

THE AI THAT WORKS INSIDE REVIT.

PULSE

The AI assistant for Autodesk® Revit®

300+ OPERATIONS • REVIT 2023-2027
MULTILINGUAL: EN • IT • FR • DE
JULY 2026 • v1.2.0

Table of Contents

01	Getting Started	Prerequisites, installation, first connection
02	How to Talk to Revit	Prompt strategies, roles, templates
03	Querying the Model	Find elements, filter, export data
04	Creating Elements	Place doors, walls, columns, grids, levels
05	Modifying Elements	Change parameters, move, copy, delete
06	Types and Families	Duplicate types, change type, manage families
07	Views and Sheets	Create plans, sections, 3D views, sheets, viewports
08	Schedules	Create schedules, export, modify fields
09	Annotations and Tags	Tag rooms, dimension, text notes, legends
10	Materials and Compound Structures	Use materials, compound layers
11	Linked Files	Navigate links, move, reload, highlight
12	Rename and Numbering	Bulk rename, number rooms/doors
13	Advanced Parameters	Shared parameters, project parameters, transfer
14	Analysis and Quality	Health check, warnings, clash, family audit
15	Model Cleanup	Purge, CAD cleanup, empty tags
16	Composite Workflows	Full audit, data roundtrip, sheet set
17	Security	Sandbox, read-only, audit log, confirmations
18	Session Optimization	Session patterns, token management
19	IFC	Import, export, native reconstruction
20	Power BI	Data publishing and reverse selection - Premium
21	Dynamo	Graph generation and execution - Premium
	Appendix	Ready-to-use prompt library by scenario

01 // GETTING STARTED

01

Getting Started

Pulse is an AI assistant that lets you control Autodesk Revit using natural language. Instead of navigating menus and dialogs, you talk to Claude and it executes operations in the model for you.

You can ask things like:

```
> How many walls are in the model and what types are used?
> Set the Manufacturer parameter to 'ACME' for all doors on Level 1
> Create a room schedule with name, number, area and department
```

Pulse Free and Pulse Premium

Pulse uses one installer. Without a valid Keygen license it runs as Pulse Free: a permanent mode with 113 read-only commands and mandatory Read-Only Mode. A valid license unlocks the 198 Premium commands, Power BI, Dynamo, and optional custom C# in the same program.

To move to Premium, open Pulse > License, enter the serial, and complete Keygen activation. There is no need to uninstall or reinstall Pulse.

What you need

1. Autodesk Revit 2023, 2024, 2025, 2026 or 2027
2. Claude Desktop with Pro or Max plan (download from claude.ai/download)
or Claude Code (CLI)
3. The official Pulse package (ZIP containing Pulse-Setup.exe)

Installation

Close Revit and Claude Desktop, fully extract the Pulse ZIP to a local folder, then run Pulse-Setup.exe and approve the UAC prompt. The installer requires administrator privileges and in a few seconds will:

- Copy the plugin to the Addins folders for all detected Revit versions
- Install the MCP server to %USERPROFILE%\pulse\server\
- Automatically configure Claude Desktop and Codex when detected

No Node.js, Python or other runtime required: the server is a self-contained executable (.exe).

After installation, verify Claude Desktop under Settings > Developer > Edit Config. The installer handles both the classic %APPDATA% path and MSIX installs under LocalCache while preserving preferences and other MCP servers:

```
{
  "mcpServers": {
    "pulse-revit": {
      "command": "C:\\Users\\<name>\\.pulse\\server\\Pulse.Server.exe",
      "args": []
    }
  }
}
```

```
}
```

Your first session

Follow these steps in order:

1. Open your Revit project
2. Click 'Pulse Switch' in the Revit ribbon to start the server
3. Restart Claude Desktop
4. Look for the hammer icon in Claude's input bar
5. Test the connection:

YOUR FIRST MESSAGE:

```
> Hello Revit, are you connected?
```

Claude will call `say_hello` and respond with a confirmation message from the plugin.

6. Orient yourself in the model:

SECOND MESSAGE:

```
> What is this project? Show me levels, links and worksets
```

You will receive a summary: project name, author, levels with elevations, links, active worksets.

TIP — The first `get_project_info` call of the session should be complete. Subsequent calls can filter for efficiency.

02

How to Talk to Revit

You don't need to learn command names or special syntax. Talk as you would to an expert BIM colleague. Claude understands context and automatically selects the right tool.

Set a role at the start

Begin each session by telling Claude who it is:

```
RECOMMENDED ROLE:  
> You are a BIM expert working in Revit. Only use the available MCP tools. Ask for confirmation before making significant changes.
```

Claude will be more precise in tool selection and will ask for confirmation on destructive operations.

Be specific

The more detail you provide, the better the results:

```
VAGUE (NOT RECOMMENDED):  
> Show me the walls  
  
SPECIFIC (RECOMMENDED):  
> Show me all walls of type 'Basic Wall: Concrete 200mm' on Level 1 with fire resistance rating
```

TIP — Always specify: category + type + level + parameters of interest.

Preview before modifying

For bulk operations, always ask for a preview first:

```
STEP 1 - PREVIEW:  
> Which doors would be modified if I change the Mark for all doors on Level 2?  
  
Claude will run a simulation (dryRun) showing how many elements would be affected.  
  
STEP 2 - CONFIRM:  
> OK, proceed with the change  
  
Revit will show a native confirmation dialog before applying the changes.
```

Useful prompt templates

Query	List all [CATEGORY] on [LEVEL] where [PARAMETER] = [VALUE]
Count	Count [CATEGORY] grouped by [PARAMETER] and show as a table
Check	Find all [CATEGORY] where [PARAMETER] is empty
Modify	Set [PARAMETER] = [VALUE] for all [CATEGORY] of type [TYPE] - show me first
Rename	Rename all [TYPE] views using the format [FORMAT]
Export	Export data for all [CATEGORY] with [PARAMETERS] as CSV
Duplicate type	Duplicate type [TYPE] as [NEW_NAME] and set [PARAMETER] to [VALUE]
Create sheets	Create sheets for all [TYPE] views, number them [FORMAT]

Compare

Compare values of [PARAMETER] between Level 1 and Level 2 for [CATEGORY]

03 // QUERYING THE MODEL

03

Querying the Model

These tools read data from the model without modifying it. They are safe to use at any time.

get_project_info

Gets general project information: name, author, levels, worksets, phases, linked files.

WHEN TO USE

At the start of every session to understand the model you are working with.

EXAMPLE 1

```
> What is this project? Show me all the details
```

Name: Residential Building A | Author: Studio XYZ | 5 levels | 3 worksets | 2 links

EXAMPLE 2

```
> Show me only the levels with their elevations
```

Ground Floor: 0.00m | Level 1: 3.50m | Level 2: 7.00m | Roof: 10.50m

TIP — The first call should include everything. Subsequent calls can filter to save tokens.

get_element_parameters

Shows all parameters (instance and type) of one or more elements, given their ID.

WHEN TO USE

When you want to inspect a specific element or understand what parameters it has.

EXAMPLE 1

```
> Show me all parameters of element 606873
```

Full list: Type Name, Family, Level, Height, Width, Fire Rating, Comments...

EXAMPLE 2

```
> What are the type parameters of this door?
```

Type params: [Type] Width = 900mm | [Type] Height = 2100mm | [Type] Fire Rating = EI60

ai_element_filter

Intelligent filter to find elements by category. Supports filters on types, instances, and limits.

WHEN TO USE

When you want a quick list of elements from a specific category.

EXAMPLE 1

> Find all structural columns in the model

Found 47 structural columns: 12 HEB240, 20 HEB300, 15 HEB360...

EXAMPLE 2

> Show me the first 5 wall types

5 types: Basic Wall 200mm | Basic Wall 300mm | Curtain Wall | Partition 100mm | Fire Wall REI120

TIP — Always wrap parameters in the 'data' object: {"data": {"filterCategory": "OST_Walls"}}

filter_by_parameter_value

Search elements by filtering on a specific parameter value. Supports: equals, contains, greater than, empty, etc.

WHEN TO USE

When you need elements matching precise conditions on a parameter.

EXAMPLE 1

> Find all doors where fire rating is empty

18 doors with no fire rating: D-101, D-105, D-112... (IDs: 12345, 12389, 12456)

EXAMPLE 2

> Which walls on Level 1 are wider than 200mm?

32 walls found, all of type 'Basic Wall 300mm' or 'Fire Wall REI120'

EXAMPLE 3

> Find rooms with no number assigned

5 rooms with no number: Room 1 (ID: 789), Room 2 (ID: 801)...

TIP — For type parameters, always use parameterType: 'type'. The default 'both' may not resolve correctly.

get_current_view_elements

Lists elements visible in the active view. You can filter by category and limit the count.

WHEN TO USE

When you want to analyze only what is visible in the current view.

EXAMPLE 1

> How many doors are in this view?

42 doors visible in view 'Ground Floor - Architectural'

EXAMPLE 2

> List all visible walls with type and length

Table: ID | Type | Length | 68 walls total, total length 1,247m

get_selected_elements

Reads elements currently selected in Revit.

WHEN TO USE

When you have selected elements in Revit and want to work with them from Claude.

EXAMPLE 1

> What do I have selected?

3 elements selected: Wall 12345 (Basic Wall 200mm), Door 12389 (M_Single-Flush 900x2100), Window 12456

EXAMPLE 2

> Show me the parameters of the selected elements

Full detail for each element with all instance and type parameters

export_elements_data

Exports element data as a JSON or CSV table. Choose categories, parameters, and apply filters.

WHEN TO USE

When you need to create a report or export data for external analysis.

EXAMPLE 1

> Export all door data as CSV with type, level and fire rating

CSV generated with 156 rows: ID, Type, Level, Fire Rating

EXAMPLE 2

> Give me a table of walls with type name and length

Formatted table: Type | Count | Total Length | 12 types, 347 walls

TIP — Always specify parameterNames and maxElements to limit the response.

export_room_data

Exports all room data: name, number, area, finishes, department.

WHEN TO USE

For space reports, area verification, or export for the project brief.

EXAMPLE 1

> Export all room data with areas and departments

85 rooms exported: Office 101 (25.3 sqm, Dept A) | Corridor 102 (12.1 sqm)...

EXAMPLE 2

> Are there any unplaced or unbounded rooms?

3 unplaced rooms, 2 unbounded rooms (area = 0)

export_to_excel

Exports element data directly to an Excel file (.xlsx).

WHEN TO USE

When you need an Excel file to share with the team or consultants.

EXAMPLE 1

```
> Export all doors to an Excel file on the desktop
```

File saved: C:/Users/.../Desktop/Doors.xlsx - 156 rows, 12 columns

EXAMPLE 2

```
> Create an Excel file with room data for the client
```

File generated with sheet 'Rooms': Name, Number, Area, Department, Floor Finish

get_linked_elements

Query elements inside linked Revit files.

WHEN TO USE

When you need to check data in linked models without opening them.

EXAMPLE 1

```
> Show all walls from the structural link
```

234 walls found in link 'Structure.rvt': 120 CLS300, 89 CLS200...

EXAMPLE 2

```
> How many doors are in the linked architectural model?
```

312 doors in link 'Architecture.rvt', distributed across 5 levels

get_elements_in_spatial_volume

Find which elements are inside a room or a custom volume.

WHEN TO USE

For room inventories, furniture checks, or space analysis.

EXAMPLE 1

```
> What furniture is in room 101?
```

8 elements: 2 desks, 4 chairs, 1 cabinet, 1 lamp

EXAMPLE 2

```
> List all elements in the director's office
```

15 elements found by category: Furniture (6), Equipment (4), Lighting (5)

get_room_openings

Find the doors and windows associated with each room.

WHEN TO USE

For opening reports, egress verification, or room data sheets.

EXAMPLE 1

```
> What doors and windows does room 305 have?
```

2 doors (M_Single-Flush 900x2100, M_Double-Flush 1800x2100) and 3 windows (Fixed 1200x1500)

EXAMPLE 2

```
> Show the openings for all rooms on Level 1
```

Table by room: Room | Doors | Windows | 25 rooms analysed

find_untagged_elements

Find elements without a tag in the current view.

WHEN TO USE

For documentation quality control.

EXAMPLE 1

```
> Which doors are not tagged in this view?
```

12 untagged doors: D-101, D-102, D-105... (placed but not annotated)

EXAMPLE 2

```
> Find untagged rooms
```

3 untagged rooms on Ground Floor

find_undimensioned_elements

Find elements with no dimensions in the active view.

WHEN TO USE

To complete documentation and verify everything is dimensioned.

EXAMPLE

```
> Which walls are not dimensioned in this floor plan?
```

8 undimensioned walls: IDs 456, 789, 1023... in view 'Ground Floor'

measure_between_elements

Measure the distance between two elements or two points.

WHEN TO USE

For quick distance checks without using Revit's dimension tool.

EXAMPLE 1

```
> How far is column A1 from column B1?
```

Center-to-center distance: 6.00m | Minimum distance: 5.76m

EXAMPLE 2

```
> Measure the distance between these two walls
```

Bounding box distance: 3.45m

04 // CREATING ELEMENTS

04

Creating Elements

These tools create new elements in the model. Revit will always show a confirmation dialog before executing important operations.

`create_point_based_element`

Place point-based elements: doors, windows, columns, furniture, equipment.

WHEN TO USE

When you need to place one or more elements at specific positions.

EXAMPLE 1

```
> Place a desk at coordinates (5, 3, 0) on Level 1
```

Desk created: ID 67890, family 'Desk 1500x800', position (5.0, 3.0, 0.0)

EXAMPLE 2

```
> Place a column at every grid intersection
```

12 columns created at intersections A1-A4, B1-B4, C1-C4

`create_line_based_element`

Create line-based elements: walls, beams, pipes, cables.

WHEN TO USE

To create walls, beams or other linear elements defined by start and end points.

EXAMPLE 1

```
> Create a wall from point (0,0) to point (10,0) at Ground Floor
```

Wall created: ID 67891, type 'Basic Wall 200mm', length 10.0m

EXAMPLE 2

```
> Add an HEB300 beam from (0,0,3.5) to (8,0,3.5)
```

Beam created: ID 67892, type HEB300, span 8.0m

create_floor

Create a floor from a boundary of points or from a room perimeter.

WHEN TO USE

To add floors to existing rooms or create custom slabs.

EXAMPLE 1

```
> Create a concrete floor in room 205
```

Floor created: ID 67893, type 'Concrete 150mm', area 25.3 sqm

EXAMPLE 2

```
> Add a slab at Level 2 with these 4 corner points
```

Slab created with custom boundary, area 120 sqm

create_room

Create a room element at a specific position.

WHEN TO USE

When rooms are missing or you need to add new ones.

EXAMPLE 1

```
> Create a room 'Office' on Level 1 at coordinates (5, 8)
```

Room created: 'Office', auto-assigned number, area 18.5 sqm

EXAMPLE 2

```
> Add room 'Archive' with number 201 on Level 2
```

Room 201 'Archive' created on Level 2

create_grid

Create a grid line system with regular spacing.

WHEN TO USE

At the start of a project or when adding a structural grid.

EXAMPLE 1

```
> Create a 5x4 grid with 6m spacing, axes A-E and 1-4
```

Grid created: 5 X-axes (A-E) + 4 Y-axes (1-4), 6.0m spacing

EXAMPLE 2

```
> Add 3 horizontal grids with 8m spacing starting from 1
```

3 grids created: 1, 2, 3 with 8.0m spacing

create_level

Create a new level at a specific elevation.

WHEN TO USE

To add floors, mezzanines, or roof levels.

EXAMPLE 1

```
> Create Level 3 at elevation 10.5 metres
```

Level 'Level 3' created at +10.500m, floor plan created automatically

EXAMPLE 2

```
> Add a mezzanine at elevation 4.5m without creating floor plans
```

Level 'Mezzanine' created at +4.500m, no associated views

create_array

Create linear or radial arrays of existing elements.

WHEN TO USE

To replicate elements with regular spacing.

EXAMPLE 1

```
> Copy this column 5 times with 3m spacing in the X direction
```

5 copies created: 3.0m spacing, IDs: 67901-67905

EXAMPLE 2

```
> Create a radial array of 8 elements around centre (5,5)
```

8 copies created at 45-degree intervals, radius 3.0m

create_structural_framing_system

Create a beam grid on a specified level.

WHEN TO USE

To quickly place a regular beam system.

EXAMPLE

```
> Create a beam system on Level 2 with 1.2m spacing between axes A-D
```

12 beams created: type IPE240, span 6.0m, spacing 1.2m

copy_elements

Copy elements with an X, Y, Z offset.

WHEN TO USE

To duplicate groups of elements with a displacement.

EXAMPLE 1

```
> Copy these elements 3 metres to the right
```

5 elements copied with offset X=3.0m, new IDs: 68001-68005

EXAMPLE 2

```
> Duplicate this group to the floor above
```

Elements copied with offset Z=3.5m

create_filled_region

Create a filled region in a view.

WHEN TO USE

To highlight zones in plans or create graphic areas.

EXAMPLE

```
> Draw a filled region around this zone
```

Region created with fill pattern 'Solid Fill', area 45 sqm

create_surface_based_element

Create surface-based elements such as roofs, floors or ceilings by defining perimeter points.

WHEN TO USE

To create elements with custom geometry from coordinates.

EXAMPLE 1

```
> Create a roof following these perimeter points
```

Roof created: type 'Generic - 400mm', area 125 sqm, 4 slopes

EXAMPLE 2

```
> Create a custom-shaped floor in this zone
```

Floor created: type 'Concrete Slab 200mm', area 68 sqm

05 // MODIFYING ELEMENTS

05

Modifying Elements

All modifications show a confirmation dialog in Revit. You can always undo with Ctrl+Z.

WARNING — Bulk modifications are powerful. Always use `dryRun` or ask for a preview before applying.

`set_element_parameters`

Set the value of one or more parameters on one or more elements.

WHEN TO USE

To modify any writable parameter: comments, mark, phase, etc.

EXAMPLE 1

```
> Set the Mark of door 12345 to 'D-001'
```

Parameter updated: door 12345, Mark = 'D-001'

EXAMPLE 2

```
> Change Comments to 'Verified' for elements 100, 200, 300
```

3 elements updated: Comments = 'Verified'

EXAMPLE 3

```
> Set fire rating 'EI60' for all selected doors
```

Revit confirmation -> 8 doors updated

`modify_element`

Move, rotate, mirror or copy elements in the model.

WHEN TO USE

To reposition elements without selecting them manually in Revit.

EXAMPLE 1

```
> Move wall 5678 by 2 metres in the X direction
```

Wall moved: translation X=2.0m, new position confirmed

EXAMPLE 2

```
> Rotate this element 45 degrees around its centre
```

Rotation applied: 45 degrees counter-clockwise

EXAMPLE 3

```
> Mirror these 3 elements about the Y axis
```

3 elements mirrored successfully

change_element_type

Change the type of one or more elements.

WHEN TO USE

When you need to update elements to a different type of the same family.

EXAMPLE 1

```
> Change all 'Basic Wall 200mm' walls to 'Basic Wall 300mm'
```

Revit confirmation -> 45 walls updated to new type

EXAMPLE 2

```
> Replace these doors with type 'Fire Door EI60'
```

12 doors changed to type 'Fire Door EI60'

operate_element

Select elements in Revit, or apply quick graphic overrides.

WHEN TO USE

To highlight problematic elements or select them for further work.

EXAMPLE 1

```
> Select elements 123, 456 and 789 in Revit
```

3 elements selected in the active view

EXAMPLE 2

```
> Highlight the problematic walls in red
```

Graphic override applied: red colour, 15 elements

color_elements

Colour elements by category based on a parameter value.

WHEN TO USE

For thematic visualisation: rooms by department, walls by type, etc.

EXAMPLE 1

```
> Colour all walls by type
```

View coloured: 5 colours assigned to 5 types, legend created

EXAMPLE 2

```
> Show rooms coloured by department
```

Rooms coloured: Marketing=blue, IT=green, HR=orange...

match_element_properties

Copy parameter values from a source element to one or more target elements.

WHEN TO USE

To apply the same properties to similar elements without retyping.

EXAMPLE 1

```
> Copy the Mark and Comments from element A to elements B, C and D
```

2 parameters copied to 3 elements: Mark, Comments

EXAMPLE 2

```
> Apply door 123's properties to all other doors of this type
```

8 parameters transferred to 24 doors

delete_element

Delete elements from the model by ID.

WHEN TO USE

When you need to remove specific elements.

EXAMPLE 1

```
> Delete elements 123, 456 and 789
```

Revit confirmation: 'Delete 3 elements?' -> 3 elements deleted

EXAMPLE 2

```
> Remove unused grids
```

dryRun: 4 grids would be deleted. Proceed? -> 4 grids removed

WARNING — Irreversible operation (beyond Ctrl+Z). Always use dryRun first.

save_selection

Save the current selection as a named set.

WHEN TO USE

To save a group of selected elements for later reuse.

EXAMPLE

```
> Save this selection as 'Walls to check'
```

Selection 'Walls to check' saved with 18 elements

load_selection

Load a previously saved selection and select it in the view.

WHEN TO USE

To retrieve a saved element set and continue working with it.

EXAMPLE

```
> Load selection 'Walls to check'
```

18 elements selected from selection 'Walls to check'

delete_selection

Delete a saved selection set.

WHEN TO USE

To clean up selection sets no longer needed.

EXAMPLE

```
> Delete saved selection 'Walls to check'
```

Selection set 'Walls to check' deleted

set_element_phase

Set the creation or demolition phase of elements.

WHEN TO USE

To manage project phases (existing, new construction, demolition).

EXAMPLE 1

```
> Set phase 'New Construction' for these walls
```

12 walls updated: Phase Created = 'New Construction'

EXAMPLE 2

```
> Mark these elements as demolished in phase 2
```

8 elements set as demolished

set_element_workset

Move elements to a different workset.

WHEN TO USE

To organise elements into the correct worksets in workshared models.

EXAMPLE 1

```
> Move all doors to workset 'Architecture - Doors'
```

156 doors moved to workset 'Architecture - Doors'

EXAMPLE 2

```
> Change the workset of selected elements
```

23 elements moved to the specified workset

override_graphics

Apply graphic overrides (colour, transparency, line weight) to specific elements.

WHEN TO USE

To highlight elements in a view without changing their properties.

EXAMPLE 1

```
> Colour these walls red with 50% transparency
```

Override applied: red (255,0,0), 50% transparency, 8 elements

EXAMPLE 2

```
> Reset graphic overrides for element 123
```

Overrides removed for element 123

send_code_to_revit

LAST RESORT: execute custom C# code inside Revit. Disabled by default - enable it in Settings > Tools (requires explicit confirmation on every run). Code is validated by a security sandbox.

WHEN TO USE

ONLY when no dedicated tool covers the operation. For parameters, filters, renaming, statistics and views always use the dedicated tools.

EXAMPLE

```
> Create a DirectShape geometry not covered by the standard tools
```

Code executed after confirmation. Geometry created.

WARNING — Disabled by default (EnableCodeExecution in settings.json). Code cannot access the filesystem, network, registry or processes. Forbidden namespaces: System.IO, System.Net, System.Diagnostics.Process.

06 // TYPES AND FAMILIES

06

Types and Families

Manage family types: duplicate to create variants, load new families, list available types.

duplicate_family_type

Duplicate a loadable family type (door, window, furniture, etc.) with a new name. You can also modify the new type's parameters in the same operation.

WHEN TO USE

When you want to create a variant of an existing type - e.g. a narrower door or taller window.

EXAMPLE 1

```
> Duplicate door '900x2100' as '800x2100' and set width to 800
```

Type created: '800x2100' in family M_Single-Flush, Width=800mm set

EXAMPLE 2

```
> Create a variant of window 'Fixed 1200x1500' called 'Fixed 1500x1500' with height 1500
```

Type duplicated: 'Fixed 1500x1500', parameter Height=1500mm applied

TIP — If the same type name exists in multiple families, specify `familyName` to disambiguate.

duplicate_system_type

Duplicate a system family type (wall, floor, roof, ceiling).

WHEN TO USE

To create variants of system types without starting from scratch.

EXAMPLE 1

```
> Duplicate wall type 'Basic Wall 200mm' as 'Basic Wall 250mm'
```

Type created: 'Basic Wall 250mm', category Walls

EXAMPLE 2

```
> Create a copy of floor type 'Concrete 150mm'
```

Type 'Concrete 150mm - Copy' created

load_family

Load a .rfa family into the model, or list available families by category.

WHEN TO USE

To import families from external files or explore the model catalogue.

EXAMPLE 1

```
> Load the family from file C:/Families/CustomDoor.rfa
```

Family 'CustomDoor' loaded with 3 types

EXAMPLE 2

```
> List available families in the Doors category
```

8 families: M_Single-Flush (4 types), M_Double-Flush (2 types)...

get_available_family_types

List all available family types, with filter by category or family name.

WHEN TO USE

To know which types you can use before creating or changing elements.

EXAMPLE 1

```
> Show all available door types
```

24 types in 5 families: M_Single-Flush (900x2100, 800x2100, 700x2100)...

EXAMPLE 2

```
> What types does the 'Fixed Window' family have?
```

3 types: 1200x1500, 1500x1500, 900x1200

export_families

Export families from the model as .rfa files to disk.

WHEN TO USE

To extract families for reuse in other projects.

EXAMPLE 1

```
> Export all door families to folder C:/Export
```

12 families exported to C:/Export/Doors/: M_Single-Flush.rfa, M_Double-Flush.rfa...

EXAMPLE 2

```
> Export furniture families grouped by category
```

8 families exported in subfolders by category

07 // VIEWS AND SHEETS

07

Views and Sheets

create_view

Create floor plans, reflected ceiling plans, sections, or 3D views.

WHEN TO USE

When you need a new view of the model.

EXAMPLE 1

```
> Create a floor plan of Level 2 at scale 1:100
```

View 'Level 2' created, scale 1:100, ID: 78901

EXAMPLE 2

```
> Create a section looking north
```

Section created: north direction, depth 20m

EXAMPLE 3

```
> Create a 3D view of the model
```

3D view created with auto-generated name

duplicate_view

Duplicate views with options: simple copy, with detailing, or as dependent.

WHEN TO USE

To create working copies of existing views.

EXAMPLE 1

```
> Duplicate this view with all detailing
```

View duplicated with details: 'Ground Floor - Copy'

EXAMPLE 2

```
> Create 3 dependent views from Level 1 plan
```

3 dependent views created

create_view_filter

Create view filters and apply them with graphic overrides.

WHEN TO USE

To highlight or hide elements based on rules.

EXAMPLE 1

```
> Create a filter that shows fire-rated walls in red
```

Filter 'Fire Walls' created, applied to view with red colour

EXAMPLE 2

```
> Hide demolished elements in the current view
```

Filter 'Demolished' applied, elements hidden

apply_view_template

Apply or remove view templates.

WHEN TO USE

To standardise the appearance of views.

EXAMPLE 1

```
> Apply the 'Structural Plan' template to all floor plans
```

Template applied to 8 views

EXAMPLE 2

```
> Remove the template from selected views
```

Template removed from 3 views

rename_views

Rename views in bulk with prefixes, suffixes, or find-and-replace.

WHEN TO USE

To standardise view names.

EXAMPLE 1

```
> Add prefix 'REV-' to all sections
```

12 sections renamed: 'Section 1' -> 'REV-Section 1'...

EXAMPLE 2

```
> Replace 'OLD' with 'NEW' in view names
```

5 views renamed

create_sheet

Create a single sheet with a title block.

WHEN TO USE

To add sheets to the drawing set.

EXAMPLE

```
> Create sheet A-101 'Ground Floor Plan'
```

Sheet A-101 created with standard title block

batch_create_sheets

Create multiple sheets at once.

WHEN TO USE

To quickly set up a project's sheet set.

EXAMPLE

```
> Create sheets from A-101 to A-110 with our company title block
```

10 sheets created: A-101...A-110 with 'Company Titleblock'

place_viewport

Place a view on a sheet.

WHEN TO USE

To compose sheets with the desired views.

EXAMPLE

```
> Place Level 1 plan on sheet A-101
```

Viewport placed at centre of sheet A-101

align_viewports

Align viewports across different sheets.

WHEN TO USE

To maintain graphic consistency between sheets.

EXAMPLE

```
> Align all viewports to the first one's position
```

5 viewports aligned to the reference position

duplicate_sheet_with_content

Duplicate a sheet with all its content (views, legends, schedules).

WHEN TO USE

To create complete sheet copies.

EXAMPLE

```
> Make 3 copies of sheet A-101 with all views
```

3 sheets duplicated: A-101a, A-101b, A-101c with copied views

duplicate_sheet_with_views

Duplicate sheet with advanced options for views (duplicate, with detailing, dependent).

WHEN TO USE

When you want to control how views are duplicated in the copy.

EXAMPLE

```
> Duplicate the sheet with dependent views
```

Sheet duplicated, 3 views created as dependents

create_views_from_rooms

Automatically create views (callouts, sections, elevations) centred on rooms.

WHEN TO USE

For automatic documentation of interior spaces.

EXAMPLE 1

```
> Create sections for all rooms on Level 1
```

25 sections created, one per room, name 'Section - Office 101'...

EXAMPLE 2

```
> Generate callouts for rooms 101-110
```

10 callouts created in the corresponding floor plans

create_placeholder_sheets

Create placeholder sheets for drawing set planning.

WHEN TO USE

To plan the sheet set before views are ready.

EXAMPLE

```
> Create placeholder sheets for the drawing set
```

15 placeholders created: S-001...S-015

manage_view_templates

List, duplicate, delete or rename view templates.

WHEN TO USE

To manage the template library.

EXAMPLE

```
> List all view templates
```

12 templates: Architectural Plan, Structural Plan, Section, 3D...

manage_unplaced_views

Find or delete views not placed on any sheet.

WHEN TO USE

For model cleanup, removing orphaned views.

EXAMPLE 1

```
> Show me all views not placed on sheets
```

34 unplaced views: 12 floor plans, 8 sections, 14 3D views

EXAMPLE 2

```
> Delete unplaced floor plans (preview first)
```

dryRun: 12 floor plans would be deleted. Confirm?

batch_modify_view_range

Modify the view range (cut plane, top, bottom) for multiple views.

WHEN TO USE

To standardise view range across multiple floor plans.

EXAMPLE

```
> Set cut plane to 1200mm for all floor plans
```

8 floor plans updated: cut plane = 1200mm

section_box_from_selection

Create a 3D view with a section box around selected elements.

WHEN TO USE

To isolate and visualise specific elements in 3D.

EXAMPLE

```
> Create a cropped 3D view around these beams
```

3D view 'Section Box - Beams' created with automatic box + 0.5m offset

get_current_view_info

Shows information about the active view in Revit.

WHEN TO USE

To know which view you are in before operating.

EXAMPLE

```
> Which view am I in?
```

Active view: 'Ground Floor - Architectural', type FloorPlan, scale 1:100

batch_export

Export views and sheets to CAD formats (DWG, DXF, DGN) or images.

WHEN TO USE

To deliver sheets in CAD format or generate images for presentations.

EXAMPLE 1

```
> Export all sheets to DWG
```

24 sheets exported to DWG in the output folder

EXAMPLE 2

```
> Export Ground Floor views as PNG images
```

3 views exported: Ground Floor - Architectural.png, Ground Floor - Structural.png...

08 // SCHEDULES

08

Schedules

create_preset_schedule

Create a schedule from a preset template: doors by room, windows, rooms, materials, sheet list.

WHEN TO USE

The fastest way to create standard schedules.

EXAMPLE 1

```
> Create a door schedule by room
```

Schedule 'Doors by Room' created with fields: Room, Door Number, Type, Dimensions

EXAMPLE 2

```
> Create a sheet list
```

Sheet list created with Number, Name, Revision

create_schedule

Create a custom schedule by choosing category, type and fields.

WHEN TO USE

When presets are not enough and you need a tailored schedule.

EXAMPLE

```
> Create a door schedule with Name, Mark and Fire Rating
```

Schedule 'Doors' created with 3 fields, sorted by Name

get_schedule_data

Read data from an existing schedule as a table.

WHEN TO USE

To consult schedule data without opening it in Revit.

EXAMPLE 1

```
> Show me the data in 'Door Schedule'
```

156 rows: Door D-001 | M_Single-Flush 900x2100 | EI60 | Level 1...

EXAMPLE 2

```
> Read the first 20 rows of the room list
```

20 rows with Name, Number, Area, Department

export_schedule

Export a schedule as a CSV/TSV file.

WHEN TO USE

To share data with external teams or import into Excel.

EXAMPLE

```
> Export the door schedule as CSV
```

File saved: DoorSchedule.csv, 156 rows exported

modify_schedule

Add/remove fields, set sorting, rename a schedule.

WHEN TO USE

To modify the structure of an existing schedule.

EXAMPLE 1

```
> Add the 'Fire Rating' field to the door schedule
```

Field added in last position

EXAMPLE 2

```
> Sort the room schedule by Level then by Name
```

Sorting applied: 1) Level, 2) Name

duplicate_schedule

Duplicate an existing schedule with a new name.

WHEN TO USE

To create schedule variants without starting from scratch.

EXAMPLE

```
> Duplicate 'Door Schedule' as 'Door Schedule - QC'
```

Schedule duplicated: 'Door Schedule - QC' with same fields and filters

delete_schedule

Delete a schedule from the model.

WHEN TO USE

To remove obsolete or duplicate schedules.

EXAMPLE

```
> Delete schedule 'Old Door Schedule'
```

Revit confirmation -> Schedule deleted

create_revision

Create a revision or add revisions to sheets.

WHEN TO USE

To manage the project revision cycle.

EXAMPLE 1

```
> Create a new revision with today's date and description 'Client Review'
```

Revision #3 created: date 2026-04-13, 'Client Review'

EXAMPLE 2

```
> Add revision 1 to sheets A-101 through A-110
```

Revision added to 10 sheets

list_schedulable_fields

Show which fields you can add to a schedule for a given category.

WHEN TO USE

Before creating a schedule, to know which parameters are available.

EXAMPLE

```
> Which fields can I use in a door schedule?
```

42 fields available: Family and Type, Width, Height, Fire Rating, Level, Room...

09 // ANNOTATIONS AND TAGS

09

Annotations and Tags

tag_rooms

Add tags to all rooms in the current view.

WHEN TO USE

To quickly annotate rooms after creating or checking them.

EXAMPLE 1

```
> Tag all rooms in this view
```

25 tags added in view 'Ground Floor'

EXAMPLE 2

```
> Add tags with leaders to rooms 101-110
```

10 tags with leaders placed

tag_walls

Add tags to walls in the current view.

WHEN TO USE

To annotate walls with their type in floor plans.

EXAMPLE

```
> Tag all walls in this view
```

68 wall tags added in the floor plan

create_dimensions

Create dimension lines between elements.

WHEN TO USE

To dimension distances between grids, walls, or other elements.

EXAMPLE

```
> Dimension the distance between grid A and grid B
```

Dimension created: 6.00m between Grid A and Grid B

create_text_note

Create text notes in views.

WHEN TO USE

To add annotations and comments in views.

EXAMPLE

```
> Add a note 'Check on site' at this position
```

Text note created in the active view

create_color_legend

Create a colour legend for parameter values.

WHEN TO USE

To accompany coloured views with a readable legend.

EXAMPLE

```
> Create a legend for rooms coloured by department
```

Legend created with 6 colours: Marketing=blue, IT=green, HR=orange...

import_table

Import a CSV/TSV table into a drafting view or legend.

WHEN TO USE

To insert external tables into Revit documentation.

EXAMPLE

```
> Import the table from the CSV file into the drafting view
```

Table imported: 25 rows x 5 columns in view 'Details'

wipe_empty_tags

Remove all tags with empty values from the current view.

WHEN TO USE

To clean documentation from unfilled tags.

EXAMPLE 1

```
> Delete empty room tags in the current view
```

dryRun: 8 empty tags found. Proceed? -> 8 tags removed

EXAMPLE 2

```
> Clean all empty door tags
```

12 empty door tags removed

10 // MATERIALS AND COMPOUND STRUCTURES

10

Materials and Compound Structures

get_materials

List materials in the model, with filter by class or name.

WHEN TO USE

To explore the model's material library.

EXAMPLE 1

```
> Show all materials in the model
```

48 materials: Concrete (5), Steel (3), Glass (2), Brick (4)...

EXAMPLE 2

```
> Find materials with 'Concrete' in the name
```

3 materials: Concrete C25/30, Lightweight Concrete, Reinforced Concrete

get_material_properties

Show detailed properties of a material.

WHEN TO USE

To inspect colour, transparency, class and assets of a material.

EXAMPLE

```
> Show properties of material 'Concrete - Cast-in-Place'
```

Class: Concrete | Colour: grey (180,180,180) | Transparency: 0% | Structural asset: present

get_material_quantities

Calculate material quantities used by category.

WHEN TO USE

For quantity takeoffs and material analysis.

EXAMPLE

```
> How much material is used in walls and floors?
```

Walls: Concrete 450 sqm, Brick 320 sqm | Floors: Ceramic 280 sqm, Concrete 150 sqm

create_material

Create a new material with colour, transparency and class.

WHEN TO USE

To add custom materials to the model.

EXAMPLE

```
> Create a material 'Blue Glass' with 40% transparency and blue colour
```

Material 'Blue Glass' created: colour (0,100,200), transparency 40%

duplicate_material

Duplicate an existing material with all its properties.

WHEN TO USE

To create variants of existing materials.

EXAMPLE

```
> Duplicate 'Concrete' as 'Special Concrete'
```

Material duplicated with all assets (appearance, structural, thermal)

delete_material

Delete a material from the model.

WHEN TO USE

To remove unused materials.

EXAMPLE

```
> Delete material 'Test Material'
```

Revit confirmation -> Material deleted

set_material_properties

Modify properties of existing materials.

WHEN TO USE

To update colours, transparency or other properties.

EXAMPLE 1

```
> Change the concrete colour to dark grey
```

Colour updated: (100,100,100)

EXAMPLE 2

```
> Set glass transparency to 50%
```

Transparency updated: 50%

get_compound_structure

Show the compound structure (layer composition) of walls, floors or roofs.

WHEN TO USE

To check the constructive composition of a type.

EXAMPLE 1

```
> Show the compound structure of 'External Wall 300mm'
```

4 layers: Plaster 15mm | Brick 120mm | Insulation 80mm | Plasterboard 12.5mm

EXAMPLE 2

```
> What is the composition of the default floor type?
```

3 layers: Screed 60mm | Concrete 150mm | Waterproofing 5mm

set_compound_structure

Modify the compound structure of walls, floors or roofs. Add, remove or replace layers.

WHEN TO USE

To change insulation thickness, add a finish coat, etc.

EXAMPLE 1

```
> Add an 80mm insulation layer to the wall
```

Layer added: Insulation 80mm, position 3/5

EXAMPLE 2

```
> Replace the finish with 12.5mm plasterboard
```

Layer 1 replaced: Plasterboard 12.5mm

11 // LINKED FILES

11

Linked Files

`get_linked_file_instances`

List all linked Revit files with their status.

WHEN TO USE

To get an overview of links in the model.

EXAMPLE

```
> Show all linked files
```

```
3 links: Structure.rvt (loaded), MEP.rvt (loaded), Architecture_OLD.rvt (unloaded)
```

`manage_links`

List, reload or unload linked files.

WHEN TO USE

To manage link status without using Revit's 'Manage Links' dialog.

EXAMPLE 1

```
> Reload all linked files
```

```
3 links reloaded successfully
```

EXAMPLE 2

```
> Unload link 'Architecture_OLD.rvt'
```

```
Link unloaded from memory
```

`add_linked_file`

Add a new Revit file as a link.

WHEN TO USE

To link a new model to the project.

EXAMPLE

```
> Link the structural model from this path
```

```
Link added: 'Structure.rvt', origin position
```

reload_linked_file_from

Reload a link from a different path.

WHEN TO USE

When the linked file has been moved or renamed.

EXAMPLE

```
> Reload the structural link from the new path
```

Link reloaded from new path, 234 elements updated

get_link_transform

Show the position and rotation of a link instance.

WHEN TO USE

To verify link alignment.

EXAMPLE

```
> Where is the structural link?
```

Position: (0.0, 0.0, 0.0) | Rotation: 0 degrees | System: Shared

align_link_to_host

Align a link to the origin or shared coordinates.

WHEN TO USE

To correctly reposition a misaligned link.

EXAMPLE

```
> Align the structural link to shared coordinates
```

Link aligned to shared coordinate system

move_link_instance

Move a link instance in the model.

WHEN TO USE

To reposition a link precisely.

EXAMPLE

```
> Move the MEP link 5 metres east
```

Link moved: translation X=5.0m

pin_unpin_link_instance

Pin or unpin link instances.

WHEN TO USE

To protect links from accidental movement.

EXAMPLE 1

```
> Pin all links
```

3 instances pinned

EXAMPLE 2

```
> Unpin the structural link to reposition it
```

1 instance unpinned

highlight_linked_element

Highlight and zoom to a specific element inside a link, with optional section box.

WHEN TO USE

To visually locate an element in a linked model.

EXAMPLE

```
> Show me element 789 in the structural link
```

Element highlighted, section box created with 2m offset

get_selected_linked_elements

Read information on selected elements that belong to links.

WHEN TO USE

When you select link elements in Revit and want to analyse them.

EXAMPLE

```
> What do I have selected in the links?
```

2 elements from link 'Structure.rvt': Beam HEB300 (ID: 45678), Column HEB240 (ID: 45690)

12 // RENAME AND NUMBERING

12

Rename and Numbering

batch_rename

Rename elements in bulk with find-and-replace, prefix, or suffix.

WHEN TO USE

To standardise names of views, sheets, levels, rooms.

EXAMPLE 1

```
> Rename all views replacing 'OLD' with 'NEW'
```

5 views renamed: 'OLD Level 1' -> 'NEW Level 1'...

EXAMPLE 2

```
> Add prefix 'QC-' to all room names
```

85 rooms renamed: 'Office' -> 'QC-Office'...

rename_families

Rename families and optionally their types.

WHEN TO USE

To standardise family names in the model.

EXAMPLE 1

```
> Add prefix 'STD-' to all door families
```

8 families renamed: 'M_Single-Flush' -> 'STD-M_Single-Flush'...

EXAMPLE 2

```
> Replace 'Generic' with 'Custom' in family names
```

3 families renamed

renumber_elements

Automatically number elements in sequence: rooms, doors, windows, parking.

WHEN TO USE

To assign sorted numbers after placing elements.

EXAMPLE 1

```
> Number all rooms starting from 101
```

25 rooms numbered: 101, 102, 103... sorted by position

EXAMPLE 2

```
> Number doors by room in alphabetical order
```

Doors numbered: D-001 (Archive), D-002 (Corridor), D-003 (Office 1)...

13 // ADVANCED PARAMETERS

13

Advanced Parameters

bulk_modify_parameter_values

Modify a parameter in bulk: set value, prefix, suffix, find-and-replace, clear.

WHEN TO USE

To update the same parameter on many elements at once.

EXAMPLE 1

```
> Set Manufacturer to 'ACME' for all doors
```

```
dryRun: 156 doors would be updated. Proceed? -> 156 doors updated
```

EXAMPLE 2

```
> Add prefix 'STR-' to the Mark of all beams
```

```
47 beams updated: 'B-01' -> 'STR-B-01'...
```

sync_csv_parameters

Synchronise parameter values from CSV data (array of elementId + parameters).

WHEN TO USE

To import data from spreadsheets or external databases.

EXAMPLE

```
> Update room numbers from this CSV data
```

```
dryRun: 25 rooms would be updated. -> 25 updated successfully
```

import_from_excel

Import parameter values from an Excel file into the Revit model.

WHEN TO USE

To bulk-update parameters by reading data from an externally prepared Excel sheet.

EXAMPLE 1

```
> Import room data from the Excel file
```

```
dryRun: 42 rooms would be updated from 'Room_Data.xlsx'. -> 42 updated successfully
```

EXAMPLE 2

```
> Update door parameters from the Excel file
```

```
Imported: 156 values on 52 doors, 3 errors (ID not found)
```

TIP — Always run with dryRun first to verify data matching.

transfer_parameters

Copy parameter values from a source element to other elements.

WHEN TO USE

To propagate properties from a 'template' element to similar ones.

EXAMPLE

```
> Copy all parameters from door 123 to doors 456, 789
```

dryRun: 8 parameters would be copied to 2 elements -> Complete

add_prefix_suffix

Add a prefix or suffix to parameter values.

WHEN TO USE

To add codes or indicators to existing parameters.

EXAMPLE

```
> Add 'REV-' as prefix to all room numbers
```

25 rooms updated: '101' -> 'REV-101'...

clear_parameter_values

Clear the value of a parameter for a category/scope.

WHEN TO USE

To reset parameters before a new data entry.

EXAMPLE

```
> Clear Comments for all walls
```

68 walls: Comments parameter cleared

add_shared_parameter

Add a shared parameter to one or more categories.

WHEN TO USE

To add custom parameters to the model.

EXAMPLE

```
> Add shared parameter 'QC Status' to Doors and Windows categories
```

Parameter 'QC Status' added as instance to Doors and Windows

manage_project_parameters

List, create or delete project parameters.

WHEN TO USE

To manage project-specific parameters.

EXAMPLE 1

```
> List all project parameters
```

18 parameters: QC Status (Text), Check Date (Text), Inspector (Text)...

EXAMPLE 2

```
> Create a parameter 'Inspector' of type Text for Rooms
```

Parameter created: 'Inspector', type Text, instance, applied to Rooms

get_shared_parameters

List shared parameters in the model.

WHEN TO USE

To check which shared parameters are available.

EXAMPLE

```
> List shared parameters for the Doors category
```

5 shared parameters: Fire Rating, Sound Insulation, QC Status...

export_shared_parameter_file

Export shared parameter definitions to a .txt file.

WHEN TO USE

To share parameters between projects.

EXAMPLE

```
> Export shared parameters to a file
```

File exported: SharedParameters.txt, 18 parameters

14 // ANALYSIS AND QUALITY

14

Analysis and Quality

check_model_health

Quick model health check with score and main issues.

WHEN TO USE

First thing in the morning or before a deliverable.

EXAMPLE 1

```
> What is the model health?
```

Score: 78/100 | 3 issues: 45 warnings, 12 unused families, 3 heavy views

EXAMPLE 2

```
> Is the model ready for delivery?
```

Score: 92/100 | 1 minor issue: 5 empty tags

analyze_model_statistics

Detailed analysis: element count by category, most used types.

WHEN TO USE

To understand the composition and complexity of the model.

EXAMPLE 1

```
> How many elements are in the model?
```

Total: 12,456 elements | Walls: 347 | Doors: 156 | Windows: 89 | Rooms: 85...

EXAMPLE 2

```
> Show statistics in compact format
```

Compact: 12.4K elements, 347 walls, 156 doors, 89 windows

get_warnings

Read active warning messages in the model.

WHEN TO USE

For quality control and problem resolution.

EXAMPLE 1

```
> Show me the first 10 warnings
```

10 warnings: 4 'Elements overlap', 3 'Room height below minimum', 2 'Missing connection'...

EXAMPLE 2

```
> Are there any errors in the model?
```

0 errors, 45 warnings. Most frequent: wall overlap (12), connections (8)

audit_families

Check loaded families: size, instance count, unused families.

WHEN TO USE

To identify heavy or removable families.

EXAMPLE 1

```
> Check door families
```

8 families, 24 types, 156 instances | 2 unused types: 'Glass Door' and 'Revolving'

EXAMPLE 2

```
> Find unused families in the model
```

15 families with no instances, occupy 12MB. Want to purge?

clash_detection

Detect geometric clashes between two element categories.

WHEN TO USE

For inter-discipline coordination.

EXAMPLE 1

```
> Check clashes between beams and ducts
```

8 clashes found: Beam B-12 vs Duct D-05 (overlap 45mm)...

EXAMPLE 2

```
> Are there any walls intersecting columns?
```

3 clashes: Wall M-23 vs Column P-05 (penetration 120mm)...

lines_per_view_count

Count lines per view to identify heavy views.

WHEN TO USE

To find views that slow down the model.

EXAMPLE

```
> Which views have the most lines?
```

Top 5: Detail 42 (15,230 lines), Ground Floor (8,450), Section A (6,200)...

list_family_sizes

List families by instance count or types.

WHEN TO USE

To identify the most used or heaviest families.

EXAMPLE

```
> What are the 20 most used families?
```

Top 20: Basic Wall (347 inst.), M_Single-Flush (124 inst.), Fixed Window (89 inst.)...

get_phases

Show project phases and phase filters.

WHEN TO USE

To understand the project's temporal structure.

EXAMPLE

```
> Show project phases
```

3 phases: Existing (1), Demolition (2), New Construction (3) | 4 phase filters

get_worksets

List worksets with element count.

WHEN TO USE

To verify workset organisation.

EXAMPLE

```
> Show model worksets
```

5 worksets: Architecture (4,500 el.), Structure (2,100 el.), MEP (1,800 el.)...

15 // MODEL CLEANUP

15

Model Cleanup

purge_unused

Delete unused families, types and materials from the model.

WHEN TO USE

To reduce file size and improve performance.

EXAMPLE 1

```
> What can I delete from the model?
```

```
dryRun: 15 families, 23 types, 8 materials can be deleted (-12MB)
```

EXAMPLE 2

```
> Clean up everything that is not used
```

```
Revit confirmation -> 46 elements deleted, file reduced by 12MB
```

WARNING — Powerful operation. Always use dryRun first to verify.

cad_link_cleanup

Find and remove imported or linked CAD files.

WHEN TO USE

To remove DWG/DXF that are weighing down the model.

EXAMPLE 1

```
> List all CAD imports in the model
```

```
8 imports: floorplan.dwg (Ground Floor), detail.dwg (Section A)...
```

EXAMPLE 2

```
> Delete all DWG imports
```

```
Confirmation -> 8 CAD imports removed
```

16 // COMPOSITE WORKFLOWS

16

Composite Workflows

These tools combine multiple operations in a single command for complex tasks.

workflow_model_audit

Complete model audit: health + warnings + family analysis in one call.

WHEN TO USE

For a complete quality check before delivery.

EXAMPLE 1

```
> Run a full model audit
```

Report: Score 82/100 | 45 warnings (12 critical) | 15 unused families | 3 heavy views

EXAMPLE 2

```
> Audit with detailed warnings and families
```

Detailed report: warnings by category, families by size, recommendations

workflow_clash_review

Detect clashes and automatically create a 3D view with section box for review.

WHEN TO USE

When you need to visualise clashes, not just count them.

EXAMPLE 1

```
> Check beam-duct clashes with a 3D view
```

8 clashes found. 3D view 'Clash Review - Beams vs Ducts' created with section box

EXAMPLE 2

```
> Review wall-column clashes
```

3 clashes found, 3D view created for each cluster

workflow_data_roundtrip

Export data to Excel, edit externally, and reimport into the model.

WHEN TO USE

For bulk parameter updates via spreadsheet.

EXAMPLE 1

```
> Export door data to Excel, I'll edit it and reimport
```

File exported: Doors_Export.xlsx, 156 rows, 8 columns. Edit it and let me know when to reimport.

EXAMPLE 2

```
> Reimport the modified Excel file
```

dryRun: 23 doors would be updated, 2 parameters per door. Proceed?

workflow_room_documentation

Automatically create sections for every room on a level.

WHEN TO USE

To quickly generate room documentation.

EXAMPLE

```
> Create sections for all rooms on Level 1
```

25 sections created, one per room. Names: 'Section - Office 101', 'Section - Corridor 102'...

workflow_sheet_set

Create a complete set of sheets with title block in a single operation.

WHEN TO USE

To quickly set up a project drawing set.

EXAMPLE

```
> Create sheets A-101 to A-105 with the company title block
```

5 sheets created: A-101 Ground Floor | A-102 Level 1 | A-103 Level 2 | A-104 Sections | A-105 Elevations

17 // SECURITY

17

Security

Pulse includes several protections to prevent accidental damage to the model.

Confirmation dialog

All destructive operations (delete, purge, modify parameters, rename) show a native Revit confirmation dialog before executing. If you cancel, the tool returns a 'Cancelled' error and nothing is modified.

Code sandbox

The `send_code_to_revit` tool validates C# code before executing it. The following namespaces are forbidden and cause a 'PermissionDenied' error:

```
System.IO           - filesystem access
System.Net          - network access
System.Diagnostics  - system processes
Microsoft.Win32     - Windows registry
System.Reflection.Emit - dynamic code generation
System.Runtime.InteropServices - native interop
```

Read-only mode

In Pulse Free, read-only mode is compiled into the build and cannot be disabled through Settings or `settings.json`. In Pulse Premium it remains an optional safety control. Only read operations remain active while it is enabled.

Audit log

Every tool execution is recorded in `~/.pulse/audit.jsonl` with timestamp, tool name, result and number of elements involved.

18 // SESSION OPTIMISATION

18

Session Optimisation

Recommended session patterns

Don't do everything in one conversation. Divide work into short, focused sessions:

Morning session (2-3 calls)

```
> What is the model health? Show me the 10 main warnings.
```

Tools: check_model_health + get_warnings. Cost: ~800 response tokens.

Parameter session (3-4 calls)

```
> Export door data, then set manufacturer 'ACME' for Fire Door type doors.
```

Tools: export_elements_data + bulk_modify (dryRun) + bulk_modify. Cost: ~2,000 tokens.

Documentation session (3 calls)

```
> Create a door schedule by room, export it as CSV, and create sheets A-101 to A-105.
```

Tools: create_preset_schedule + export_schedule + workflow_sheet_set. Cost: ~1,500 tokens.

Golden rules

1. The first get_project_info must be complete. Subsequent calls must filter.
2. Use maxWarnings: 10 for quick checks, not the default 500.
3. With dryRun, read only modifiedCount/skippedCount, not the full list.
4. When context exceeds ~15,000 response tokens, start a new conversation.
5. Don't mix QA tasks with authoring in the same long session.
6. Use the most targeted tool: check_model_health before analyze_model_statistics.

Performance

- Read operations: safe in parallel (5+)
- Write operations: max 3-4 in parallel
- Heavy queries (analyze_model_statistics, purge_unused): run individually
- create_view 3D: particularly heavy, avoid alongside other writes

Language and localisation

Revit translates category and parameter names based on language. Pulse auto-detects the language. Always use OST_codes (language-independent) for categories:

OST_Walls = Walls / Muri / Murs / Wände

OST_Doors = Doors / Porte / Portes / Turen

OST_Windows = Windows / Finestre / Fenêtres / Fenster

OST_Rooms = Rooms / Vani / Pieces / Raume
OST_Floors = Floors / Pavimenti / Sols / Geschossdecken
OST_StructuralFraming = Structural Framing
OST_StructuralColumns = Structural Columns

Storage and analysis tools

These tools operate server-side to store data between sessions:

store_project_data

Save project metadata for retrieval in future sessions.

WHEN TO USE

To keep a history of analysed projects.

EXAMPLE

```
> Save information about this project
```

Project 'Building A' saved to local database

store_room_data

Save a snapshot of room data.

WHEN TO USE

To compare rooms over time.

EXAMPLE

```
> Save the room data for later comparison
```

85 rooms saved for project 'Building A'

query_stored_data

Query previously saved data.

WHEN TO USE

To retrieve data from past sessions.

EXAMPLE

```
> Show saved projects
```

3 projects: Building A, School B, Hospital C

report_token_usage

Report on MCP tool usage.

WHEN TO USE

To understand which tools you use most.

EXAMPLE

```
> Which tools have I used most this week?
```

Top 5: get_element_parameters (45x), filter_by_parameter_value (23x), export_elements_data (18x)...

analyze_journal

Analyse Revit journal files for diagnostics.

WHEN TO USE

To investigate crashes, memory issues, or abnormal behaviour.

EXAMPLE

```
> Analyse the last 3 Revit sessions
```

Session 1: 45 min, 3.2GB RAM peak, 12 transactions | Session 2: crash after 23 min...

19 // IFC - IMPORT, EXPORT AND NATIVE RECONSTRUCTION

19

IFC - Import, Export and Native Reconstruction

IFC (Industry Foundation Classes) is the open format for BIM interoperability. Pulse offers three tool groups: import/linking, configurable export, and native reconstruction - the most advanced workflow that converts IFC elements into editable native Revit elements.

Verification and diagnostics

| ifc_get_capabilities

Check IFC support for this Revit installation: supported versions, revit-ifc add-in.

WHEN TO USE

Before working with IFC, to know what is available.

EXAMPLE 1

```
> Which IFC versions does this Revit installation support?
```

```
| IFC 2x3, IFC 4, IFC 4x3 supported | revit-ifc add-in: v24.1.0 | Schema: ifcXML supported
```

EXAMPLE 2

```
> Does Revit support IFC 4.3?
```

```
| IFC 4x3 supported (add-in v24.1.0). Export and import available.
```

| ifc_validate_request

Validate an IFC file before importing: checks path, extension and schema version.

WHEN TO USE

Before importing or linking an IFC file to catch errors early.

EXAMPLE 1

```
> Check if file C:/Models/Structure.ifc is valid
```

```
| OK: file found, .ifc extension valid, IFC 2x3 schema detected
```

EXAMPLE 2

```
> Verify the IFC file before linking it to the project
```

```
| OK: IFC 4 schema, 45MB size, valid structure
```

Import and linking

ifc_link

Link an IFC file to the project as an external link (stays updatable).

WHEN TO USE

When you want to reference the IFC model without embedding it in the project.

EXAMPLE 1

```
> Link the structural IFC model from C:/Models/Structure.ifc
```

IFC link added: 'Structure.ifc', 1,245 elements, shared coordinate system

EXAMPLE 2

```
> Add the MEP IFC file as a link
```

IFC link 'MEP.ifc' linked: 3,102 elements, origin position

ifc_reload_link

Reload an existing IFC link, optionally from a new path.

WHEN TO USE

When the IFC file has been updated by the structural or MEP consultant.

EXAMPLE 1

```
> Reload the structural IFC link
```

Link 'Structure.ifc' reloaded: 1,287 elements (+42 vs previous version)

EXAMPLE 2

```
> Reload the MEP link from the new updated path
```

Link reloaded from new path, 3,150 elements updated

ifc_open_or_import

Open an IFC file as a new Revit project or import it into the active document.

WHEN TO USE

When you want to permanently incorporate the IFC model into the project.

EXAMPLE 1

```
> Open the IFC file as a new Revit project
```

File opened as new Revit project, 2,456 elements converted to DirectShape

EXAMPLE 2

```
> Import the IFC file into the current project
```

1,245 DirectShapes imported into the active document

Export

ifc_list_export_configurations

List available IFC export configurations (built-in and custom).

WHEN TO USE

Before exporting, to choose the right configuration.

EXAMPLE

```
> What IFC configurations can I use for export?
```

6 configurations: IFC2x3 Coordination View, IFC4 Reference View, IFC4 Design Transfer, GSA (2010), COBie 2.4, Custom_Arch

ifc_get_export_configuration

Show full details of a specific IFC export configuration.

WHEN TO USE

To verify a configuration's parameters before using it.

EXAMPLE

```
> Show details of the 'IFC4 Design Transfer' configuration
```

Schema: IFC4 | Classification: OmniClass | Exports: walls, doors, windows, spaces | Include quantities: yes

ifc_export_basic

Export the model to IFC with basic options: schema version, output path.

WHEN TO USE

For quick export without complex configurations.

EXAMPLE 1

```
> Export the model as IFC 4 to folder C:/Deliverables
```

Export complete: Building_A.ifc (45MB), IFC4 schema, 3,456 elements

EXAMPLE 2

```
> Export as IFC 2x3 for compatibility with legacy systems
```

IFC 2x3 export complete: 3,201 elements, compatible format

ifc_export_with_configuration

Export using a named configuration with the option of specific overrides.

WHEN TO USE

For professional export with controlled parameters.

EXAMPLE 1

```
> Export using the 'IFC4 Design Transfer' configuration with OmniClass classification
```

Export with config 'IFC4 Design Transfer': 3,456 elements, 156 spaces, OmniClass applied

EXAMPLE 2

```
> Export only Ground Floor with the COBie 2.4 configuration
```

COBie export complete: 245 Ground Floor elements, COBie sheet generated

ifc_set_family_mapping_file

Set a mapping file to associate Revit families with the correct IFC types.

WHEN TO USE

To ensure families are exported with the appropriate IFC type.

EXAMPLE

```
> Set the family mapping file from C:/Config/FamilyMapping.txt
```

Mapping file set: 45 family->IFC type association rules loaded

Native reconstruction (IFC -> native Revit)

When you import an IFC file, Revit creates DirectShapes - non-editable geometric objects. The reconstruction workflow analyses these elements and converts them into native Revit elements (walls, floors, beams, doors) that are fully editable.

TIP — Recommended workflow: analyse -> list candidates -> rebuild by category -> cut openings -> place doors/windows -> compare -> tag non-reconstructable.

ifc_analyze_rebuildability

Analyse IFC DirectShapes in the model and calculate native reconstruction feasibility for each.

WHEN TO USE

First step of the reconstruction workflow: understand how many elements can be converted.

EXAMPLE 1

```
> Analyse the imported IFC model: how many elements can I reconstruct?
```

Analysis complete: 1,245 DirectShapes | Walls: 312 (89% reconstructable) | Floors: 45 (95%) | Beams: 67 (78%) | Non-reconstructable: 123 (10%)

EXAMPLE 2

```
> Check reconstructability for walls only
```

312 walls analysed: 278 reconstructable (confidence >80%), 34 complex (possible geometry loss)

ifc_list_rebuild_candidates

List elements with reconstruction confidence above a specified threshold.

WHEN TO USE

To decide which elements to reconstruct, starting with the most reliable.

EXAMPLE 1

```
> Show all reconstructable elements with at least 85% confidence
```

847 candidates: 278 walls, 43 floors, 62 beams, 45 columns (min confidence: 85%)

EXAMPLE 2

```
> Which walls can I safely reconstruct?
```

278 walls with confidence >80%: 245 simple rectangular, 33 with irregular profile

ifc_rebuild_walls

Rebuild walls from IFC DirectShapes as native Revit walls with correct thickness and type.

WHEN TO USE

After analysis, to convert IFC walls into editable Revit walls.

EXAMPLE 1

```
> Rebuild all walls from the imported IFC
```

278 walls rebuilt as native Revit walls: 'Basic Wall 200mm' (142), 'Basic Wall 300mm' (89), others (47)

EXAMPLE 2

```
> Rebuild only ground floor walls
```

68 ground floor walls rebuilt, base level set to 'Ground Floor'

ifc_rebuild_floors

Rebuild floors from IFC DirectShapes as native Revit floors.

WHEN TO USE

To convert IFC floors into editable elements with compound structure.

EXAMPLE 1

```
> Rebuild floors from the IFC model
```

43 floors rebuilt: type 'Generic Floor 200mm', total area 4,560 sqm

EXAMPLE 2

```
> Rebuild slabs with the architectural floor type
```

43 floors rebuilt with type 'Architectural Floor 150mm'

ifc_rebuild_roofs

Rebuild roofs from IFC DirectShapes as native Revit roofs.

WHEN TO USE

To convert the IFC roof into an editable element with slopes.

EXAMPLE 1

```
> Rebuild the roof from the IFC file
```

5 roof panels rebuilt as native Revit roof, slopes calculated automatically

EXAMPLE 2

```
> Rebuild the flat roof
```

1 flat roof rebuilt: type 'Flat Roof 300mm', area 980 sqm

ifc_rebuild_structural_members

Rebuild columns and beams from IFC DirectShapes as native Revit structural elements.

WHEN TO USE

To convert the IFC structural frame into analysable and editable elements.

EXAMPLE 1

> Rebuild all structural framing from IFC

Complete: 45 HEB240/300 columns rebuilt, 67 IPE300/400 beams rebuilt

EXAMPLE 2

> Rebuild only Level 2 beams

22 Level 2 beams rebuilt: HEB300 (12), IPE400 (10), average span 7.2m

ifc_rebuild_openings

Cut openings in rebuilt walls and floors, matching the original IFC openings.

WHEN TO USE

After rebuilding walls and floors, to create the correct openings.

EXAMPLE 1

> Cut openings in rebuilt walls

234 openings cut: 156 for doors, 78 for windows

EXAMPLE 2

> Also create openings in rebuilt floors

12 openings for stairwells and services cut in floors

ifc_rebuild_family_instances

Place doors and windows in rebuilt openings, using native Revit families.

WHEN TO USE

Last reconstruction step: populate openings with native elements.

EXAMPLE 1

> Place doors and windows in the rebuilt openings

156 doors placed (M_Single-Flush 900x2100), 78 windows placed (Fixed 1200x1500)

EXAMPLE 2

> Place doors using 'Fire Door EI60' family for fire-rated openings

45 fire doors placed, 111 standard doors

ifc_compare_original_vs_rebuilt

Compare volume and geometry between original IFC elements and rebuilt Revit elements.

WHEN TO USE

To verify reconstruction fidelity before deleting the DirectShapes.

EXAMPLE 1

```
> Compare rebuilt elements with original IFC
```

Average fidelity: 97.3% | Walls: 98.1% | Floors: 96.8% | 12 elements with >5% deviation flagged

EXAMPLE 2

```
> Verify fidelity of rebuilt walls
```

278 walls compared: 265 (95%) fidelity >95%, 13 with geometric divergence to check

ifc_tag_unreconstructable_elements

Tag IFC elements that cannot be rebuilt as native (geometry too complex).

WHEN TO USE

To document which elements remain as DirectShapes and must be handled manually.

EXAMPLE 1

```
> Tag all elements I could not reconstruct
```

123 elements tagged as 'IFC_Non_Reconstructable': 45 stairs, 34 ramps, 44 complex geometries

EXAMPLE 2

```
> Add a 'Reconstruction Failure Reason' parameter to non-rebuilt elements
```

123 elements: parameter added with reason (stair, curved geometry, etc.)

20 // POWER BI - PREMIUM

20

Power BI - Premium

Power BI integration is available exclusively in Pulse Premium. It publishes elements, schedules, and selections to push datasets and can bring a filtered report selection back into Revit. The commands remain blocked by the Keygen gate while Pulse is in Free mode.

The custom visual ships in the package, in the powerbi-visual folder, and is installed BY HAND in under a minute -- the installer does not install it. Recommended way: in Power BI Desktop open Visualizations > "..." > Import a visual from a file and point to the package's .pbviz; Power BI itself puts it where it belongs. Alternatively copy the .pbviz into Documents > Power BI Desktop > Custom Visuals and restart Power BI Desktop.

With OneDrive the Documents folder may live under OneDrive rather than in C:\Users\<you>; if you are unsure of the path, use the import from Power BI.

The visual follows the Power BI Desktop language: with Power BI in Italian its interface is Italian, with any other language it is English. If the Power BI version does not report its language, the visual falls back to the Revit language.

| pbi_check_auth / pbi_list_workspaces

Check Microsoft sign-in and discover available workspaces and datasets.

WHEN TO USE

Before creating a new Power BI binding.

EXAMPLE

```
> Sign in to Power BI and show my workspaces
```

| Sign-in complete. Workspaces: Coordination, Tower Project, Sandbox.

| pbi_publish_elements / pbi_publish_schedules

Publish verified element or schedule snapshots to the standard Pulse tables.

WHEN TO USE

To feed Power BI reports with current BIM data.

EXAMPLE

```
> Publish walls and columns to the Tower Project dataset
```

| Snapshot published: 1,284 elements; metadata updated; binding saved.

pbi_query

Query the bound dataset and select or isolate matching ElementIds in Revit.

WHEN TO USE

To close the Power BI -> Revit review loop.

EXAMPLE

```
> Isolate in Revit the Level 2 elements filtered in Power BI
```

42 elements isolated in the current view.

21 // DYNAMO INTEGRATION - PREMIUM

21

Dynamo Integration - Premium

Dynamo Integration is available exclusively in Pulse Premium and remains disabled by default. Enable it under Settings > General after confirming that a compatible Dynamo for Revit is installed. In Free mode the integration remains blocked by the Keygen gate.

dynamo_get_status / dynamo_list_graph_io

Check Dynamo availability and inspect graph inputs/outputs without running the graph.

WHEN TO USE

Before generating or executing any graph.

EXAMPLE

```
> Check Dynamo and list the inputs of Doors.dyn
```

Dynamo available. Inputs: LevelName, FamilyName, DryRun.

dynamo_generate_graph

Generate a .dyn file from a controlled specification, with Python sandbox validation.

WHEN TO USE

To create repeatable Dynamo automation without wiring every node manually.

EXAMPLE

```
> Generate a graph that reads walls, type, and level
```

Graph generated and saved as WallInventory.dyn.

dynamo_run_graph

Run an existing graph with explicit input values.

WHEN TO USE

After verifying inputs, file, and a safe test model.

EXAMPLE

```
> Run Doors.dyn with LevelName = Level 1 and DryRun = true
```

Preview run completed; no changes applied.

Appendix - Prompt Library

Ready-to-use prompts organised by scenario. Copy, adapt to your project and paste into Claude.

ORIENTING IN A NEW MODEL	
Full overview	"What is this project? Show me levels, worksets, phases and linked files."
Quick stats	"How many elements are there by category? Show me the top 10 categories."
Active view	"Which view am I in? How many elements are visible?"
Model health	"Do a quick model health check and show me the 5 most important warnings."
FINDING AND FILTERING ELEMENTS	
By category	"Find all [structural columns] in the model, show me type and level."
By parameter	"Find all doors where the 'Fire Rating' field is empty."
By value	"Which walls on Level 1 have thickness greater than 250mm?"
In active view	"List all visible walls in this view with type and length."
Selected element	"What do I have selected? Show me all parameters."
MODIFYING PARAMETERS	
Single element	"Set the Mark of door 12345 to 'D-001'."
Bulk (safe)	"How many doors would be modified if I set Manufacturer to 'ACME'? (preview)"
Bulk (execute)	"OK, proceed: set Manufacturer 'ACME' for all doors."
From Excel	"Import the parameter values from Excel file 'DoorData.xlsx' (show me preview first)."
Rename	"Rename all views replacing 'Copy of ' with " (remove the prefix)."
Number rooms	"Number all rooms starting from 101, ordered left to right."
CREATING ELEMENTS	
Wall	"Create a 'Basic Wall 200mm' wall from point (0,0) to point (8,0) at Ground Floor."
Floor	"Create an architectural floor in room 205 with type 'Concrete 150mm'."
Door	"Place a 'M_Single-Flush 900x2100' door in wall 5678 at 1.5m from the left end."
Room	"Create a room 'Director's Office' number 101 on Level 1 at coordinates (12, 8)."
Level	"Create Level 3 at elevation 10.5 metres."
Grid	"Create a 4x3 grid with 6m spacing, axes A-D and 1-3."
VIEWS AND SHEETS	
Floor plan	"Create a floor plan of Level 2 at scale 1:100."
Section	"Create a section cutting the building from west to east at Ground Floor."
Isolated 3D	"Create a 3D view with section box around the selected elements."
Batch sheets	"Create sheets A-101, A-102, A-103 with 'Company Titleblock'."
Place view	"Put the Level 1 plan on sheet A-101, centred."
Views for rooms	"Create a section for every room on Level 1."
Export DWG	"Export all sheets to DWG in folder C:/Deliverables."

SCHEDULES AND DATA

Quick schedule	"Create a door schedule with Type Name, Level and Fire Rating."
Room schedule	"Create a room list with name, number, area and department."
Export CSV	"Export the door schedule as a CSV file."
Export Excel	"Export all door data to an Excel file on the desktop."
Read schedule	"Show me the first 20 rows of the 'Door Schedule'."

QUALITY AND AUDIT

Morning check	"What is the model health? Score and main issues."
Warnings	"Show me the 10 most frequent warnings in the model."
Clash detection	"Check clashes between beams and ducts. Create a 3D view for the clashes found."
Heavy families	"What are the 10 most used families? Are there any unused families?"
Pre-delivery	"Run a full audit: health, critical warnings, unused families, heavy views."

MATERIALS AND COMPOUND STRUCTURES

Material list	"List all materials in the model grouped by class."
Compound structure	"Show the layer composition of wall 'External Wall 300mm'."
Create material	"Create a material 'Silk-screen Glass' with 30% transparency and light grey colour."
Quantities	"How many sqm of concrete are used in the model?"

IFC - IMPORT AND EXPORT

Check support	"Which IFC versions does this Revit installation support?"
Validate file	"Check if file C:/Deliverables/Structure.ifc is valid before linking it."
Link IFC	"Link the structural IFC model from file C:/Models/Structure.ifc."
Export IFC4	"Export the model as IFC 4 to folder C:/Deliverables."
Config export	"Export using the 'IFC4 Design Transfer' configuration."

IFC - NATIVE RECONSTRUCTION

Analysis	"Analyse the imported IFC elements: how many can I rebuild as native Revit elements?"
Candidates	"List all elements with reconstruction confidence above 85%."
Rebuild walls	"Rebuild all walls from the imported IFC as native Revit walls."
Rebuild all	"Rebuild walls, floors, beams and columns. Then cut openings and place doors and windows."
Compare	"Compare the geometry of rebuilt elements with the original IFC. What is the average fidelity?"
Tag residuals	"Tag the IFC elements that cannot be rebuilt as native."

CLEANUP AND PROJECT CLOSE

Purge	"What can I delete from the model? Unused families, types and materials. (preview first)"
Stale CAD	"List all CAD imports in the model. Then delete those no longer needed."
Orphan views	"Show me views not placed on any sheet. Delete the working ones?"
Empty tags	"Remove all empty tags from the current view."
Worksets	"Move all doors to workset 'Architecture - Doors'."

CUSTOM C# SCRIPT (Revit 2025+)

Batch operation	"Add the 'STD-' prefix to all 'Generic' families."
Complex type	"Write C# code to create floor type 'FL_001' with 3 layers: ceramic 10mm, screed 60mm, concrete 150mm."
Advanced read	"Calculate total wall area by level and return a table."