Selected files

`1 printable files`

`CHANGELOG.md`

`CHANGELOG.md`

Changelog

All notable changes to this project will be documented in this file.

The format is based on Keep a Changelog, and this project adheres to Semantic Versioning.

[Unreleased]

Added

- New documentation: `docs/MEMORY_COMMAND_SUMMARY.md` providing a comprehensive overview of all memory-related commands.

Fixed

- **Recreate Flag Memory Safety Fix**: Resolved a critical memory safety vulnerability where the `--recreate` flag could lead to accidental data loss. The fix ensures that user memories are preserved by default and are only cleared when explicitly requested via the `--recreate` flag, with proper synchronization across both local SQLite and LightRAG graph memory systems. See ADR-009 for details.

[0.8.8] - 2025-07-10

Changed

- **CLI Refactor for Maintainability**: The `agno_main.py` file was refactored to improve maintainability and organization. Memory-related CLI commands and initialization logic were extracted into new modules (`src/personal_agent/cli/` and `src/personal_agent/core/agno_initialization.py`). This significantly reduced the size and complexity of the main CLI file, enhancing modularity, testability, and extensibility while maintaining full backward compatibility. See ADR-008 for details.
- **Comprehensive Memory System Technical Summary**: Updated `COMPREHENSIVE_MEMORY_SYSTEM_TECHNICAL_SUMMARY.md` to reflect the CLI refactor, enhanced memory tool CLI, and new testing infrastructure.

[0.8.7] - 2025-07-10

Added

- **Enriched Graph Ingestion Pipeline**: The `store_graph_memory` tool now uses a sophisticated, hybrid approach to build the knowledge graph. The agent performs local NLP (coreference resolution, entity/relationship extraction) to generate rich metadata, which is then embedded in a text file and uploaded to LightRAG. This guides the server's native ingestion pipeline, resulting in a more accurate and detailed graph. See ADR-007 for details.
- New `nlp_extractor.py` module for NLP tasks.
- New test suites (`test_knowledge_graph_relationships.py`, `run_kg_relationship_test.py`) to validate the new memory system.
- New `inject_eric_facts_v2.py` script to leverage the new memory system.

Changed

- The `store_graph_memory` tool was completely overhauled to use the new Enriched Graph Ingestion Pipeline, replacing the previous direct text upload.

[0.8.6] - 2025-07-10

Added

- The `docmgr` tool now supports a `--json` flag for machine-readable output.

Changed

- Project version bumped to `0.8.6`.

[0.8.52] - 2025-07-09

Added

- Implemented an intelligent Memory Restatement System to improve knowledge graph mapping. User facts are now converted to the third person for graph storage while remaining in the first person for local semantic memory. For more details, see ADR-004: Memory Restatement System.
- Added a comprehensive test suite for the Memory Restatement System (`tests/test_fact_restatement.py`).

Changed

- The `store_graph_memory()` method now uses restated content for file uploads to ensure proper entity mapping in the knowledge graph.

- The core agent's `_restate_user_fact()` method was enhanced with more robust pronoun and possessive conversion using regex with word boundaries.

[0.8.51] - 2025-07-09

Added

- New `show-config` tool with color-coded output and a `--json` option for viewing system configuration.
- Healthcheck for the `lightrag_memory_server` to monitor its operational status.

Fixed

- Increased service timeouts and added TCP keepalive settings in the `lightrag_memory_server` to prevent connection drops during long-running operations.

Chore

- Updated Python dependencies, including `lightrag-hku`, `googlesearch-python`, `pycountry`, and `pyyaml`.

[0.8.50] - 2025-01-09

Added

- **Unified Memory Management System** to clear semantic (local) and graph (LightRAG) memories simultaneously. Includes a core `MemoryClearingManager` and a `clear_all_memories.py` CLI tool with safety features like dry-run and verification. See ADR-005 for details.
- **Enhanced LightRAG Document Manager** (`docmgr.py`) using modern, API-based operations, eliminating the need for server restarts. Includes features like batch deletion, pattern matching, and a retry mechanism. See ADR-005 for details.

Removed

- Removed legacy and outdated scripts: `scripts/fix_lightrag_data.py` and `scripts/fix_lightrag_persistent_storage.py`.

[0.8.3] - 2025-07-05

Added

- **Knowledge Graph Memory System** and a **Unified Knowledge Architecture**. This introduces a dual memory paradigm combining local semantic search with a graph-based (LightRAG) memory system. See ADR-003 for the full architecture.

- **Knowledge Coordinator** to intelligently route queries between the local semantic and LightRAG graph systems.
- A dedicated, containerized **LightRAG Memory Server** for memory-specific operations with user isolation.

Changed

- The `store_user_memory` tool now persists memories to both the local and graph systems simultaneously.
- The `query_knowledge_base` tool now uses the Knowledge Coordinator for intelligent routing.
- The topic classifier was enhanced with whole-word matching and an expanded keyword dictionary for greater accuracy.

Fixed

- Resolved a Pydantic validation issue (`"file_path": null`) by switching to a file-upload-based approach for storing memories in the LightRAG graph.

[0.8.1] - 2025-07-04

Added

- The LightRAG service is now a **decoupled, standalone service** with its own Docker configuration, managed independently of the main application. For more details, see ADR-001.
- A `LIGHTRAG_URL` environment variable now acts as the single source of truth for the LightRAG server's address.
- The `switch-ollama.sh` script was enhanced to also switch the `LIGHTRAG_URL` between local and remote environments.

Fixed

- **Critical Multi-User Path Configuration:** Fixed a critical bug where all users were sharing the same data directories. The system now correctly generates user-specific paths for storage and knowledge, ensuring proper data isolation. See ADR-002 for details.

[0.7.10] - 2025-07-03

Added

- **LightRAG Document Manager V2** (`lightrag_docmgr_v2.py`), a rewritten document management tool that uses the core LightRAG library for more stable and reliable operations.
- The `send_to_lightrag.py` script was enhanced with SCP support for transferring files to remote LightRAG deployments.
- Foundational design for a **Multi-User Architecture** was created in `docs/MULTI_USER_DESIGN.md`.

[0.7.9] - 2025-07-01

Added

- A `--recreate` flag for the `paga_cli` command-line interface to enable on-demand recreation of the knowledge base.
- **Instruction-Level Performance Analysis System**, allowing the agent's instruction sophistication to be tuned for different models. Includes a four-tier `InstructionLevel` (MINIMAL, CONCISE, STANDARD, EXPLICIT).
- A comprehensive performance testing framework (`tests/test_instruction_level_performance.py`) for scientific-grade performance measurement.
- **Google Search** tool (`GoogleSearchTools`) was integrated into the agent.
- **Structured JSON Response System** using Ollama's JSON schema validation for more reliable tool call detection and response parsing.

Fixed

- **Critical LightRAG Timeout Fix:** Resolved an `httpx.ReadTimeout` error that occurred during large document ingestion by correcting the timeout value in the `lightrag` service's `config.ini`. See ADR-006 for details.