AI data transfer

Window 2.1.1

The **EXPORT | IMPORT** switches and the selected action button stay above the scrolling page. Switching opens the beginning of the selected section; each mode keeps its own selections and filters. Sources, filters, chats, other data and run options have separate nested blocks. Application, project and history occupy three rows without an unrelated column divider.

Chats have a dedicated wide search row for title, project or ID, a clear button and their own scrolling area. The list stays expanded; searching preserves hidden selections and explicitly reports no matches. Import recovery and Claude memory merge are shown only in **IMPORT**.

The **EXPORT & IMPORT / AI** page in 2.1.0 discovers local Claude Code, Claude Desktop, Codex and Grok CLI sessions, plus settings, memory, instructions, rules, skills, plugins and automation.

Claude Code is discovered independently of Desktop. Desktop chat-list entries are linked to their Code transcripts instead of duplicated. Codex includes archived sessions and compressed `.jsonl.zst` logs. Grok sessions without their replay stream are marked incomplete and can be archived, but cannot be restored through the verified adapter. ChatGPT Desktop detection is shown separately; cloud chat history is not extracted from browser caches.

Missing data, partial access and unsupported formats have explicit states. Refresh works for empty lists and failures. Option providers share a local catalog cache under `._runtime/ai_catalog` for up to 20 seconds; an explicit refresh bypasses it.

Export

Select chats independently of other data. No chat is selected initially. Application, project, archive and text filters preserve hidden selections; ALL and NONE affect visible rows only. Essential and All supported data presets change categories without changing chat or authentication selections. Essential leaves out the plugins of Codex: Codex installs them itself, and they come to hundreds of megabytes.

Authentication is off by default. Installation identifiers, process locks and shared conversation databases are excluded. Cache transfer is restricted to supported files without conversation data, such as the Codex model catalog.

Preview export first. The terminal names a full JSON plan in the operation report. Disable preview to publish `ai_backup` under the selected TARGET. Changed selected files invalidate the preview. Schema 3 records file hashes, ownership and selected chat dependencies. Publication is atomic and retains the previous backup on failure. SQLite snapshots do not create sidecar files in the source profile.

Import

SOURCE can be the backup itself or its parent containing `ai_backup`. Import selections are independent of export selections. Integrity is checked before writing. Schema 1/2 backups are supported with limitations; a manifestless backup requires explicit permission and cannot have its original integrity verified.

Preview is on by default. Review ADD, SAME, CONFLICT and REPLACE in the report. Local conflicts are preserved unless replacement is explicitly enabled. A JSON project mapping changes structured working-directory metadata and session directories, never message text. Other absolute paths in configuration require explicit permission.

Close affected applications before applying the import. The program does not terminate them. Changing the backup, selection or destination invalidates a saved plan. Only selected data is restored; unrelated local sessions and files remain.

Registration and conversation reading are verified using the official app-server of **Codex CLI 0.163.0**. Other versions are blocked until their adapter is verified. **Grok CLI 1.0.62** validates its native Markdown replay in a separate temporary profile. These checks do not request model inference or separately certify a particular Desktop client's sidebar. Claude Code files receive local format checks; tests do not start a Claude conversation.

Recovery

Before applying changes, the program saves recovery files and hashes under `backup/ai_import_<id>/journal.json`. Failures trigger automatic rollback, including app-server metadata changes.

The Import recovery card accepts a journal from this program's backup directory. Preview first, then disable preview to restore the preceding state. Files changed since the import are conflicts by default. Recovery copies are verified before use. The journal also supports recovering an interrupted import.

Compatibility

`devops_cli.cmd run --operation ai_backup_inventory` prints a source summary; `--result <file>` saves the full JSON result. The modern form explicitly submits `transfer_version: 3` and fields named `ai_export__...` / `ai_import__...`; use `devops_cli.cmd describe` for their exact definitions. Existing calls without that marker and existing command wrappers keep their previous behavior.

Memory merge remains specific to Claude Markdown memory. Codex semantic memory merging is not implemented.

Adapter references: Codex app-server, Grok CLI, Grok replay reader.