

PULSE / Navisworks

Guide and commands

Autodesk Navisworks Manage 2023-2027

Installation, first steps and natural-language examples. Technical command names remain unchanged.

Verified source: <https://github.com/LuDattilo/pulse-for-navisworks>

Branch: main | Commit: f55379709be7bc94a58104dbfa227d4db0c4323f

2026-10-04 | EN

Descriptions in English. Command names, parameters and technical values are unchanged. Source catalogue dated 4 October 2026; translations updated on 8 October 2026.

<https://pulse-ai.it/guida-navisworks.html>

Installation and connection

Close Navisworks Manage, extract the package and run its EXE installer. Restart Navisworks, open the federated model and switch PULSE on. Manage is required for Clash Detective.

Connect an MCP-compatible AI client: Claude, Codex, opencode or another client, including local models.

Open the project and describe the goal. Check the result and approve changes when requested.

Work on a project copy. Request a read operation or preview before changes, check the selection and approve only the intended operations.

On the first eligible registration, a personal 14-day Premium Trial. Another request does not renew an existing trial.

Example requests

In Navisworks Manage, open the NWF/NWD file and start Pulse from its tab. The dedicated MCP server is pulse-navisworks. After changing the port in settings, restart Navisworks before checking the connection.

Revit and Navisworks can run together as separate products and processes. Check both connections before transferring a selection between applications.

Clash review: list tests, run the selected test, filter results, isolate involved components and request a preview of status updates. Export the report or BCF after checking the result.

TimeLiner: inspect tasks, identify the correct task and attach the selection only after checking it. Export the schedule as CSV for sharing.

If the Pulse tab is missing, close Navisworks, use the official installer for the supported version and ask IT to check possible blocks. Do not manually mix DLLs from different versions.

Command catalog

Descriptions in English. Command names, parameters and technical values are unchanged. Source catalogue dated 4 October 2026; translations updated on 8 October 2026.

Clash

`create_clash_test`

Create a clash test with two selection sides. `testType`: Hard/HardConservative/Clearance/Duplicate/Custom. `tolerance` in metres. `runImmediately` runs it right after create.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\ClashTools.cs`

`create_clash_test_from_categories`

Create a clash test from two Revit category names. Auto-resolves locale/export variants (Element/Category, Elemento/Categoria, Entità/Categoria, LcRevitData_Element/LcRevitPropertyElementCategory), e.g. 'clash test between muri and pavimenti'.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\ClashTools.cs`

`delete_clash_test`

Delete a Clash Detective test and all its results, by `guid` (preferred) or `displayName`. Use `list_clash_tests` to find the identifier. Irreversible - asks for confirmation before removing.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\ClashTools.cs`

`export_bcf`

Export clash results as a BCF 2.1 archive (Solibri/BIMcollab/Tekla). Pass `testGuid/testDisplayName` for one test, or omit both for ALL tests. Each clash becomes one BCF Topic with a viewpoint at the clash center.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\ClashTools.cs`

`export_clash_report`

Export clashes to a JSON or CSV file. Defaults to JSON across all tests; scope to one via `guid/displayName`. Returns output path and exported count. Emits a Cortex-shaped report, not the native Navisworks HTML/XML report.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\ClashTools.cs`

`focus_clash`

Focus a single clash: clear prior focus overrides, unhide all, select the two items, color them red/blue, save a recallable viewpoint.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\ClashTools.cs`

`get_clashes`

Read clash results of one test (by GUID or display name). Filters: `status`, `maxResults`, `offset`.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\ClashTools.cs`

group_clash_results

Group a clash test's results into native ClashResultGroup containers (the Results tab), AFTER running it. Identify the test by testGuid (preferred) or testName. Does NOT change SelectionA/SelectionB. Returns groupsCreated, resultsMoved and per-group counts.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\ClashTools.cs

list_clash_tests

List Clash Detective tests. Each entry: type (Hard/HardConservative/Clearance/Duplicate/Custom), status (New/Old/Partial/Complete), tolerance, last-run timestamp, per-status clash counts. Returns ALL tests by default; pass a positive maxResults to page (with offset; hard cap 5000). summaryOnly=true returns only the rollup (per-status + per-testType counts) without per-test entries.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\ClashTools.cs

run_clash_test

Run a clash test (by GUID or display name), returning updated status + counts. runAll=true runs every test. After running, use get_clashes to read results.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\ClashTools.cs

set_clash_status

Mark one or more clash results with a review status - New, Active, Reviewed, Approved, or Resolved (i.e. approve, review, or resolve clashes), optionally assigning them. Identify clashes by GUID (from get_clashes). Returns a per-clash result list with the previous status.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\ClashTools.cs

set_clash_test_options

Edit options of an existing clash test (NOT selections - use update_clash_test_selection). Pass guid (preferred) or displayName, plus any of newDisplayName, testType, tolerance, mergeComposites, priority. Omitted fields preserved.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\ClashTools.cs

update_clash_test_selection

Replace items on side A and/or B of an existing test (by guid preferred, or displayName). Pass only the side(s) to change; each accepts selectionSetGuid, modelFileName, paths, instanceGuids, useCurrentSelection. Changing selections clears the test's existing clash results.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\ClashTools.cs

Interop

cross_app_selection

Symmetric Revit<->Navis selection bridge. export -> emit PulseElementRefs from CurrentSelection (default), or a single clash via clashGuid, or a whole test via clashTestGuid (Manage edition only). import -> resolve PulseElementRefs to ModellItems and select them (replaces selection unless append=true). Import with no resolved refs clears the selection only when append=false; use append=true to leave it untouched. Resolution priority: revitUniqueld -> ifcGuid -> revitElementId.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\InteropTools.cs

get_revit_references

Resolve Navisworks items to cross-app Revit references. Pass path(s) or instanceGuid(s); returns sourceFile, Revit ElementId/Uniqueld when present, IfcGUID, category, family, type.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\InteropTools.cs

publish_selection_scope

PREVIEW - the cross-app resolve-ack shape is not yet schema-locked and will change with ack-1.0 (the published envelope and payloadSha256 are stable). Publish a selection-intent to the cross-app interop channel (~/.pulse/interop/selection-scopes) as a new immutable revision (ADR-0005). Pass 'intent' (a selection-intent object with selectionId + selectionType). Returns the revision, on-disk path, payloadSha256 and folder name. Write tool: blocked in read-only mode and on the Free tier.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\InteropTools.cs

resolve_selection_scope_in_navisworks

PREVIEW - the ack this writes is not yet schema-locked and will change with ack-1.0; do not build a consumer against it yet. Resolve a published selection-scope into a Navisworks Search Set. Reads the highest revision for 'selectionId' from the interop channel and, for a rule-based intent, builds a live Search Set from its rules. Writes a resolve ack (status derived from evaluated/targets). Returns evaluated, status, searchSetGuid, liveltemCount. Write tool: blocked in read-only mode and on the Free tier.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\InteropTools.cs

Meta

flow_step

Declare which step of a registered flow you are on. Returns the flow envelope (flow, flowStep, flowStepCount) and lights the Navisworks ribbon indicator - the one layer no MCP client can decline. Call at the START of every step (including the last), before that step's work tool; then call ONCE MORE with complete:true when the flow ends. Requires a Pulse license.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\MetaTools.cs

get_project_info

Get information about the active Navisworks document: title, file, loaded models, units, selection-set and viewpoint counts.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\MetaTools.cs

`say_hello`

Test MCP connection to Pulse.Navisworks. Returns a greeting and the detected locale. Pass an optional message to echo it back.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\MetaTools.cs

Property

`find_items_by_property`

Find items whose property matches a criterion: a category matcher (e.g. `categoryDisplayName='Element'`), a property matcher (e.g. `propertyDisplayName='Category'`), and optional value (e.g. `'Doors'`). Returns matches with `instanceGuid` + optional metadata (`path`, `sourceFile`, `class`, `matched value`). Default substring + case-insensitive; opt-in `exactMatch/caseSensitive`. Set `includePath/includeSourceFile/includeClassData/includeHasGeometryInPayload=false` for ~80% smaller payloads.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\PropertyTools.cs

`get_item_properties`

Get the full `PropertyCategories` (BIM) tree of items identified by display-name path (root-to-leaf) or `InstanceGuid`. Use when you already know the item (e.g. from `get_selected_items` / `find_items_by_property`).

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\PropertyTools.cs

Selection

`clear_selection`

Clear the current selection. Idempotent (succeeds when nothing is selected). Returns `wasEmpty` + `previousCount`.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\SelectionTools.cs

`get_selected_items`

Get items currently selected in Navisworks. Each item includes: ancestor display name, path (root-to-leaf), source file, class, geometry presence, hidden state, instance GUID. Optionally the full `PropertyCategories`/BIM tree (`Family`, `Type`, `ElementId`, `Phase`, `parameters`).

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\SelectionTools.cs

`invert_selection`

Invert the current selection. Default universe is every tree node; `geometryOnly=true` restricts the universe (and result) to leaf geometry items. Replaces the current selection.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\SelectionTools.cs

`select_all`

Select every item in the active document. Default selects all tree nodes; `geometryOnly=true` restricts to leaf geometry items (faster on huge models; usual for clash/hide/color). Replaces current selection unless `append=true`.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\SelectionTools.cs

`select_items_by_path`

Select items identified by display-name path (root-to-leaf) and/or InstanceGuid. Replaces the current selection unless `append=true`. Returns selected count and a `notFound` list for unresolved items.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\SelectionTools.cs`

Selection Sets

`create_folder`

Create a folder in the Selection Sets tree ('Gruppi' panel in Italian). Folders hold sets and sub-folders. Nest under an existing folder via `parentGuid`.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\SelectionSetsTools.cs`

`create_search_set`

Create a Search Set from one or more property criteria (AND-combined); re-evaluates live. Single criterion: `categoryDisplayName + propertyDisplayName + value + op` (or `*InternalName` variants). Multi-rule: pass a 'rules' array, each element with the same fields - needed for scopes like `Category=Pipes AND Level=L02`. Optional `modelFileName` scopes the search to ONE loaded model (else all loaded models).

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\SelectionSetsTools.cs`

`create_selection_set`

Create a Selection Set. Captures current selection by default; pass paths/instanceGuids for explicit items. Returns GUID and item count. Appended to SelectionSets root.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\SelectionSetsTools.cs`

`delete_selection_set`

Delete a selection set, search set, or folder by GUID. Folders are removed recursively with children. Get GUIDs via `list_selection_sets`.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\SelectionSetsTools.cs`

`list_selection_sets`

List saved Selection Sets and Search Sets. Each entry reports kind ('selection'|'search'), folder path, name, GUID, item count. Use `kindFilter` for one kind.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\SelectionSetsTools.cs`

`rename_selection_set`

Rename a selection set, search set, or folder by GUID.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\SelectionSetsTools.cs`

Takeoff

get_takeoff_summary

Get a high-level snapshot of the document's Quantification (5D Takeoff) state: database initialization, currency symbol, available catalogs (Items, Resources, Steps, Objects). Returns whether each catalog is present without enumerating its rows. When the document has no Takeoff data, returns { available: false, reason: "..." } instead of erroring - call this first to gate any 5D workflow.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\TakeoffTools.cs

Timeliner

create_timeliner_task

Create a Timeliner (4D) task. Required: name. Optional: taskType (an EXISTING simulation task type name, e.g. Construct/Demolish/Temporary), plannedStart/plannedEnd/actualStart/actualEnd (ISO-8601 dates like 2026-03-01), progressPercent (0-100), parentIndexPath (from list_timeliner_tasks' indexPath field) or parentDisplayName to nest under an existing task; omit both for top level. When you pass parentIndexPath you MUST also pass expectedParentDisplayName - the index is positional and nests under the wrong task if the hierarchy shifted, so a mismatch is rejected. parentGuid is rejected - every Timeliner task reports the same empty GUID. Returns the new task's indexPath.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\TimelinerTools.cs

delete_timeliner_task

Delete a Timeliner (4D) task by indexPath (from list_timeliner_tasks' indexPath field, preferred) or displayName - guid is rejected, every Timeliner task reports the same empty GUID. When you pass indexPath you MUST also pass expectedDisplayName - the index is positional and a delete against a shifted index would remove the wrong task, so a mismatch is rejected. Removing a summary/parent task removes its sub-tasks too. Irreversible - asks for confirmation before removing.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\TimelinerTools.cs

export_timeliner_csv

Export the full Timeliner schedule to JSON or CSV. One row per task: planned/actual dates, durations, status, costs, and first three user columns. Default JSON; format='csv' writes a UTF-8-BOM file Excel opens correctly.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\TimelinerTools.cs

link_task_to_selection

Attach a Timeliner task to model items OR to a saved set/search, replacing the task's existing selection. selectionSetGuid links a saved selection set (snapshot) or search set (DYNAMIC - task re-evaluates the filter). Otherwise items default to the current Navisworks selection; pass path(s)/instanceGuid(s) to override. Identify the task by indexPath (from list_timeliner_tasks' indexPath field, preferred) or displayName. When you pass indexPath you MUST also pass expectedDisplayName - the index is positional and links the wrong task if the hierarchy shifted, so a mismatch is rejected. guid is rejected, every Timeliner task reports the same empty GUID.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\TimelinerTools.cs

`list_timeliner_tasks`

List TimeLiner (4D) tasks with planned/actual dates, simulation type, progress percent, cost breakdown, and linked-selection kind (explicit/search/sources). Narrow with folderFilter; includeCosts=false skips cost columns.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\TimelinerTools.cs

`set_timeliner_task`

Edit an existing TimeLiner (4D) task (does NOT touch its selection - use link_task_to_selection). Identify by indexPath (from list_timeliner_tasks' indexPath field, preferred) or displayName, plus any of: newDisplayName, plannedStart, plannedEnd, actualStart, actualEnd (ISO-8601), progressPercent (0-100), taskType (existing type name). When you pass indexPath you MUST also pass expectedDisplayName - the index is positional and edits the wrong task if the hierarchy shifted, so a mismatch is rejected. guid is rejected - every TimeLiner task reports the same empty GUID. Omitted fields preserved. Setting an actual date turns the actual track on.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\TimelinerTools.cs

User Flow

`manage_user_flows`

Create, get, list or delete user-defined Pulse flows (stored in ~/.pulse/navisworks/user-flows.json). create needs a 'flow' JSON object {id,title,description,triggers[],steps[{name,gate,tools[]}]}; get and delete need 'id'. Requires a Pulse license.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\UserFlowTools.cs

Viewpoint

`apply_viewpoint`

Apply a saved viewpoint by GUID or display name: restores the saved camera plus any visibility/appearance overrides. Default resets the current selection; preserveSelection=true snapshots and restores it across the apply.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\ViewpointTools.cs

`create_viewpoint`

Save the current view as a new SavedViewpoint, preserving the current camera and any active visibility/appearance overrides.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\ViewpointTools.cs

`delete_viewpoint`

Delete a single saved viewpoint by GUID or display name. Only leaf viewpoints - folder/group items are not deletable here.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\ViewpointTools.cs

`list_viewpoints`

List saved viewpoints in the active document. Each entry reports folder path, display name, GUID, and override flags. Narrow with `folderFilter`.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\ViewpointTools.cs`

Visualization

`clear_overrides`

Reset color/transparency overrides. `scope='selected'` (default) uses `path/InstanceGuid` or current selection; `scope='all'` clears every override. Idempotent.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\VisualizationTools.cs`

`hide_items`

Hide items by path or `InstanceGuid` (or current selection if none given). Reverse with `unhide_all`.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\VisualizationTools.cs`

`isolate_by_category`

Isolate by property in one pass: hides everything except items where the property matches value (`find_items_by_property` + `isolate_items` combined). Defaults target Revit 'Element'/'Category' (e.g. `value='Doors'`). Returns counts only. Reverse with `unhide_all`.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\VisualizationTools.cs`

`isolate_items`

Isolate items: hide everything not in the set (or current selection if none passed). Ancestors and full subtrees of seeds stay visible. Reverse with `unhide_all`.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\VisualizationTools.cs`

`override_color`

Apply a permanent color (and optional transparency) override to items. Color via r/g/b ints [0,255] or hex '#RRGGBB'. No items = current selection. Reverse with `clear_overrides`.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\VisualizationTools.cs`

`unhide_all`

Reveal all items hidden via `hide_items/isolate`. Idempotent.

Scope: | `sources\navisworks\src\Pulse.Navisworks.Server\Tools\VisualizationTools.cs`

Workflow

`workflow_clash_review`

One-call clash review on a single test: optionally rerun, optionally export to JSON/CSV, optionally color items by status (New=red, Active=orange, Reviewed=yellow, Approved=green, Resolved=gray). Returns counts per status.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\WorkflowTools.cs

`workflow_clash_to_bcf`

End-to-end clash coordination in one call: optionally rerun the clash test (or ALL tests when neither guid nor displayName is given), then export a BCF 2.1 archive for the issue tracker (BIMcollab/Solibri/ACC). Returns the rerun summary plus the BCF export result (topics, skipped). Confirms before a rerun; a pure export is not prompted.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\WorkflowTools.cs

`workflow_morning_coordination`

Read-only one-call coordination-session snapshot: model count + source formats, viewpoint/selection-set/search-set totals, and (Manage edition) clash test counts per status. Replaces `get_project_info` + `list_selection_sets` + `list_viewpoints` + `list_clash_tests` in one round-trip.

Scope: | sources\navisworks\src\Pulse.Navisworks.Server\Tools\WorkflowTools.cs

Verified source

<https://github.com/LuDattilo/pulse-for-navisworks>

main / f55379709be7bc94a58104dbfa227d4db0c4323f

<https://pulse-ai.it/guida-navisworks.html>