

PULSE / Tekla

Guide and commands

Tekla Structures 2025

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

Verified source: <https://github.com/LuDattilo/TeklaCortex>

Branch: main | Commit: 24dd5cab856e9c600038211a77648211a0f5919

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-tekla.html>

Installation and connection

The specified Tekla Structures version is required. Its EXE server connects to open Tekla through Open API and does not require an in-process Tekla plugin. Open the model and configure the AI client using the package instructions. Commercial installer and licensing are being prepared.

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.

Automatic download and serial delivery are not yet connected to this form. Contact us for this product.

Example requests

- > Summarize the selection by profile and material and calculate its total weight.
- > Find the beams in the selected phase and show their assemblies.
- > Create drawings for the selected assemblies and export them to PDF.

FEATURES

Models and assemblies

Inspect parts, assemblies, selections, phases and properties; summarize selection and weight.

Parameters and UDAs

Read and update properties, UDAs and project parameters, including element groups.

Parts and connections

Create beams, columns and plates; manage cuts, bolts, welds and available components.

Reinforcement

Inspect and create bars, groups, meshes and rebar sets through exposed operations.

Drawings and reports

Create and manage assembly, cast-unit, single-part and general-arrangement drawings; export PDFs and generate reports.

Catalogs and geometry

Query profiles, materials, measurements, base points, grids and reference models.

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.

Analysis

`create_analysis_model`

Creates a new structural analysis model with a given name, optional analysis method (FIRST_ORDER/SECOND_ORDER/P_DELTA) and construction method (FULL/BOX/SELECTED_PARTS/...).

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Analysis\CreateAnalysisModelTool.cs

`create_load_group`

Creates a load group (Tekla.Structures.Model.LoadGroup) with a name, load type and direction. The analysis-side AnalysisLoadGroup is read-only, so the physical load group (which feeds analysis models) is created instead.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Analysis\CreateLoadGroupTool.cs

`get_analysis_model_properties`

Returns detailed properties of one analysis model (selected by 'name' or 'guid'): analysis/construction method, accuracy/iteration settings, child-object counts and reported issues.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Analysis\GetAnalysisModelPropertiesTool.cs

`get_part_analysis_member`

Returns the analysis representation (analysis part, bars and members) of a physical part identified by GUID, matched via AnalysisPart.PartID across analysis models.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Analysis\GetPartAnalysisMemberTool.cs

`list_analysis_models`

Lists structural analysis models in the open Tekla model. Returns name, guid, id, analysis method and construction method per model.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Analysis\ListAnalysisModelsTool.cs

`list_load_combinations`

Lists analysis load combinations (name, type, referenced load-group count) for one analysis model (via model_name) or for all analysis models when no name is given.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Analysis\ListLoadCombinationsTool.cs

`list_load_groups`

Lists analysis load groups (name, type, direction, self-weight flag) for one analysis model (via model_name) or for all analysis models when no name is given.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Analysis\ListLoadGroupsTool.cs

Batch

`get_elements_batch`

Fetches multiple model objects by GUID in a single Tekla call. Returns the serialized elements plus the list of GUIDs that were not found.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Batch\GetElementsBatchTool.cs

`get_report_properties_batch`

Reads a matrix of report properties for many model objects: for each GUID, attempts every requested property name using the string/double/int fallback. Missing objects are listed separately.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Batch\GetReportPropertiesBatchTool.cs

Catalogs

`create_catalog_material`

Creates a new material in the Tekla material catalog (name, type, densities, modulus of elasticity in MPa, Poisson, thermal dilatation).

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Catalogs\CreateCatalogMaterialTool.cs

`delete_catalog_material`

Deletes a material from the Tekla catalog by name. Fails if the material does not exist or is in use by parts in the model.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Catalogs\DeleteCatalogMaterialTool.cs

`get_material_properties`

Reads physical properties (type, density, modulus of elasticity, Poisson, thermal dilatation) of a material from the Tekla material catalog.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Catalogs\GetMaterialPropertiesTool.cs

`get_profile_properties`

Reads section properties (area, inertia, weight per meter, height, width, type) of a profile from the Tekla profile catalog.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Catalogs\GetProfilePropertiesTool.cs

`list_catalog_bolts`

Lists bolt assemblies from the Tekla bolt catalog (name, standard, nominal diameter) with optional name substring filter.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Catalogs\ListCatalogBoltsTool.cs

`list_catalog_materials`

Lists material names from the Tekla material catalog with optional name substring and material type filters.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Catalogs\ListCatalogMaterialsTool.cs

`list_catalog_profiles`

Lists profile names from the Tekla profile catalog with optional case-insensitive 'contains' filter and an upper bound.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Catalogs\ListCatalogProfilesTool.cs

`modify_catalog_material`

Updates properties (densities, modulus of elasticity in MPa, Poisson, thermal dilatation, type) of an existing material in the Tekla material catalog.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Catalogs\ModifyCatalogMaterialTool.cs

Clash

`get_clash_check_options`

Reads the global Tekla clash check options (BoltHeadDiameter, NutThickness). Returns the current persisted values when Tekla is running, otherwise the SDK defaults.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Clash\GetClashCheckOptionsTool.cs

`run_clash_check`

Runs the Tekla clash check engine on the open model and returns the detected clashes. Optionally post-filters results to a set of GUIDs.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Clash\RunClashCheckTool.cs

Components

`get_component_inputs`

Returns the inputs (parts and points) of a component identified by GUID. Walks the ComponentInput sequence in order.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Components\GetComponentInputsTool.cs

`get_component_properties`

Reads the public properties of a component (Connection/Detail/Seam/CustomPart/Component) identified by GUID.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Components\GetComponentPropertiesTool.cs

`list_components`

Lists component objects in the open Tekla model with optional case-insensitive name/number filter.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Components\ListComponentsTool.cs

Connections

`apply_component`

Applies a Tekla Component (by name and/or number) to a primary part and optional secondary parts. At least one of name/number is required.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Connections\ApplyComponentTool.cs

`create_bolt_array`

Creates a rectangular BoltArray bolt group between a main part and a secondary part. Grid is built from dist_x/dist_y lists; defaults create a single M16 bolt.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Connections\CreateBoltArrayTool.cs

`create_bolt_circle`

Creates a circular BoltCircle bolt group between a main part and a secondary part. Bolts are evenly spaced on a circle of the given diameter; defaults create 4 M16 bolts on a 100mm circle.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Connections\CreateBoltCircleTool.cs

`create_polygon_weld`

Creates a PolygonWeld between a main part and a secondary part along a polyline (2+ points). Defaults create a 5 mm weld above the line.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Connections\CreatePolygonWeldTool.cs

`create_weld`

Creates a normal Weld between a main part and a secondary part (two GUIDs). Defaults create a 5 mm fillet weld above the line.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Connections\CreateWeldTool.cs

`get_bolt_properties`

Reads the properties of a bolt group (BoltArray/BoltCircle/BoltXYList) by GUID: size, standard, type, length, tolerance and bolt count.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Connections\GetBoltPropertiesTool.cs

`get_weld_properties`

Reads the properties of a Weld or PolygonWeld identified by GUID: sizes, types, angles, lengths, contour/finish and flags.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Connections\GetWeldPropertiesTool.cs

Cuts

`add_part_to_part`

Adds the volume of an operative part into a target part (BooleanPart BOOLEAN_ADD). Tekla deletes the operative part automatically after a successful add.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Cuts\AddPartToPartTool.cs

cut_part_with_part

Cuts a target part using another part as the cutting volume (BooleanPart BOOLEAN_CUT). The operative part is consumed by Tekla to carve a void into the target.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Cuts\CutPartWithPartTool.cs

cut_part_with_plane

Cuts a part with an infinite plane (CutPlane) defined by an origin point and a normal vector. The plane removes material on the normal's positive side; it never extends the part.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Cuts\CutPartWithPlaneTool.cs

fit_part_end

Fits the end of a beam-like part to a plane (Fitting) defined by an origin point and a normal vector. A fitting can both shorten and lengthen the part end.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Cuts\FitPartEndTool.cs

get_part_cuts

Lists the boolean modifiers attached to a part (boolean cuts/adds, cut planes, fittings, edge chamfers) via Part.GetBooleans().

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Cuts\GetPartCutsTool.cs

Drawings

add_text_to_drawing

Adds a free text note to a view of the active drawing at the given paper point. Requires a drawing open in the editor (use open_drawing first).

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Drawings\AddTextToDrawingTool.cs

close_active_drawing

Closes the drawing currently open in the Tekla drawing editor. Requires an active drawing; set 'save' to control whether changes are persisted on close.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Drawings\CloseActiveDrawingTool.cs

create_assembly_drawing

Creates an assembly drawing from an Assembly or Part GUID. If a Part is given, its owning assembly is used. Uses standard drawing attributes from the active environment.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Drawings\CreateAssemblyDrawingTool.cs

create_cast_unit_drawing

Creates a cast unit drawing from a cast unit (Assembly) or Part GUID. For a Part, its owning assembly (the cast unit) is used. Uses standard drawing attributes from the active environment.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Drawings\CreateCastUnitDrawingTool.cs

`create_ga_drawing`

Creates an empty general arrangement (GA) drawing with standard attributes. No GUID required. Optional sheet size as "WIDTHxHEIGHT" in model units (e.g. "841x594").

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Drawings\CreateGaDrawingTool.cs

`create_single_part_drawing`

Creates a single part drawing from a Part GUID. Uses standard drawing attributes from the active environment.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Drawings\CreateSinglePartDrawingTool.cs

`export_drawing_pdf`

Exports a drawing to a PDF file via `DrawingHandler.PrintDrawing`. Targets the drawing with the given GUID, or the active drawing when omitted. The drawing cannot remain active while printing, so if the target is the active drawing it is saved and closed first.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Drawings\ExportDrawingPdfTool.cs

`freeze_drawing`

Marks a drawing as frozen (excluded from automatic numbering/update). Defaults to the active drawing; pass 'guid' to target a specific one.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Drawings\FreezeDrawingTool.cs

`get_active_drawing`

Returns the drawing currently open in the Tekla drawing editor, or null when no drawing is open.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Drawings\GetActiveDrawingTool.cs

`get_drawing_objects`

Enumerates all objects on the active drawing's sheet (marks, dimensions, texts, etc.) with counts grouped by type. Requires a drawing open in the editor (use `open_drawing` first).

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Drawings\GetDrawingObjectsTool.cs

`list_drawing_views`

Lists the views placed on the active drawing's sheet (name, type, scale, origin). Requires a drawing open in the editor (use `open_drawing` first).

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Drawings\ListDrawingViewsTool.cs

`list_drawings`

Enumerates drawings stored in the current Tekla model. Returns title, number, type and revision per drawing.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Drawings\ListDrawingsTool.cs

`open_drawing`

Opens the drawing with the given GUID in the Tekla drawing editor and makes it the active drawing.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Drawings\OpenDrawingTool.cs

save_active_drawing

Saves the drawing currently open in the Tekla drawing editor. Requires an active drawing.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Drawings\SaveActiveDrawingTool.cs

set_drawing_titles

Sets Title1/Title2/Title3 of a drawing (active by default, or pass 'guid'). Omitted fields keep their current value.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Drawings\SetDrawingTitlesTool.cs

unfreeze_drawing

Clears the frozen flag on a drawing so it can be updated/renumbered again. Defaults to the active drawing.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Drawings\UnfreezeDrawingTool.cs

update_drawing

Regenerates the content of the drawing with the given GUID (same as 'Update' in the drawing list). The drawing must not be currently active and numbering must have been run.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Drawings\UpdateDrawingTool.cs

Elements

create_beam

Creates a straight Tekla beam between two points. Defaults create a small clearly named test beam.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Elements\CreateBeamTool.cs

create_column

Creates a vertical Tekla column using a start point and height. Implemented as a Beam with vertical endpoints.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Elements\CreateColumnTool.cs

create_contour_plate

Creates a Tekla contour plate from a polygon of points with profile thickness. Use for walls, slabs, and custom plates.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Elements\CreateContourPlateTool.cs

create_pad_footing

Creates a rectangular concrete pad footing at a given center point with specified width, length, and thickness.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Elements\CreatePadFootingTool.cs

create_polybeam

Creates a Tekla PolyBeam following a polyline of 2+ points. Profile, material and class follow the same conventions as create_beam.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Elements\CreatePolybeamTool.cs

delete_element

Deletes a model object by GUID. Destructive: use only when the object is confirmed.

Scope: destructive | sources\tekla-official\src\TeklaCortex.Tools\Elements\DeleteElementTool.cs

find_elements

Finds model parts by optional type/name/profile/material/class filters without changing UI selection.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Elements\FindElementsTool.cs

get_assembly_for_part

Returns the Assembly that owns a given Part (GUID, ID and main part GUID).

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Elements\GetAssemblyForPartTool.cs

get_assembly_main_part

Returns the main part of an assembly (by Assembly GUID), with profile, material, name and class.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Elements\GetAssemblyMainPartTool.cs

get_assembly_tree

Returns the main part and secondaries of an assembly. Input may be either a Part GUID or an Assembly GUID.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Elements\GetAssemblyTreeTool.cs

get_numbering_status

Returns the numbering status of a Part: main-part flags, configured prefix/start, and currently assigned assembly/part positions.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Elements\GetNumberingStatusTool.cs

list_assemblies

Lists assemblies in the active model with main part profile/material and secondary child count.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Elements\ListAssembliesTool.cs

set_part_numbering

Sets assembly/part numbering prefix and start number on a Part. Triggers Part.Modify() and optionally commits.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Elements\SetPartNumberingTool.cs

update_part_properties

Updates basic Part properties by GUID: name, profile, material and class.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Elements\UpdatePartPropertiesTool.cs

Environment

`get_advanced_option`

Reads a Tekla advanced option (XS_*) by key and type. Type must be 'string', 'double' or 'int' and must match how the option is stored. Returns NotFound if the option does not exist.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Environment\GetAdvancedOptionTool.cs

`get_environment_variable`

Reads a single Tekla environment variable or advanced option (typically XS_*). Falls back from TeklaStructuresVariables to TeklaStructuresSettings and System.Environment.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Environment\GetEnvironmentVariableTool.cs

`get_module_info`

Returns Tekla Structures program version, build, configuration, role, environment, language, and (if available) module configuration and loaded modules.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Environment\GetModuleInfoTool.cs

`get_tekla_paths`

Returns the well-known Tekla attribute search directories (firm/project/system/model/current). Tries TeklaStructuresFiles first, falls back to XS_* environment variables.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Environment\GetTeklaPathsTool.cs

`list_environment_variables`

Lists Tekla-related environment variables. Probes a curated XS_* list via TeklaStructuresVariables.Get and merges with System.Environment vars filtered by prefix. Tekla source wins on duplicates.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Environment\ListEnvironmentVariablesTool.cs

Geometry

`compute_total_weight`

Sums the WEIGHT report property over a list of GUIDs (or, if empty, the current UI selection). Returns total in kg and per-element breakdown.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Geometry\ComputeTotalWeightTool.cs

`get_bounding_box`

Returns the axis-aligned bounding box (min/max/center/size in model coordinates) of one or more model objects. Pass either 'guid' or 'guids'.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Geometry\GetBoundingBoxTool.cs

`get_element_neighbors`

Returns model objects (default BEAM scope) within max_distance of the target element's AABB center, sorted by distance.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Geometry\GetElementNeighborsTool.cs

get_geometric_center

Returns the geometric center of one or more model objects (AABB center for one, mean of AABB centers for many).

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Geometry\GetGeometricCenterTool.cs

measure_angle

Measures the angle (degrees) between the centerlines of two beam-like model objects.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Geometry\MeasureAngleTool.cs

measure_distance

Measures the Euclidean distance between the AABB centers of two model objects (by GUID).

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Geometry\MeasureDistanceTool.cs

Grids

create_control_circle

Creates a construction control circle from three points (center, a point on the plane, and a radius point) mapped onto the ControlCircle 3-point constructor.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Grids\CreateControlCircleTool.cs

create_control_line

Creates a construction control line between two points. The 'infinite' flag is forwarded to the ControlLine constructor (Tekla treats it as the line's magnetic flag).

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Grids\CreateControlLineTool.cs

create_control_point

Creates a construction control point (auxiliary reference point) at the given coordinate.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Grids\CreateControlPointTool.cs

create_grid

[Grids/Write] Creates a rectangular structural grid. Coordinates and labels are space-separated strings (e.g. '0 6000 6000' / '1 2 3'). Tekla shorthand like '0 4*3000' is also accepted. Use extension_left_x=0 / extension_right_x=0 to suppress X-axis line markers (useful for single inclined Y-axis grids).

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Grids\CreateGridTool.cs

create_radial_grid

Creates a radial (polar) structural grid from radial coordinates (arc radius distances), angular coordinates (degrees) and Z levels, all as space-separated strings. Optional origin sets the centre point.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Grids\CreateRadialGridTool.cs

`get_grid_properties`

Reads coordinate and label strings of a rectangular or radial grid by GUID. Read-only.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Grids\GetGridPropertiesTool.cs

`list_grids`

Lists all rectangular and radial grids in the model with their coordinates and labels. Read-only; does not change UI selection.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Grids\ListGridsTool.cs

`modify_grid`

Modifies an existing rectangular or radial grid by GUID. Omitted coordinate/label fields keep their current values.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Grids\ModifyGridTool.cs

Meta

`get_version_info`

Returns TeklaCortex assembly versions and runtime info. Diagnostics for support reports.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Meta\VersionInfoTool.cs

`say_hello`

Quick handshake. Returns server name, version and current UTC time. Use to verify the bridge is alive.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Meta\SayHelloTool.cs

Model

`get_connection_status`

Checks whether a Tekla Structures instance is running and reachable. Returns connected=true/false plus diagnostics.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Model\GetConnectionStatusTool.cs

`get_model_info`

Returns name and path of the currently open Tekla model, plus working units.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Model\GetModelInfoTool.cs

Operations

`create_report`

Generates a Tekla report from all model objects using a report template, written to an output file. The template name must EXIST in XS_TEMPLATE_DIRECTORY or the model folder (e.g. 'A_Assembly_List', 'P_Part_List', 'B_Bolt_List' in a standard environment) - a non-existent template name can make Tekla block on an error dialog. Report creation is asynchronous; the tool polls for the output file up to `wait_seconds`.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Operations\CreateReportTool.cs

`create_report_from_selected`

Generates a Tekla report from the currently selected model objects using a report template, written to an output file. The template name must EXIST (e.g. 'A_Assembly_List', 'P_Part_List') or Tekla may block on an error dialog. Report creation is asynchronous; the tool polls for the file up to `wait_seconds`.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Operations\CreateReportFromSelectedTool.cs

`run_numbering`

Runs Tekla model numbering by invoking a numbering macro (the public Open API has no direct numbering-execute method; see tool notes). Reports numbering status.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Operations\RunNumberingTool.cs

Parameters

`get_all_element_udas`

Dumps every User Defined Attribute (UDA) set on a model object identified by GUID. No need to know the names in advance.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Parameters\GetAllElementUdasTool.cs

`get_all_report_properties`

Reads the requested report properties for a model object in one batch call. Tekla does not expose an enumeration of all available report names, so the caller must supply them.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Parameters\GetAllReportPropertiesTool.cs

`get_element_parameters`

Reads element parameters: all UDA/user properties, selected UDA names, and selected dynamic string properties.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Parameters\GetElementParametersTool.cs

`get_project_parameters`

Reads model/project parameters from ProjectInfo, plus selected project UDA/user and dynamic string properties.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Parameters\GetProjectParametersTool.cs

`set_element_parameters`

Sets element UDA/user properties and dynamic string properties. Dynamic string properties can be created by name; UDA names must exist in the Tekla environment.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Parameters\SetElementParametersTool.cs

set_project_parameters

Updates ProjectInfo fields and sets project-level UDA/user properties and dynamic string properties.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Parameters\SetProjectParametersTool.cs

Phases

create_phase

Creates a new phase with the given number, name and optional comment. Optionally sets it as the current phase.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Phases\CreatePhaseTool.cs

delete_phase

Deletes a phase identified by phase_number. Will fail if any objects are still assigned to that phase - move them first.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Phases\DeletePhaseTool.cs

find_elements_by_phase

Returns GUIDs of model objects assigned to a given phase. Scope is configurable via 'type_filter' (default BEAM).

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Phases\FindElementsByPhaseTool.cs

get_current_phase

Returns the current phase of the open Tekla model, or null if no phase is currently selected.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Phases\GetCurrentPhaseTool.cs

get_element_phase

Returns the phase a model object belongs to, identified by GUID. Returns phase: null if the object has no assigned phase.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Phases\GetElementPhaseTool.cs

list_phases

Lists all phases defined in the open Tekla model. Each phase reports number, name, comment and whether it is the current phase.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Phases\ListPhasesTool.cs

move_elements_to_phase

Bulk-assigns a list of model objects (by GUID) to an existing phase. Returns per-GUID success/failure details.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Phases\MoveElementsToPhaseTool.cs

set_current_phase

Sets the current phase of the open Tekla model by phase number. The phase must already exist in the model.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Phases\SetCurrentPhaseTool.cs

`set_element_phase`

Assigns a model object (identified by GUID) to an existing phase. Phase must already exist in the model.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Phases\SetElementPhaseTool.cs

Project

`convert_base_point`

Converts a point between Tekla model coordinates and a selected base point coordinate system.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Project\ConvertBasePointTool.cs

`get_project_info`

Returns project metadata (name, project number, designer, builder, address, dates).

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Project\GetProjectInfoTool.cs

`list_base_points`

Lists Tekla base points with model location, east/north/elevation values, angle to north, and project/current flags.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Project\ListBasePointsTool.cs

`upsert_base_point`

Creates or updates a Tekla base point by name/GUID, with model location, east/north/elevation, angle to north, and project/current flags.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Project\UpsertBasePointTool.cs

Properties

`delete_element_uda`

Clears a UDA by writing its type-appropriate empty value (Tekla has no DeleteUserProperty). Specify `value_type` so the right `SetUserProperty` overload is picked.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Properties\DeleteElementUdaTool.cs

`get_element_properties`

Reads report properties for a model object identified by GUID. Pass a list of property names (e.g. WEIGHT, LENGTH, PROFILE).

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Properties\GetElementPropertiesTool.cs

`get_element_uda`

Reads User Defined Attributes (UDA) for a model object identified by GUID. Pass UDA names to read.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Properties\GetElementUdaTool.cs

`set_element_class`

Bulk sets Part.Class on a list of GUIDs. Returns a per-GUID success/failure breakdown.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Properties\SetElementClassTool.cs

`set_element_lock`

Locks or unlocks model objects via the MODELOBJECT_LOCK UDA. Pass 'locked: true' to lock, false to unlock.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Properties\SetElementLockTool.cs

`set_element_uda`

Writes a User Defined Attribute on a model object identified by GUID. Value is auto-typed: int / double / string.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Properties\SetElementUdaTool.cs

Reinforcement

`add_rebar_set_leg_face`

Adds an extra leg face (from an [x,y,z] polygon) to an existing RebarSet and regenerates it via RebarSet.Modify(). Defensive build; returns the resulting leg-face count.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Reinforcement\AddRebarSetLegFaceTool.cs

`create_rebar_group`

Creates a reinforcement bar group (RebarGroup) inside a father part. Provide the father GUID and a list of polygons (group faces); optionally supply spacing values to distribute the bars.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Reinforcement\CreateRebarGroupTool.cs

`create_rebar_mesh`

Creates a reinforcement mesh (RebarMesh) inside a father part from a polygonal outline. Size is applied to both the longitudinal and cross bar directions.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Reinforcement\CreateRebarMeshTool.cs

`create_rebar_set`

Creates a modern RebarSet on a host part from one leg-face polygon (the face the bars span) and a guideline polyline (the bar direction + spacing). Sets RebarProperties (size/grade/class). The most complex reinforcement object in the Tekla API; build is defensive.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Reinforcement\CreateRebarSetTool.cs

`create_single_rebar`

Creates a single reinforcement bar (SingleRebar) following a polyline path inside a father part. Provide the father part GUID and a polygon (2+ points) describing the bar centre-line.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Reinforcement\CreateSingleRebarTool.cs

`get_rebar_properties`

Reads common properties (size, grade, class, name, numbering series) of a reinforcement object (SingleRebar, RebarGroup, RebarMesh or RebarSet) by GUID.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Reinforcement\GetRebarPropertiesTool.cs

`get_rebar_set_properties`

Reads the properties of a modern RebarSet by GUID: size, grade, class, name, bending radius, numbering series, leg-face count, guideline count and father part. Defensive reads.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Reinforcement\GetRebarSetPropertiesTool.cs

`list_reinforcement`

Lists reinforcement objects in the open Tekla model: legacy SingleRebar/RebarGroup/RebarMesh and modern RebarSet. Returns guid, id, type and (when available) size/grade.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Reinforcement\ListReinforcementTool.cs

Selection

`get_selected_elements`

Returns the elements the user has currently selected in Tekla. Lists Identifier.GUID, type and basic identification.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Selection\GetSelectedElementsTool.cs

`invert_selection`

Inverts the current UI selection within straight Beams (selected -> unselected, unselected -> selected). Pass max_scan to limit scope on huge models. Tekla's BEAM enum covers the most common Part subclass - use this as a fast Part-like proxy.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Selection\InvertSelectionTool.cs

`select_elements_by_filter`

Selects model objects matching profile/material/name/class. Each filter is optional; non-empty filters are AND-combined.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Selection\SelectElementsByFilterTool.cs

`summarize_selection`

Returns counts of currently UI-selected objects grouped by type, profile, material and class.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Selection\SummarizeSelectionTool.cs

UDA

`ensure_uda_definition`

Creates or updates `objects.inp` so a UDA appears in a tab for model object types. Follows the `objects_esempio.inp` pattern from the attached guide.

Scope: write | `sources\tekla-official\src\TeklaCortex.Tools\Uda\EnsureUdaDefinitionTool.cs`

`get_objects_inp_status`

Locates the current model `objects.inp` file and lists UDA definitions found in it.

Scope: read | `sources\tekla-official\src\TeklaCortex.Tools\Uda\GetObjectsInpStatusTool.cs`

`install_objects_inp`

Copies an `objects.inp` file into the current model folder, optionally backing up any existing file. Tekla must be restarted to load changes.

Scope: write | `sources\tekla-official\src\TeklaCortex.Tools\Uda\InstallObjectsInpTool.cs`

UI

`clear_ui_selection`

Clears the current selection in the Tekla Structures UI editor.

Scope: write | `sources\tekla-official\src\TeklaCortex.Tools\UI\ClearUiSelectionTool.cs`

`delete_reference_model`

Deletes a reference model from the active Tekla model by GUID.

Scope: write | `sources\tekla-official\src\TeklaCortex.Tools\UI\DeleteReferenceModelTool.cs`

`get_view_properties`

Returns the properties of a single Tekla model view selected by name or by zero-based index.

Scope: read | `sources\tekla-official\src\TeklaCortex.Tools\UI\GetViewPropertiesTool.cs`

`insert_reference_model`

Inserts a reference model file (IFC/DWG/...) at the given insertion point and scale.

Scope: write | `sources\tekla-official\src\TeklaCortex.Tools\UI\InsertReferenceModelTool.cs`

`list_model_views`

Lists Tekla model views (all or visible-only) with name, projection, rendering and display type.

Scope: read | `sources\tekla-official\src\TeklaCortex.Tools\UI\ListModelViewsTool.cs`

`list_reference_models`

Lists reference models (IFC/DWG/...) inserted in the active model with filename, title, visibility and placement.

Scope: read | `sources\tekla-official\src\TeklaCortex.Tools\UI\ListReferenceModelsTool.cs`

`select_objects_in_ui`

Selects the given model objects (by GUID) in the Tekla Structures UI editor, replacing the current selection.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\UI\SelectObjectsInUiTool.cs

`set_reference_model_visibility`

Shows or hides a reference model (by GUID) in the active Tekla model.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\UI\SetReferenceModelVisibilityTool.cs

Workplane

`get_current_workplane`

Returns the current working plane (transformation plane) of the open Tekla model: origin, axis X, axis Y, axis Z and whether it matches the global identity plane.

Scope: read | sources\tekla-official\src\TeklaCortex.Tools\Workplane\GetCurrentWorkplaneTool.cs

`set_workplane_to_global`

Resets the current working plane to the global identity plane (origin (0,0,0), axes aligned with global X/Y/Z). Returns the previous plane.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Workplane\SetWorkplaneToGlobalTool.cs

`set_workplane_to_object`

Sets the current working plane to match the local coordinate system of the model object identified by GUID (typically a Part). Returns the resulting plane.

Scope: write | sources\tekla-official\src\TeklaCortex.Tools\Workplane\SetWorkplaneToObjectTool.cs

Verified source

<https://github.com/LuDattilo/TeklaCortex>

main / 24dd5cab856e9c600038211a77648211a0f5919

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