a file from the project, you can say, **"Gemini, initiate File Purge Protocol."** The AI will then ask you to confirm which file(s) to remove.