062: Context Storage & Durability Summary

Date: January 15, 2026
Status: Draft (Discovery Notes)

---

1. Overview

This document summarizes how the Universal Agent persists state, tool execution, and context across runs. The system combines SQLite-backed durability, tool-call idempotency, and file-based research artifacts to survive restarts or long-running tasks.

---

2. Runtime State Database

The durability layer persists to `AGENT_RUN_WORKSPACES/runtime_state.db` via the SQLite schema in `durable/migrations.py`.

Core tables:
• `runs`: current run spec, iteration count, completion promise, status
• `run_steps`: step-level state (phase, status, errors)
• `tool_calls`: prepared/running/succeeded tool calls + idempotency keys
• `tool_receipts`: pending receipts for crash recovery
• `checkpoints`: serialized snapshots (phase, cursor, corpus cache)

This enables recovery on restart and allows ledger reconciliation of in-flight tools.

---

3. Checkpoints & Corpus Cache

Checkpoints persist state snapshots and optionally corpus data. A notable case is `refined_corpus.md`, cached via `on_post_research_finalized_cache` so a restarted agent can resume synthesis without re-reading raw data.

Checkpoint use cases:
• Phase boundaries (pre-side-effect, post-replay)
• Research corpus caching (`refined_corpus.md`)
• Resume packet reconstruction

---

4. Tool Ledger + Idempotency

The tool ledger prepares tool calls with an **idempotency key** derived from normalized inputs and side-effect classification. It prevents replaying tools that already succeeded and supports crash-safe receipts:

1. `prepare_tool_call` inserts a prepared ledger row.
2. On success, tool responses are stored; external IDs (e.g., Gmail message id) are captured.

3. `tool_receipts` records pending responses before final commit.
4. On resume, pending receipts are promoted and replays are skipped when safe.

This is the basis for replay-safe external tool calls.

---

5. Context Handoff Mechanisms

The system uses **file-based context handoff** to avoid needing long chat transcripts:

• **Inbox pattern**: search results written to `search_results/` and processed into `tasks/<task>/refined_corpus.md`.
• **Evidence ledger** (optional): `build_evidence_ledger` writes `handoff.json` for context resets.
• **MessageHistory**: soft-truncation within a single session; harness handles cross-session context resets.

---

6. Key Observations

• Verification currently checks **artifact files**, not tool receipts. If a mission requires `email_sent_confirmation`, it must exist on disk even if Gmail returned a message id.
• Tool receipts capture external correlation IDs, which are valuable for verification but are not yet used by `TaskVerifier`.

---

7. Related Documents

• 031_LONG_RUNNING_HARNESS_ARCHITECTURE.md
• 050_HARNESS_SYSTEM_ARCHITECTURE.md
• 033_HARNESS_EMAIL_DEBUG_REPORT.md