SPERT® Story Map: Connect AI

User Guide

Pair a compatible AI assistant with your story map to build and edit structure in real time · storymap.spertsuite.com

WHAT IS CONNECT AI?

Connect AI is a feature in SPERT® Story Map that lets a compatible AI assistant build or edit your story map in real time. When the connection is active, the AI can build an entire map structure — themes, backbones, and rib items — from a single description, or create them one at a time; organize work into releases, size unsized rib items on explicit request, re-plan by reassigning rib items between releases, move rib items to a different backbone (or reorganize many at once), describe themes and backbones, and enrich rib cards with descriptions, notes, and acceptance criteria, adding to notes without overwriting what you wrote — all changes appear on your map as the AI works. You stay in control at every step: the AI asks what you want to build before doing anything, and you choose exactly what it can access.

Connect AI works through the **MCP protocol** (Model Context Protocol) — a standard for giving AI assistants access to external tools. SPERT Story Map shares a single MCP server with the rest of the SPERT Suite, at mcp.spertsuite.com, that the AI calls to read and write your map. The AI assistant must be configured to reach this server before the feature can be used — the same one-time setup works for every SPERT app.

HOW IT WORKS — EACH SESSION

#	Action	Detail
1	Open a product and click Connect AI	The Connect AI button is at the right of the product's header, beside the settings gear. The first time, and after you disconnect or sign out, a window titled **Connect an AI assistant** asks for your consent. No sign-in is needed: Connect AI also works when your products are kept in local storage.
2	Choose permissions	**Write Mode** (required, already ticked) — the AI can create themes, backbones, rib items and releases; rename themes, backbones and rib items; write their descriptions, and rib categories and notes; and size, allocate, unassign and move rib items. **Read Mode** (optional, unticked to start) — the AI can see your current map before making changes: its structure, descriptions, notes, release assignments and sizes. Read Mode is required when asking the AI to allocate, unassign, size or move existing rib items, update rib text fields or backbone descriptions in bulk, or re-plan an existing map. Click **Connect**.
3	Copy the pairing code and prompt	A session code (like CRANE · 0384216791) appears with two buttons: **Copy Code** (the code alone) and **Copy Prompt** (a ready-to-paste message with the code and detailed instructions for the AI). The code is valid for 15 minutes and the panel replaces it automatically, so always copy the one on screen.
4	Paste the prompt into your AI assistant	Open your AI assistant — it must already be configured with the SPERT MCP server (see One-Time Setup). Paste the copied prompt and send it. The assistant will call the MCP server to claim the session, then ask you what product you are planning and which story map structure fits best before building anything.
5	Review and disconnect	Themes, backbones, rib items, descriptions, notes, release assignments, and sizes appear on your map in real time. To undo, open the Map or Sizing tab and press Ctrl+Z (Cmd+Z on a Mac); undo keeps your last 30 steps, and a bulk call is one step. Switch **Read Mode** at any time with the **Turn on Read** / **Turn off Read** button in the Connect AI panel. Click **Disconnect** when you are done. Signing out ends the session too; otherwise it ends by itself 7 days after it was last used, by you or by the AI. Closing the panel (×) doesn't end it: the header button then reads **AI**, and clicking it reopens the panel.

Note: Stay on the product while the AI works. Rib items with sprint progress on their release are locked: no tab and no AI can change their size, and the AI cannot change their release. To change a locked rib's release, drag it yourself on the Map or the Release Planning board — its progress moves with it. A locked rib **can** still be moved to a different backbone by the AI — that part of a move never touches progress data.

ONE-TIME SETUP: CONNECT YOUR AI CLIENT

Connect AI requires an AI client that supports MCP servers. Complete this setup once; every subsequent session — in Story Map or any other SPERT app — starts at step 1 above. **For large maps or bulk changes, use ChatGPT or Claude** — they handle big batch operations in a single request. **For large enrichment tasks** (adding descriptions, notes, and acceptance criteria to many cards at once), **Claude is the most reliable choice**. Any connected assistant works well for small, targeted edits.

Claude — web and desktop app (any plan, including Free)	Lowest-friction path for non-developers — no terminal or config file required. The Free plan allows one custom connector. Go to **Customize** → **Connectors**, click **+ Add**, then **Add custom connector**. Name it **SPERT Suite** and paste https://mcp.spertsuite.com as the URL. Click **Continue** through the next screens, choosing **No sign in** for Authentication, and finish with **Add**. On a Team or Enterprise plan, an Owner adds the connector first: **Organization settings** → **Connectors** → **Add** → **Custom** → **Web**, then the same name, URL and No sign in. Each member then opens Customize → Connectors, finds SPERT Suite (marked Custom) and clicks Connect. A connector added on claude.ai also appears in the Claude desktop app automatically. There is no second setup, no Node.js and no config file. You can also add it from inside the desktop app, the same way. To use it: open a new conversation, click **+** (lower left) → **Connectors**, and make sure SPERT Suite is switched on, open SPERT Story Map in your browser, click Connect AI, and paste the Copy Prompt.

Claude Code (Pro, Max, Ultra, Team, Enterprise or Console account — not the Free plan)	Best for developers — one command. Run this once, on one line:
	`claude mcp add --transport http --scope user spert-suite https://mcp.spertsuite.com`
	`--scope user` makes it available in every folder. Without it, it is added only for the folder you run the command in. Confirm with:
	`claude mcp list`
	spert-suite should be listed as Connected. If you already added the connector on claude.ai and sign in to Claude Code with that account, it is there already — check with:
	`/mcp`
	To use it: start a Claude Code session, open SPERT Story Map, click Connect AI, and paste the Copy Prompt.

Grok	Go to grok.com/connectors → **New Connector** → **Custom**, paste https://mcp.spertsuite.com and finish — no sign-in is needed. On Grok Business or Enterprise, a team admin adds the connector in xAI's cloud console first. To use it: open a Grok conversation, open SPERT Story Map in your browser, click Connect AI, and paste the Copy Prompt.

ChatGPT (Plus / Pro / Business / Enterprise / Edu — web only)	On Plus or Pro, you add it yourself. On Business, a workspace admin or owner adds it once — on Enterprise and Edu, so can a member they authorise. Go to **Plugins** → **+** → **Add custom MCP server**. Name it **SPERT Suite**, paste https://mcp.spertsuite.com under Connection, choose **No authentication**, tick **I understand and want to continue**, and click **Create as a plugin**. To use it: install the SPERT Suite plugin, start a new chat, type @ and pick SPERT Suite, then paste the Copy Prompt. Pick SPERT Suite with @ again in each later message that asks it to change something — ChatGPT uses a plugin only in the message where you pick it. ChatGPT may ask you to confirm changes. **Recommended for large maps, bulk release planning, and allocation** — confirmed to build a 50-story map with releases and full allocation in a single request. **Note:** For large enrichment tasks (many descriptions and notes), Claude is more reliable — ChatGPT may pause for confirmation between steps. If ChatGPT shows fewer tools than expected or behaves unexpectedly, check that you picked SPERT Suite with @ in that message.
Copilot Studio (Microsoft 365, advanced)	In a Copilot Studio agent, open **Tools** → **Add a tool** → **New tool** → **Model Context Protocol**. Fill in the three required fields: **Server name** (SPERT Suite), **Server description** (for example: Builds and edits SPERT Story Map products) and **Server URL** (https://mcp.spertsuite.com). Set Authentication to **None** and click Create, then **Create a new connection** → **Add to agent**. Test it in the agent's test panel. It needs a Copilot Studio licence or the free Copilot Studio trial — sign up with a work or school email; a trial agent can be tested but not published. The Copilot Studio features bundled with ordinary Microsoft 365 plans don't include the connectors that MCP tools run on. **Note on Microsoft 365 Copilot Chat (the standard chat experience):** Copilot Chat connects and works well for small, targeted edits — including moving one rib item at a time — but struggles with bulk changes on large maps — it may stall partway through or return an error on batch operations. For a big reorganization (many releases, many allocations, or redistributing many rib items across backbones), use ChatGPT or Claude instead. Copilot Studio (this row) is a stronger orchestrator than Copilot Chat and handles larger operations; it requires additional admin configuration to deploy to Copilot Chat.

Choosing the right assistant for the job: Claude (claude.ai, Claude Code, Claude Desktop) is the most reliable choice for large enrichment tasks — adding descriptions, notes, and acceptance criteria to many cards at once. Confirmed: 100-rib map enriched in one pass with Claude Code. **ChatGPT and Claude** both handle bulk structural changes well — creating many releases, allocating dozens of stories, redistributing many rib items across backbones, or re-planning a large map in a single request. **Microsoft 365 Copilot Chat** is great for quick, targeted edits (add a rib, rename a theme, move a single rib item) but may stall or return an error on large batch operations. For a big reorganization with Copilot, switch to ChatGPT or Claude. **Grok and Copilot Studio** connect and work; bulk performance on large maps is best confirmed with a short test run on your own account.

WHAT STILL DOESN'T WORK

Connect AI requires an AI client with MCP server support configured. If you paste the Copy Prompt into an AI that has no MCP connection, the assistant reads the tool-call instructions but has no actual channel to your story map — nothing changes in the app. In some cases the AI may even fabricate a response, describing a map it claims to have built. The tell: tool names like **resolve_session_code** appear in the reply, but your map stays empty.

Not supported today: ChatGPT Free and Go. Any other AI assistant without MCP support configured.

Not yet tested with SPERT: Gemini web app (US, personal Google accounts): Google now lets you add a custom MCP app under Settings → Connected Apps → Custom apps → Add a custom app. **Antigravity CLI** (Google's replacement for Gemini CLI): add SPERT to `~/.gemini/config/mcp_config.json`:

```
{ "mcpServers": { "spert-suite": { "serverUrl": "https://mcp.spertsuite.com" } } }
```

WHAT'S COMING

MCP adoption has moved quickly. Claude (on the web, in the desktop app and in Claude Code), Grok, ChatGPT and Copilot Studio work today.

As MCP adoption continues to expand, Connect AI will work with new clients automatically — no changes to the SPERT server or your workflow will be required. The same pairing-code workflow works across all supported clients today, and across every SPERT app that offers Connect AI.

WRITE MODE, READ MODE & PRIVACY

Write Mode (required)	Lets the AI create themes, backbones, and rib items, organize work into releases, size unsized rib items on explicit request, re-plan by unassigning and reallocating rib items between releases, move rib items to a different backbone (optionally reassigning the release in the same step), write theme and backbone descriptions, update rib descriptions, categories, and notes in bulk, and add to a rib's notes without overwriting them. Eight bulk tools collapse N individual calls into one for an existing map — critical for large maps and AI clients that yield between tool calls: **storymap_bulk_create_releases**, **storymap_bulk_allocate**, **storymap_bulk_size**, **storymap_bulk_unassign**, **storymap_bulk_update_ribs**, **storymap_bulk_move_ribs**, **storymap_bulk_append_rib_notes**, and **storymap_bulk_update_backbones**. A ninth, **storymap_bulk_import**, is a fast path for the AI to build a brand-new map structure in one call instead of one theme/backbone/rib at a time (22 Story Map tools, plus 2 session tools shared by every SPERT app that offers Connect AI). The AI cannot delete items or record progress; when it moves a rib to another release, the rib's progress moves with it, as a drag does. Rib items with sprint progress cannot be resized by the AI or on any tab, and the AI cannot reassign their release — drag such a rib yourself to change its release. Rib items with sprint progress **can** still have their backbone changed, and their descriptions, categories, and notes updated, by the AI.
Read Mode (optional)	Shares a compact snapshot of your current map with the AI — structure, release assignments, current rib sizes, size mapping, theme, backbone and rib descriptions, and rib notes. No progress data, allocation percentages, or assessment notes are included. On a very large map the snapshot leaves out rib notes first, then theme and backbone descriptions, to stay within its size limit, and tells the AI it has done so. Read Mode uploads a copy of your open product to the Connect AI relay even if you keep your products in local storage; turning Read Mode off, or disconnecting, deletes it. The snapshot does flag whether a rib's release allocation is a partial percentage, so the AI can warn you before a move silently skips reassigning it. Read Mode is required when asking the AI to allocate or unassign ribs, size existing ribs, update rib text fields or backbone descriptions in bulk, move rib items between backbones, or re-plan release assignments, and the AI needs it to see a description or note before changing one; leave it off when building from scratch. Switch it at any time with the **Turn on Read** / **Turn off Read** button in the Connect AI panel. **Note on rib notes:** the AI reads rib notes in the snapshot, so it can check what it wrote; if the snapshot leaves them out, open the rib's panel on the Map to check.
Session data & expiry	Pairing codes are single-use and expire after 15 minutes. Session data (the ops queue and any Read Mode snapshot) is stored in Firebase and deleted seven days after the session was last used, by you or by the AI. Turning Read Mode off deletes the snapshot. Click Disconnect to end a session and delete it from the relay immediately; closing the panel doesn't end it. Signing out ends the session too; if Story Map can't reach the relay at that moment, it tells you and keeps trying. Full details at spertsuite.com/ai-privacy.

AI-built map limits	One storymap_bulk_import call builds up to 5 themes, 10 backbones per theme, and 10 rib items per backbone — up to 500 rib items — and carries at most 800,000 bytes of text; a larger map takes more than one call, or the one-at-a-time tools, which have no size cap. These limits apply only to the AI, not to the app. Bulk operations handle up to 500 rib items per call (100 when appending notes or writing backbone descriptions) and up to 50 releases per call. Existing maps and all other editing tools are otherwise unchanged.
Tips & caveats	Enable Read Mode before asking the AI to edit an existing map, allocate work to releases, unassign or reassign rib items, move rib items between backbones, update rib text fields in bulk, or size ribs — without it, the AI cannot see your current structure and may duplicate work. Undo (Ctrl+Z, or Cmd+Z on a Mac, on the Map or Sizing tab) works on AI changes too, back to your last 30 steps. Review the AI's output before saving a JSON backup, as undo history clears when you reload the page or leave the product. Don't switch products mid-session (see the Note under How It Works). The AI never deletes themes, backbones, rib items, or releases — only you can delete. When you ask the AI to add to a rib's notes, it adds after what is there; it replaces notes or a description only when you ask it to. Moving rib items between backbones may reorder them within their release's column on the Release Planning board, even when the release itself didn't change — a cosmetic side effect, not a data change.

ENRICHING RIB CARDS WITH THE AI

Once your story map structure exists — themes, backbones, and rib items — you can ask the AI to enrich the rib cards with descriptions, acceptance criteria, and implementation notes. Descriptions and categories go through **storymap_bulk_update_ribs**, which updates up to 500 rib cards in a single call; notes are added with **storymap_bulk_append_rib_notes**, up to 100 rib cards a call, without overwriting what is there. The AI can also describe themes and backbones (**storymap_bulk_update_backbones** covers up to 100 backbones a call). Enable **Read Mode** first so the AI can read your map, including the notes and descriptions already there.

Example prompts that work well:

Add descriptions	"Read Mode is on. For each rib item in the map, write a 1–2 sentence description that explains what the feature does and who it benefits."
Add acceptance criteria	"For each rib item, add acceptance criteria in the Notes field. Use a simple Given/When/Then format. Keep each criterion to 2–3 lines."
Enrich a specific backbone	"Look at the rib items under the 'User Authentication' backbone and add detailed notes covering edge cases, dependencies, and any security considerations."
Describe backbones	"Write a one-sentence description for each backbone that doesn't have one yet, saying what it covers."
Clear all notes	"Remove the Notes content from every rib item in the map." (The AI will set each notes field to empty — you can undo it with Ctrl+Z on the Map or Sizing tab.)

Note on rib notes: say "add" to keep what a rib's notes already hold, and "replace" or "clear" only when you mean it — the AI adds unless you ask otherwise. Notes hold up to 2,000 characters a rib; an addition that would go past that is skipped for that rib, not cut short. Descriptions, categories and notes are all in the snapshot returned by **storymap_get_product**, so you can ask the AI to read the map again to check its work. On a very large map the snapshot may leave notes out; then open the rib's panel on the Map to check them.

MOVING RIB ITEMS WITH THE AI

Once your story map has backbones and releases in place, you can ask the AI to move rib items — relocate a card to a different backbone, reassign its release, or both in the same request. This uses **storymap_move_rib** for a single card or **storymap_bulk_move_ribs** to redistribute many cards in one call — for example, splitting every rib item under one backbone across several new ones. Enable **Read Mode** first so the AI can see your current structure.

Example prompts that work well:

Move one rib to a new backbone	"Move the 'Reset password' rib item to the 'Account Recovery' backbone."
Redistribute many ribs at once	"Look at every rib item under the 'Legacy Checkout' backbone. Decide which of the new backbones — 'Cart', 'Payment', or 'Fulfillment' — each one fits best, and move it there."
Move and reassign the release together	"Move the 'Add to cart' rib item to the 'Cart' backbone and allocate it to the 'v2.1' release."
Reorganize the whole map	"Create backbones that reflect the actual feature areas in this product, then move every rib item into the backbone it best fits." (Best with ChatGPT or Claude on a large map — see Choosing the right assistant, above.)

Locked rib items: a rib item with recorded sprint progress can still be moved to a different backbone by the AI. Only the release-reassignment part of a move is blocked on a locked rib — to change its release, drag it yourself on the Map or the Release Planning board; its progress moves with it.

Split or partial release allocations: if a rib item's release doesn't change after asking the AI to move it, check whether that rib was split across multiple releases or partially allocated (under 100%) to one. The AI skips the release change on these rib items on purpose, so it never silently overwrites a percentage you set by hand. To replace a split or partial allocation, ask the AI to unassign the rib item first, then move it — an unassigned rib is always eligible for a new release.

Good to know: the target backbone for a move can be in any theme, not just the rib's current one — useful if you keep more than one theme on a map. Moving rib items between backbones may also reorder them within their (unchanged) release's column on the Release Planning board — a cosmetic side effect of the move, not a bug. For a large reorganization, prefer asking for it in one request (**storymap_bulk_move_ribs**) over moving items one at a time — it completes in a single step and is more reliable on AI clients that pause between tool calls, such as Microsoft 365 Copilot Chat.

TROUBLESHOOTING

Symptom	Likely cause	Fix
AI shows fewer tools than expected (e.g. no storymap_get_product, or missing bulk tools)	Stale cached manifest. AI clients cache the tool list at connect time.	Fully remove the MCP connector and re-add it. A soft Refresh is often insufficient. The connector lists every tool the shared suite server advertises across all SPERT apps — 56 today, of which 22 are the storymap_ tools this guide covers. The suite total grows as other SPERT apps add tools, so the check that stays reliable is the second number: after re-adding, confirm the full storymap_ family, storymap_get_product included, is present before starting a session. On ChatGPT Business, Enterprise or Edu, an admin or owner refreshes the tool list from the workspace's app settings instead.

ChatGPT behaves unexpectedly: loops, does not follow instructions, or reports few tools	SPERT Suite wasn't picked with @ in that message.	Pick SPERT Suite with @ in the message, or start a new conversation and pick it again.
Tool names appear in the AI's reply but nothing changes in the app	The AI has no live MCP connection — it is reading the prompt but cannot reach the server.	The AI assistant must be configured with the SPERT MCP server (see One-Time Setup). Paste the Copy Prompt only after the connector is set up. Tool names in a reply without any actual changes is the classic sign of a missing connection.
appVersion shows an old version number after a deploy	Browser tab is using a cached JS bundle from before the deploy.	Click Disconnect, reload the page, then press Connect AI and give the AI the new code. A session you resume keeps the version it started with.
The AI says rib notes were left out of what it read (notesIncluded: false)	The map is very large, so the snapshot drops notes to stay within its size limit.	Nothing is removed from the product. Open the rib's panel on the Map to check notes, and ask the AI to add to notes rather than replace them, since it cannot see what it would replace.
Rib item cannot be resized by AI	Rib is locked — progress above 0% is recorded for it on a release it is on.	Locked ribs cannot be resized by the AI or on any tab. Use **Clear progress** on the rib's expanded row on the Progress tab, which deletes its records, or drag it to Unassigned, change the size and drag it back, which keeps them. Note: descriptions, categories, and notes can still be updated on locked ribs, and a locked rib can still be moved to a different backbone.
Moving a rib item doesn't change its release (backbone move still works)	The rib is locked, or its release allocation is a split or under 100% (partial).	The AI intentionally skips the release change on a split or partially-allocated rib item, so it never overwrites a percentage you set by hand. Ask the AI to unassign the rib item first, then move it again — an unassigned rib is eligible for the new release. A locked rib can't be unassigned by the AI either: drag it to the new release yourself.
Rib items appear in a different order on the Release Planning board after an AI move	Moving a rib between backbones repositions it within its release's column as a side effect.	This is a cosmetic side effect of the move — no release or progress data changed. Drag the card back into place on the Release Planning board if you'd like a different order.
Ctrl+Z does nothing after an AI change	You are not on the Map or Sizing tab, or you are on a Mac.	Open the Map or Sizing tab and press Ctrl+Z (Cmd+Z on a Mac). Undo keeps your last 30 steps.
The AI says a change went through, but nothing changed in the app	Story Map skips, without an error, a change it cannot apply: the item, release or size label does not exist, the rib is locked, or this tab is older than the tool the AI used.	Ask the AI to read the product again (storymap_get_product) and check. If the tab was open before the latest Story Map update, reload it, press Connect AI, and ask the AI to send the change again.

SPERT® Story Map · Connect AI Guide · © 2026 William W. Davis, MSPM, PMP · Licensed under GNU GPL v3 storymap.spertsuite.com