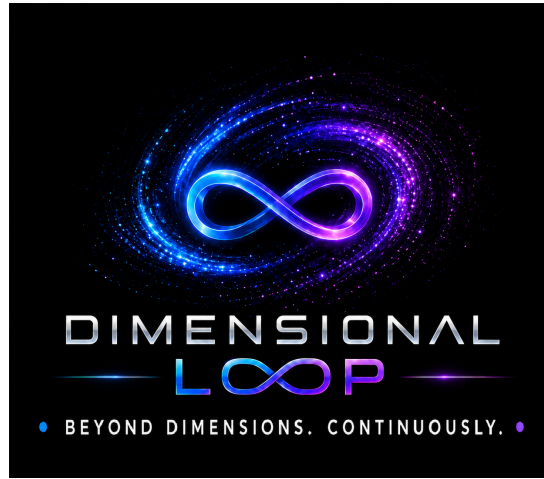




UPMXS USER GUIDE

Universal Project Memory Exchange System

Version 1.0

A practical guide to creating, preserving, exchanging and continuing project memory.

Powered by PME - Project Memory Engine

Created by Dimensional Loop

Copyright © 2026 Dimensional Loop

1. What UPMXS Is

UPMXS (Universal Project Memory Exchange System) is a desktop project-memory system designed to preserve the knowledge needed to understand and continue a project across time, tools and AI systems.

UPMXS stores structured project memory in a portable .upmx file. PME (Project Memory Engine) organizes the memory behind the application.

Core principle: Your project memory belongs to the project, not to the AI that helped create it.

2. Getting Started

1. Launch UPMXS and create a new project or open an existing .upmx project.
2. Enter or review the project identity and Continuation Context.
3. Add useful knowledge, documents and visual assets.
4. Use File > Save regularly to preserve the portable project file.
5. Use Help > Getting Started at any time for the in-app workflow guide.

3. Continuation Context

Continuation Context records where the project stands now. It can preserve the current goal, current state, current task, next step, blockers, constraints and critical decisions.

Use File > Continuation Context, enter only information supported by the project, then save. The context is stored in the .upmx file and restored when the project is reopened.

4. Working With AI - Two Paths

UPMXS is designed so project memory is not dependent on one AI provider.

Direct AI connection: Select a supported provider and configure credentials when required. Relevant Project Context can build a focused package of stored project knowledge for a particular question or task.

No direct API connection: Use AI > Project Knowledge Prompt, choose the appropriate project type and copy the provider-neutral prompt into an AI service you already use.

The V1.0 prompt instructs a capable AI service to create a downloadable .json file containing valid UPMXS Exchange JSON. Download that file and import it directly into UPMXS. This avoids manually creating a JSON file from chat text.

Some external AI products may not support downloadable file attachments. In that case, valid UPMXS Exchange JSON may still need to be saved as a .json file before import.

5. Importing AI Project Knowledge

1. Open the project that should receive the knowledge.
2. Choose AI > Import AI Knowledge.
3. Select the downloaded UPMXS Exchange JSON file.
4. Review the preview and validation results.
5. Proceed only when blocking validation errors have been resolved.

Imported knowledge can include project identity, Continuation Context, Project Progress, decisions, constraints, blockers, knowledge items and relationships.

6. Controlled Updates and Conflicts

UPMXS does not silently overwrite authoritative project memory when incoming AI knowledge overlaps with information already stored in the project.

Incoming information can be identified as New, Unchanged, Changed or Conflict. Where review is required, compare the Existing and Incoming values and choose Accept, Reject or Defer.

Accept applies the reviewed change. Reject keeps the existing project memory. Defer leaves the update available for later review so a decision does not have to be made immediately.

7. Memory History and Versioning

When accepted project knowledge changes, UPMXS can preserve the previous and new values together with supported information about the change. History helps explain how project memory evolved rather than showing only the latest value.

Where restoration is available, an earlier value can be restored without erasing history. The restoration itself becomes another recorded change.

8. Project Knowledge and Relationships

UPMXS supports specialized knowledge such as Characters, Stories, Pages, Assets, Tasks and Decisions, while also supporting project-specific types such as Requirement, Feature, Risk, Agent, ResearchFinding and other appropriate categories.

Relationships can use supported types such as Contains, References, Uses, AppearsIn, DependsOn, CreatedFrom and RelatedTo.

Use File > Knowledge Graph to inspect stored knowledge and relationships visually.

9. Documents and Portable Files

Use File > Import Document for supported TXT, Markdown, DOCX and text-based PDF documents.

Linked Document mode stores a reference to the original file. It keeps the .upmx file smaller, but the link can become unavailable if the source is moved, renamed or deleted. UPMXS supports relinking.

Include File in UPMXS stores a portable copy so the source remains available from project memory even if the original file is unavailable.

Current PDF text extraction is intended for text-based PDFs. Scanned or image-only PDFs require future OCR support.

10. Images and Visual Assets

Use File > Import Image to add supported visual assets such as logos, illustrations, blueprints and reference images. Current image formats include PNG, JPG, JPEG and WEBP.

Images can be linked to their source or included in UPMXS for portability. Visual assets can carry roles, including Custom roles, and approval information.

UPMXS preserves the original image data for portable images. The normal inspector preview uses high-quality display scaling.

Large image viewer: Double-click an available image preview to open a larger viewer. The viewer supports Fit, 100% view, zoom controls and scrolling so large diagrams and blueprints can be inspected inside UPMXS.

Save Original Image As: For an image included inside the .upmx project, select Save Original Image As... to extract the original image to the local drive. UPMXS writes the original embedded image bytes without resizing, re-encoding or changing DPI.

11. Search, Context, Dashboard and Suggestions

Use File > Project Knowledge Search to locate stored project knowledge.

Use File > Relevant Project Context to build a focused context package from Continuation Context and relevant stored knowledge.

Use File > Project Dashboard for current goal, state, task, next step, memory counts, knowledge breakdown, project health and supported Project Progress.

Use File > Knowledge Suggestions to review possible memory gaps. Suggestions are advisory and do not automatically modify authoritative project memory.

12. Project Progress

Project Progress is optional. UPMXS preserves progress when the source actually supports it and does not invent an overall percentage merely because individual stages or tasks exist.

A stage can exist without a numeric percentage. Status values can describe states such as Complete, Working, Current Stage, Not Started, Blocked or Verified when supported.

13. Saving, Portability and Recovery

Use File > Save regularly. A .upmx file carries the project's structured memory and can be copied to another location and reopened.

UPMXS includes safe-save and recovery behavior using backup (.bak) and interrupted-save temporary (.tmp) files. Unsupported or invalid project files are rejected rather than silently accepted.

Provider credentials are not stored inside .upmx project memory.

14. License and Activation

UPMXS Personal includes an 11-day full trial. After the trial, the project remains accessible in read-only mode so project files and project memory remain available.

A UPMXS Personal license is a \$29 one-time license. Use Help > License / Activation to review status, enter a license key or select Buy UPMXS Personal to open the purchase website.

Activation is device-bound. A license already activated on another device may require the supported device-reset/replacement process before activation on a different device.

15. Troubleshooting

AI provider requests credentials: configure the provider credentials for the current UPMXS session, or use the no-direct-API Project Knowledge Prompt workflow.

AI Exchange JSON will not load: check the validation panel. Malformed JSON is rejected, and valid JSON can still contain blocking UPMXS validation errors.

Linked document or image is missing: use the relevant inspector/relink workflow to point UPMXS to the source file's new location. Included portable content remains available from the project.

Need to inspect a large portable image: double-click the preview to open the large viewer, then use Fit, 100% or the zoom control.

Need the original portable image outside UPMXS: use Save Original Image As... to restore the original embedded file to the local drive.

Project will not open: read the UPMXS error message. Recovery may be offered when a valid .bak or .tmp file exists.

Knowledge Graph item is isolated: the project may simply contain no supported relationship for that item. UPMXS does not invent relationships merely to make the graph connected.

16. Recommended Everyday Workflow

1. Create or open the project.
2. Update Continuation Context so the current state and next action are clear.
3. Add or import useful project knowledge, documents and visual assets.
4. Use Search, Dashboard, Knowledge Graph and Knowledge Suggestions to understand the project.
5. Work with AI directly when credentials are available, or use Project Knowledge Prompt with an external AI.
 6. Download the AI-produced UPMXS Exchange .json file when the AI supports downloadable files.
7. Import the JSON and review validation, updates and conflicts.
8. Accept, Reject or Defer reviewed changes as appropriate.
9. Use Memory History when you need to understand or restore earlier project-memory values.
10. For portable visual assets, inspect large images inside UPMXS or save the original image to the local drive when needed.

UPMXS - Universal Project Memory Exchange System

UPMXS - Universal Project Memory Exchange System

Version 1.0

Powered by PME - Project Memory Engine

Created by Dimensional Loop

Copyright © 2026 Dimensional Loop