

Current State And Last 10 Hours Progress Report

Build line: DraftAtlas 0.5.0 | Date: 2026-07-23 | Scope: recent implementation window and latest visual map phase

1. Executive Summary

DraftAtlas is now a substantial local-first story planning, manuscript, editorial intelligence, and production-prep application. The app has moved well beyond a prototype planning board. Its strongest identity is now the combination of a Living Encyclopedia, Editorial Intelligence, local-first manuscript planning, structured exports, support/reporting, and a visual story map that can explain how story elements connect.

The latest completed phase strengthened the visual story map search and @ entry-point behavior. Focused graph search now behaves more like a command surface: it has an active result, clearer command copy, keyboard navigation, and consistent selection/focus behavior.

This report pauses feature work after that phase and summarizes what has been completed recently, what the app currently does, and what remains before broad beta distribution.

2. What Was Completed In The Latest Phase

This phase is complete and validated. It specifically improves the graph search / @ entry path rather than opening another feature lane.

- Added a shared command-summary model for focused story map search.
- Search and @ entity-picker states now explain what the user can do next.
- Search results now maintain an active selected result.
- Arrow Up and Arrow Down move through graph search results.
- Enter selects a visible result or focuses the map on an off-map result.
- Escape clears the graph search.
- Search result rows now show a visibly active state.
- Empty search state now uses the same command-summary language instead of a disconnected message.
- Added regression coverage for idle, ready, @ picker, and empty command states.

Files touched in this phase:

- src/lib/visualStoryMap.ts
- src/lib/visualStoryMap.test.ts
- src/components/DraftAtlasApp.tsx
- src/styles.css

Validation for this phase:

- npm test -- src/lib/visualStoryMap.test.ts: passed, 68 tests.

- npm test: passed, 219 tests.
- npm run build: passed.

3. Major Progress Over The Recent Work Window

The recent work window included several connected areas:

DraftAtlas Rebrand And Application Identity

- StoryForge was effectively superseded by DraftAtlas as the active product name.
- App branding, icons, title metadata, and website direction were shifted toward DraftAtlas.
- The app now uses a compass/map identity rather than the older generic book identity.
- Old StoryForge naming still has some historical holdouts in generated files and legacy internals, but the user-facing direction is DraftAtlas.

Website And Support Funnel

- The website was updated around DraftAtlas positioning.
- The landing page now has product sections for Dashboard, Project Library, Story Bible, Backlinks, Manuscript Mode, Editorial Intelligence, Cookbook layout planning, Backup, Health, and Diagnostics.
- Screenshot-style product visuals were added instead of generic generated art.
- Gallery images now open in a professional overlay with previous/next navigation and hover affordance.
- A tester-interest/contact form was added instead of exposing a public download link too early.

Support And Feedback Infrastructure

- Support & Feedback was strengthened from a simple email handoff into a structured reporting workflow.
- Reports can be generated in readable Markdown/JSON-style formats.
- Report privacy controls let users include or hide project details, project type, manuscript information, logs, diagnostics, and related context.
- The workflow is designed so reports can later be read by an AI support agent and converted into implementation briefs.
- Local report saving was added so the user has an attachable artifact instead of only an email body.

Private Beta Evidence And Offline Retention

- Local-only beta evidence tracking was added or extended.
- The app does not need accounts or telemetry to capture useful private-beta signals.
- Testers can manually export local usage summaries instead of the app uploading analytics.
- Evidence categories include sessions, active days, tutorial progress, project creation, backups, exports, Story Health runs, and feedback reports.

Engineering Foundation And Release Readiness

- Repository documentation and maintenance instructions were strengthened.
- CI and PR scaffolding were added in the repo state.
- Compatibility fixtures and release process documentation were introduced or improved.
- Private beta packaging and release evidence concepts were expanded.

- The app now has clearer internal expectations around beta proof, support reports, compatibility fixtures, and release validation.

Editorial Intelligence And Story Health

- Editorial Intelligence is established as a separate creative integrity system from Project Health.
- It distinguishes deterministic findings from AI opinions.
- Findings can be treated as review items rather than arbitrary percentage scores.
- The page now has actionable review paths, though this area still needs deeper workflow polish and better item-specific decision handling.

Cookbook Planning And Layout Intelligence

- Cookbook import/layout planning was improved to think in sections, recipes, repeated variations, and editorial structure.
- The app better recognizes that large local-Ollama tasks must be chunked instead of sent as one massive context request.
- The layout planner is moving toward a phased editorial assistant: source intake, recipe extraction, section assignment, duplicate/variation detection, page planning, and review operations.

Visual Story Map And Evidence Board

- This is one of the most important recent upgrades.
- Tutorial walkthroughs can deep-link into focused map overlays.
- Relationship importance explanations were added.
- Search and @ picker previews were improved.
- Selected-node association stacks were added.
- Node and relationship action toolkits were added.
- Lens summaries and action chips were added.
- Selected-node spotlight summaries were added.
- Selected-relationship verdict cards were added.
- Edge-label layering was corrected so labels are clickable instead of being intercepted by nodes.
- The latest phase adds command-palette behavior to graph search.

4. What DraftAtlas Is Now

DraftAtlas is a local-first Living Encyclopedia Studio for authors, family cookbook creators, worldbuilders, lore bible maintainers, and structured creative projects. It is not primarily a word processor, though it now includes Manuscript Mode.

Core identity:

- Plan the project.
- Build the Story Bible.
- Link every meaningful idea to every related idea.
- Draft with entity awareness.
- Detect creative and software health issues.
- Use optional AI without surrendering control.

- Export useful packets and backups.
- Keep user work local and private.

Current major systems:

- Dashboard and Project Library.
- Medium-aware project setup and guided workflows.
- Story Bible and backlinks.
- Manuscript Mode.
- Entity-aware @ mentions.
- Characters, chapters, scenes, spreads, recipes, organizations, locations, objects, symbols, themes, plot threads, mysteries, notes, tasks, assets, and research-style records.
- Editorial Intelligence / Story Health.
- Project Health and diagnostics.
- Visual Story Map / Evidence Board.
- Local AI and provider abstraction.
- Cookbook and picture-book layout planning.
- Backup, restore, support reports, and beta evidence.
- Website, brand assets, and private-tester contact path.

5. Current Strengths

- Strong product identity: local-first creative planning with editorial memory and a living encyclopedia.
- Broad medium support: novels, picture books, cookbooks, lore bibles, comics, screenplays, and short stories.
- Useful for real personal projects now, especially cookbook organization and story bible planning.
- The visual map is becoming a differentiating feature rather than a simple graph.
- Support and beta reporting are unusually thoughtful for a young app.
- Offline retention strategy fits the privacy promise.
- Tests and build currently pass after the latest phase.

6. Remaining Risks And Gaps

- The workspace is very dirty from broad rename and feature work; a clean branch/release cut is needed before distributing installers.
- Some legacy StoryForge holdouts may remain in internal file names, generated artifacts, support keys, and older report filenames.
- The app needs more real destructive testing: force-close during edits, corrupt backups, invalid restores, asset deletion, and recovery drills.
- The visual map is functionally much stronger, but still needs premium layout polish before it becomes a marketing-grade centerpiece.
- Editorial Intelligence review workflows need more mature approve/edit/reject handling for each category.
- Cookbook layout planning still depends on phased context management when local models have limited context.

- Printer-ready exports still need deeper artifact QA for real printers and editors.
- Mac packaging still needs clean-machine verification if Mac beta distribution is planned.
- Local AI provider behavior should continue to be tested against running and stopped Ollama states.

7. Commercial Readiness Assessment

Personal-use readiness: strong.

Private beta readiness: close, assuming the next pass is cleanup, packaging, and destructive QA rather than more features.

Commercial public launch readiness: not yet. The app has enough value, but needs proof: clean installer, clean workspace, no personal project leakage, backup/restore evidence, export artifact QA, support workflow validation, and known-limits documentation.

Estimated rating today:

- Feature depth: 9.0 / 10.
- Product identity: 9.1 / 10.
- Personal usability: 8.6 / 10.
- Private beta readiness: 8.0 / 10.
- Commercial polish: 7.5 / 10.
- Data-safety confidence before destructive QA: 7.2 / 10.

The ceiling is high. The next meaningful gains come less from adding features and more from proving that existing systems behave reliably under messy real-world use.

8. Recommended Next Steps

1. Freeze a clean beta branch and remove personal-project leakage from the distributable build.
2. Run destructive backup/restore and force-close testing.
3. Create one clean installer package and install it on a fresh Windows user profile.
4. Audit all remaining StoryForge naming holdouts and decide which are acceptable legacy internals versus user-facing defects.
5. Continue polishing the visual story map as a marketable feature: graph overlay, edge dossiers, and relationship-weight clarity.
6. Do cookbook layout QA on a real large document with phased chunking and visible progress.
7. Generate a fresh private beta package only after the above evidence is recorded.

9. Pause Point

The current visual story map command-search phase is complete and validated. Feature work should pause here until the user reviews the report and decides whether the next lane is beta packaging, destructive QA, visual-map polish, or cookbook layout reliability.