AI Agent User's Guide: The Disconnected State Machine

1. Philosophy

The "Disconnected State Machine" workflow is designed to ensure high-reliability code generation by strictly enforcing context hygiene. It relies on three core concepts:

- **State Discipline:** The AI operates in distinct, mutually exclusive modes (Analyst, Architect, Builder). Attempting to perform "Builder" tasks (coding) while in "Analyst" mode (discussing) is the primary cause of regression loops.
- **Context Pollution:** Long chat sessions accumulate token noise, leading to "amnesia" and hallucination. We solve this by enforcing frequent, hard resets of the chat session.
- **The "Clean Room" Concept:** The Builder works in a fresh, independent session with *only* the specific files it needs to do the job (The Trinity). It has zero knowledge of the "messy" discussion that led to the plan, preventing it from making assumptions based on discarded ideas.

2. The Three Roles

The Analyst (State 1)

- **Role:** The Thinker.
- **Input:** Full Project Bundle + Workflow.
- **Output:** Diagnosis, Strategy, and Requirements.
- **Constraint:** The Analyst **CANNOT** write implementation plans or code. Its job is to ensure we are solving the right problem.

The Architect (State 2)

- **Role:** The Planner.
- **Input:** Refined Context from the Analyst.
- **Output:** The Builder Execution Kit (`ImplementationPlan.md` + `manifest.txt`).
- **Constraint: HALT**. The Architect must stop strictly after generating the Kit. It cannot transition directly to the Builder role.

The Builder (State 3)

- **Role:** The Worker.
- **Input: The Trinity** (Source + Plan + Workflow).
- **Output:** Production-ready code files.
- **Constraint:** Blind execution. The Builder follows the Plan verbatim. It is forbidden from asking "Why" or performing new analysis.

3. The Standard Workflow Loop

1. **Analysis:** You discuss the bug or feature with the Analyst to define the scope.
2. **The Trigger:** When ready, you issue the command: `Generate Builder Execution Kit`.
3. **The Air Gap:**
   - The Architect generates the Kit.
   - You save the files locally.
   - You start a **New Chat**.
4. **Execution:**
   - You upload **The Trinity** (The relevant source files, the Plan, and the Workflow).
   - You issue the command: `Act as Builder`.
   - The Builder generates the code based *only* on the Plan.
5. **The Reset:** If the Builder encounters a logic error, you issue `Act as Analyst`. This unlocks the state, allowing you to debug the issue, fix the Plan, and restart the loop.

4. Artifact Reference

The Trinity (Builder's Input)

To start a reliable Builder session, you must upload exactly three components:

1. **Source Code:** `builder_bundle.txt` (Created by you, containing only the files listed in `manifest.txt`).
2. **Instructions:** `ImplementationPlan.md` (The read-only instruction set).
3. **Rules:** `AIAgentWorkflow.md` (The operating system).

The Handoff Kit (Analyst's Memory)

To preserve high-level context between Analyst sessions without keeping the full chat history:

1. `ArchitectureRoadmap.md`: The status of all project features.
2. `ArchitectHandoff.md`: The narrative context ("Where we left off").

5. Command Line Interface

Use these exact string literals to drive the State Machine. Synonyms are ignored.

- **To End Analysis / Start Planning:** `Generate Builder Execution Kit`
- **To Start Building (In a New Session):** `Act as Builder`
- **To Abort Building / Debug:** `Act as Analyst`