ArchPlug

DETAILED INSTRUCTIONS

Automatic Room Naming and Scheduling

Advanced room management and intelligent tagging for multi-family apartment projects.

For Autodesk Revit 2022–2026 · Windows

archplug.com · July 2026

Advanced room management and intelligent tagging for multi-family apartment projects — Autodesk Revit.

Overview

Automatic Room Naming and Scheduling (shown in Revit as **ArchPlug Rooms**) turns the most repetitive part of multi-family apartment documentation — placing rooms, naming them, grouping them into apartments, and building per-level area schedules — into a guided, four-step wizard.

On a typical apartment building, room documentation means hours of clicking: placing a room in every enclosed space on every level, typing "Bedroom", "Bathroom", "Kitchen" hundreds of times, numbering apartment entrances, keeping room numbers consistent within each apartment, and assembling net/circulation/gross area schedules level by level. ArchPlug Rooms automates the entire chain:

- **Places rooms and room tags** automatically on every selected level.
- **Names circulation** (stair core and adjacent hallways) from the stair geometry itself.
- **Numbers apartment entrance doors** with a dedicated entrance tag, ordered clockwise or counter-clockwise around each floor.
- **Detects which rooms belong to which apartment** by walking the plan through doors and room separation lines (a breadth-first search seeded at each entrance door), writes an `ApartmentID` to every room, and renumbers rooms 1, 2, 3… within each apartment.
- **Names rooms from their furniture**: a room containing a sofa becomes "Living Room", a room with a bed becomes "Bedroom" — driven by a rules table you control.
- **Builds live Revit schedules** (net areas grouped per apartment, circulation areas, gross areas, parking tables) and drops them on your sheets, plus per-level gross **area plans** with automatically traced boundary lines around the building footprint.

Everything it creates is native Revit data — real rooms, real tags, real schedules, real area plans — so the output stays live and updates with the model. The wizard window is modeless: it floats above Revit and you can keep working in the model while it is open.

Key Benefits & Use Cases

Who it's for: architects and BIM technicians documenting residential buildings — especially multi-family / multi-storey apartment projects where the same room structure repeats across many units and levels.

Concrete scenarios:

- **Apartment building documentation.** A 6-storey building with 12 apartments per floor: place rooms on all levels with one click, tag them, name the stair core and hallways per level, number all entrance doors around each floor plate, then let the plugin flood-fill each apartment from its entrance door — every room receives its apartment number and a sequential room number, and rooms are named from the furniture placed in them.
- **Area statements for permits and sales.** Per-level "Net areas" schedules grouped by apartment with subtotals per apartment and a grand total; "Circulation" schedules for common areas; per-level gross-area plans with auto-traced boundary lines and a gross-area schedule reading from them.
- **Underground garage levels.** A separate Garage Naming tool names garage bays from the parking families placed in them, names stair cores, fire-barrier antechambers and installation shafts next to the stairs, and even names completely empty rooms — with its own level selection so it never touches the residential floors.
- **Parking documentation.** Parking tables that count parking spots per level and sum their clean net area (including the extra access strip of accessible/disabled spots), as a native Revit schedule.
- **Sales-oriented door tags.** Optionally write each apartment's total area (in m^2) straight into its entrance door tag.

Why it saves time: each of these tasks is otherwise manual, error-prone and must be redone after every plan change. Because the plugin reads the model geometry (stairs, doors, separation lines, furniture) instead of relying on manual selection, re-running a step after a design change simply refreshes the result.

Requirements

- **Autodesk Revit 2022–2026** (Windows). Revit is Windows-only, and so is the plugin.
- The plugin is part of the **ArchPlug suite** of Revit productivity tools (archplug.com). It shares the ArchPlug ribbon tab and visual theme with the other ArchPlug plugins.
- A model with **bounding walls / room separation lines** (rooms must be enclosable), **doors with the Mark parameter filled**, and — for furniture-based naming — furniture, plumbing fixture or specialty equipment families placed in the rooms.
- The entrance-door tag is a Generic Annotation family (`ArchPlug_EDT_EntranceDoorTag`) that ships with the plugin and is loaded into your project automatically the first time it is needed.

Installation

1. Run the supplied **.msi installer** and follow the prompts.

2. Start (or restart) Revit and open a project.
3. Find the **ArchPlug** ribbon tab and launch the Rooms tool. The wizard window **"ArchPlug Rooms - Wizard"** opens, floating on top of Revit.

The window is modeless and always-on-top: you can pan, zoom and edit the model while it is open. Press **Esc** to close it (if a child window such as Level Settings is open, the first Esc closes that child; the next Esc closes the wizard). All fields, the window position and size are remembered between sessions.

User Interface Walkthrough

Main window — "ArchPlug Rooms - Wizard"

The window contains four step groups, top to bottom, plus a framed 2×2 button area at the bottom.

Step 1 - Room Placement

Master scope for the whole plugin, plus automatic room/tag placement.

Control	What it does
Current level (radio)	Every step (placement, stairs, tags, naming) works only on the active view's level.
All levels (selected in level settings) (radio)	Every step works on the levels you tick in Level Settings. At least one level must be ticked, otherwise all action buttons are disabled.
Level Settings > (button)	Opens the Level Settings side panel (see below). Always clickable; its selection only takes effect in "All levels" mode.
(N levels selected) (label)	Live count of levels ticked in Level Settings.
Place rooms (button)	Places Revit rooms automatically in every enclosed region (equivalent of Revit's "Place Rooms Automatically") on the target levels.
Place tags (button)	Places a room tag on every room that is not yet tagged, in each target floor-plan view. Existing tags are never duplicated.

Step 2 - Stairs / Hallway Naming

Control	What it does
Stairs room name: (text)	Name template for rooms inside the stair core (e.g. "Stairs").
Hallway room name: (text)	Name template for hallways adjacent to the stair core — rooms that share a room separation line with a Stairs room.
Room comment: (text)	Value written to the Comments parameter of every renamed Stairs and Hallway room (default "Common"). This comment is what the Circulation schedule later filters on.
Append level prefix to name (checkbox)	ON: names become "Stairs Level 1", "Hallway Level 2", etc. OFF: the bare name is used on every level.
Run this step (button)	Runs the stairs/hallway naming on the target levels.

Step 3 - Entrance Door Tagging

This section unlocks as soon as rooms exist in the model (Step 1 completed). Step 2 is optional — you can tag entrance doors without naming stairs and hallways first.

Control	What it does
Clockwise (radio)	Number entrance-door tags going clockwise around each level's center.
Counter-clockwise (radio)	Number them counter-clockwise.
Hide door tags after creation (checkbox)	After creation, the tags are hidden in their owning views. They stay in the model (Step 4 still finds them) but won't print.
Entrance door comment: (text)	The exact text a door's Comments parameter must equal (case-insensitive) for the door to count as an apartment entrance door. Default: "entrance door". Only such doors get tagged here and act as apartment seeds in Step 4.
Run this step (button)	Deletes previously created entrance tags (in the targeted views) and re-creates them with fresh numbering.

Step 4 - Room Naming

Unlocks as soon as rooms exist in the model (Step 1 completed). Steps 2 and 3 are optional: if you have run Step 3, rooms are grouped into apartments and numbered per apartment; if no entrance-door tags exist, Step 4 still assigns room names from your rules table — rooms simply get names only, without apartment grouping and apartment numbering.

Left side — the naming rules table:

Control	What it does
Family in room (dropdown column)	A furniture / plumbing-fixture / specialty-equipment family (or type) name from your project. Free text is also accepted; matching is case-insensitive "contains".
Room name to assign (text column)	The room name written when a matching family is found inside the room.
▲ Up / ▼ Down (buttons)	Reorder rules. Rule order is priority: the first matching rule (top-down) wins.
Add row / Delete row (buttons)	Manage rules.

Right side — options:

Control	What it does
Foyer auto-detect (checkbox)	The first room reached from an apartment entrance door that contains no recognized furniture family is treated as the foyer.
Foyer name: (text)	Custom name written to the detected foyer. Empty falls back to "Foyer". Dimmed while Foyer auto-detect is off.
Terrace via comment: (checkbox)	Doors whose Comments contain the text below mark the room behind them as a terrace.
(terrace comment text field)	Text searched for inside a door's Comments to identify it as a terrace door (default "terrace door").
Terrace name: (text)	Custom name written to terrace rooms (default "Terrace"; type your own, e.g. "Terasa"). Both terrace fields dim when the checkbox is off.
Preserve names ending with: (checkbox)	Rooms whose name ends with the suffix below are never renamed — protection for manually locked names.
(suffix field)	The lock suffix, default "`.`" — e.g. a room named "My Studio." is preserved as-is.
Add apartment area to door tag (checkbox)	Writes each apartment's total area (sum of its rooms, in m^2) into the entrance door tag's apartment-area parameter.
Run this step (button)	Runs apartment detection, room naming, ApartmentID writing and per-apartment renumbering.

Bottom button frame (2×2)

Button	What it does
Save Template	Saves all current settings (every field in every step) to a JSON file. Default folder: `Documents\ArchPlug\Templates\`.
Import Template	Loads settings from a previously saved JSON template.
Area Plans	Opens the Area Plans window (per-level gross area plans).
Place tables	Opens the Tables window (net/circulation/gross schedules + parking tables).
Garage Naming >	Opens the independent Garage Naming window.

There is no Close button — close with the window's X or Esc; every field is saved automatically on close.

Level Settings panel — "ArchPlug Rooms - Views"

A non-modal side panel that docks to the right of the wizard.

- Header: **"Tick floor-plan views (Shift+click for range)"**. The list (column **"Floor plan view"**) shows every non-template *Architectural* floor-plan view tied to a level — MEP/structural/site plans are filtered out. Shift+click range-toggles all items between the last clicked and the current one.
- **Current view / Checked views** (checkboxes, mutually exclusive) — the scope for the two buttons below.
- **Hide all rooms / Unhide all rooms** (buttons) — hide or unhide every room in the scoped views (useful for checking or printing plans without room fills).
- **Check all / Clear** (buttons) — select or deselect all views.
- **Close** (button).

View granularity matters for Step 3 (door tags are view-specific and are placed in exactly the ticked views); Steps 2 and 4 work per level, so ticking any one view of a level is enough.

Tables window — "ArchPlug Rooms - Tables (target levels)"

Top panel — **"Levels selected in Level Settings:"** — one row per ticked level with three buttons:

- **Place net areas** — creates (or reuses) a schedule named `Net areas – <Level>`: room number, room name, area; only rooms that have an ApartmentID; grouped by apartment with per-apartment subtotal footers and a grand total; filtered to that level.
- **Place circulation** — creates `Circulation – <Level>`: rooms carrying the Step 2 room comment (e.g. "Common"), with a grand total, filtered to that level.

- **Place gross** — creates `Gross area – <Level>`: an Areas schedule (gross area scheme) listing the gross area element the plugin places on that level's area plan, with a total.

Bottom panel — **"Parking tables — broj mesta + neto površina po spratu:"**:

- A grid with one column, **"Parking type (click to choose)"** — a dropdown of parking-family type names found in the project. **Add row / Delete row** manage the list.
- **Place parking tables** — stamps every matching parking instance with its computed net area and level label, then creates/updates the schedule **"Parking mesta"** (grouped per level, with piece count and summed area) and drops it on the active sheet. Re-running re-stamps areas without duplicating the table.
- **Close**.

All Place buttons require the **active view to be a sheet** — the table is dropped at the center of the visible sheet area. If a table is already placed, the plugin tells you which sheet it is on instead of duplicating it.

Area Plans window — "ArchPlug Rooms - Area Plans (target levels)"

Header **"Levels selected in Level Settings:"**, one row per ticked level with two buttons:

- **Place area plan** — creates (or reuses) a gross area plan view named `Area Plan – <Level>` and places it as a viewport centered on the active sheet. Requires an Area Scheme in the project (the "Gross Building" scheme is preferred; every Revit project ships with one).
- **Update lines** — traces the building's exterior footprint on that level (from the wall solids, with floors and columns as backup) and redraws the area boundary lines on the area plan. Existing boundary lines on the view are cleared first, so this is a true refresh. A gross Area element with an area tag is placed inside the outline (only if none exists yet), which feeds the "Place gross" schedule.

Bottom: **Update all** — runs Update lines for every ticked level and reports how many succeeded — and **Close**.

Garage Naming window — "ArchPlug Rooms - Garage Naming"

An independent tool with its own level selection and its own rules.

- **Level Settings >** (button, top-right) — opens its own view list ("ArchPlug Rooms - Garage Levels"): tick only the garage levels. This selection is separate from the main plugin's Level Settings. A label shows "(N garage levels selected)".
- Rules grid with three columns:
- **"Parking family in room (click to search)"** — clicking the cell opens the **"Search family"** popup: type to filter all family/type names in the project live, pick with double-click or Enter; the first entry **"\<none>"** clears the cell. (Unlike Step 4, this list covers every category, so any parking or generic-model family shows up.)
- **"Room name to assign"** — the name written to rooms containing that family.

- **"Comment?"** (checkbox) — rooms named by this rule also receive the custom comment below.
- **▲ Up / ▼ Down / Add row / Delete row** — manage the rules.
- **Custom comment:** (text) — written to the Comments parameter of every room named by a rule row with "Comment?" ticked (e.g. "komunikacije"); it is also stamped on stairs, antechamber, shaft and empty rooms so they all land in the circulation-style schedule.
- **Name stairs rooms** (checkbox) — rooms whose footprint contains a stair get the *main window's* stairs name (with level prefix, if enabled in Step 2).
- **Antechamber by stairs** (checkbox + name field) — an empty room connected by a door to a stairs room, with area **at or above** the threshold, is named as a fire-barrier antechamber.
- **Install. shaft by stairs** (checkbox + name field + **max** *area* **m²**) — an empty room connected by a door to a stairs room, with area **below** the threshold (default 2 m²), is named as an installation shaft.
- **Name completely empty rooms** (checkbox + name field) — any remaining room on the garage levels with no real content is renamed (e.g. an empty garage bay). Rooms holding only structural columns, framing, foundations, doors, windows, annotations or detail items still count as "empty".
- **Run garage naming** / **Close**.

Rule priority within Garage Naming: family match → stairs → installation shaft → antechamber → empty rooms.

Step-by-Step Workflow

A complete run on a multi-family project:

1. **Open a floor plan** of the building and launch ArchPlug Rooms from the ArchPlug ribbon tab.
2. **Choose the scope** in *Step 1 - Room Placement*: keep **Current level** for a single floor, or select **All levels**, click **Level Settings >** and tick the floor-plan views of every residential level (Shift+click for ranges).
3. Click **Place rooms** — rooms are created in every enclosed region on the target levels. Then click **Place tags** to tag every untagged room in the target views.
4. **Prepare your entrance doors** (one-time modeling convention): each apartment's entrance door needs (a) its **Comments** parameter set to the text in "Entrance door comment:" (default `entrance door`), and (b) a filled **Mark** value.
5. In *Step 2 - Stairs / Hallway Naming*, set **Stairs room name**, **Hallway room name** and **Room comment** (or keep the defaults Stairs / Hallway / Common), decide on **Append level prefix to name**, and click **Run this step**. The plugin finds the rooms over the stair

cores, renames them, renames the adjacent hallways (rooms sharing a room separation line with a stairs room), writes the comment, and numbers them 1 (stairs) and 2 (hallway).

6. (Optional) In *Step 3 - Entrance Door Tagging*, pick **Clockwise** or **Counter-clockwise** and click **Run this step**. Every entrance door receives an entrance tag next to it, rotated to match the door orientation, numbered continuously level by level (lowest level first) around each floor's center starting from the top of the plan. The tag also records the door's Mark.
7. In *Step 4 - Room Naming*, build your rules table — e.g. `Sofa → Living Room`, `Bed → Bedroom`, `Toilet → Bathroom`, `Sink → Kitchen` — and set the options (Foyer auto-detect, Terrace via comment, Preserve names ending with, Add apartment area to door tag). Click **Run this step**. For each entrance door the plugin walks the apartment through doors and separation lines (stopping at circulation rooms), then: - writes the entrance number into every room's **ApartmentID** parameter (the shared parameter is created automatically in a fresh project), - renames rooms by the first matching furniture rule, the detected foyer, and terraces, - renumbers rooms **1..N within each apartment** (entrance/foyer first, terraces last), - optionally writes the apartment's total m² into the entrance door tag.
8. **Schedules:** open a sheet, click **Place tables**, and drop **Place net areas** / **Place circulation** per level. For gross areas, first open **Area Plans**, click **Update lines** (or **Update all**) to trace the footprint and create the gross Area, then place the area plan and the **Place gross** table on your sheets.
9. **Garage levels:** open **Garage Naming >**, tick the garage levels in its own Level Settings, set the parking-family rule plus the stairs/antechamber/shaft/empty options, and click **Run garage naming**. Back in the Tables window, add your parking types and click **Place parking tables** on a sheet.
10. Re-run any step after design changes — steps refresh their own elements instead of duplicating them.

Feature Reference

Master level mode

One switch (Step 1) governs the whole plugin: *Current level* works on the active view's level; *All levels* works on the levels of the views ticked in Level Settings. In "All levels" mode with nothing ticked, all Run/Place buttons are disabled until you tick at least one view. Step 3 places tags in exactly the ticked views (same door = same number in every view of a level); Steps 2 and 4 process one view per level.

Room placement and tagging

"Place rooms" calls Revit's automatic room placement per level (every enclosed region gets a room). "Place tags" tags only rooms that are not yet tagged in that view, at each room's location point — safe to re-run.

Stairs & hallway naming (Step 2)

Stairs rooms are found geometrically: the plugin takes the stair elements visible in the active view, and marks any room (on any target level) that contains a stair footprint center — with a bounding-box fallback for irregular ground-floor stair rooms. Hallways are rooms on the same level that share a room separation line with a stairs room. Renamed rooms get: the name (with optional level suffix), the Step 2 comment, and room number 1 (stairs) or 2 (hallway). Names are model data, so they apply across all views.

Entrance door tagging (Step 3)

- A door is an entrance door when its Comments **equals** the configured text (case-insensitive).
- Tags are instances of the shipped Generic Annotation family; if the family is not in the project it is loaded automatically from the plugin's family folder.
- Numbering is continuous across levels (sorted by elevation) and ordered around each level's door centroid, clockwise or counter-clockwise, starting from the top of the plan.
- The tag is offset 60 cm from the door toward the plan-consistent side and rotated to match the door's facing; it stores the sequence number and the door's Mark (which is the link Step 4 uses).
- Re-running the step deletes the previously created entrance tags first — scoped to the selected views when a view selection is active — so numbering is always clean, never duplicated.
- With "Hide door tags after creation" the tags are hidden in their views but remain in the model, so apartment detection still works and step detection still registers them.

Apartment detection (Step 4)

- **Seeding:** each entrance door starts a breadth-first search on the interior side of the door.
- **Adjacency:** two rooms are connected if a door joins them, or if they physically flank the same room separation line (tested by offsetting perpendicular from the line's midpoint — a deliberately conservative rule that prevents a long shared separation line from "bridging" two neighbouring apartments).
- **Boundaries:** the search never crosses circulation rooms (your Stairs/Hallway names, plus common circulation keywords).
- **Arbitration:** if a room is reachable from several entrances (e.g. a continuous shared terrace slab), it is awarded to the apartment whose entrance is physically closest.
- **Terrace pinning:** a terrace is assigned to the apartment whose interior door opens onto it — top priority — and is always numbered after the interior rooms.
- **Result:** every room in an apartment gets the entrance tag's number written into its **ApartmentID** parameter, and rooms are renumbered 1..N per apartment in walk order (entrance/foyer first, terraces last). Revit's "duplicate room number" warnings are suppressed automatically, since repeating 1..N in every apartment is intentional.

Naming rules (Step 4)

Priority per room: **family-rule match** → **foyer** → **terrace**. Family matching scans Furniture, Plumbing Fixtures and Specialty Equipment placed in the view, matches the rule text case-insensitively against the element's family/type names, and applies the first matching rule in table order. Circulation rooms are left alone (Step 2 owns them). Rooms whose name ends with the protected suffix are never touched. The foyer is the first room behind the entrance door with no recognized furniture. Terrace rooms are identified through terrace-door comments.

ApartmentID shared parameter

Created automatically on first use (a Text instance parameter bound to Rooms, with a fixed GUID so it is identical across all your projects). It powers the per-apartment grouping in the Net areas schedule and is available for your own schedules, filters and view color schemes.

Apartment area on the door tag

When enabled, the plugin sums each apartment's room areas (circulation excluded), converts to m², and writes the value into the entrance tag's apartment-area parameter (it looks for a suitable writable parameter such as "Površina stana" / "Apartment Area" / "Area"; text parameters receive a formatted value like `54.20 m²`).

Schedules ("Place tables")

All schedules are built from scratch in code — no template needed, they work in a fresh project — and are live Revit schedules that keep updating with the model. Naming convention: `<Base> – <Level name>` (e.g. `Net areas – Level 2`), so per-level tables are recognized and reused instead of duplicated. Net-areas and circulation tables dropped at the same time are automatically offset 3 cm apart on the sheet.

Gross area plans ("Area Plans")

"Update lines" shrink-wraps the building's exterior on the level: it unions the wall solids (adding floor slabs and columns as needed), extracts the outer loop, and draws area boundary lines on the level's gross area plan. If the exterior wall ring has small gaps (1–2 cm) that would make the outline "snake" around interior walls, a gap-bridging pass closes them automatically. A gross Area element plus an area tag are placed inside the outline if none exists yet. "Update" always clears the previous lines first, so the outline follows design changes.

Parking tables

Parking families are typically face-hosted (they carry no Level) and their real footprint needs computing, so the plugin stamps two instance parameters on each selected parking spot: the computed spot area (width × depth, **plus a 1.5 m wide access passage across the full depth for each ticked accessible side** of the family) and the level label derived from the spot's elevation. The "Parking mesta" schedule then groups by level and type, showing piece counts

("Komada") and summed areas ("Površina") with subtotals and a grand total. Re-running re-stamps the values — edit-safe.

Garage Naming

Fully independent mechanism with its own level list, its own rules grid (searching all family categories, including Parking and Generic Models), the stairs/antechamber/installation-shaft logic described above, empty-room naming, and a shared custom comment for consistent scheduling. Z-safe room lookup means parking spots hosted on a foundation slab slightly below the room's base are still matched to the right room. If the run renames 0 rooms, a diagnostic dialog explains the likely cause (no rule set, family-name mismatch, or unenclosed rooms) instead of failing silently.

Templates

"Save Template" exports every setting of every step to a JSON file (default folder `Documents\ArchPlug\Templates\`); "Import Template" restores it. Ideal for reusing an office standard (rules table, naming conventions, options) across projects. Window positions are not stored inside templates.

Undo/Redo and general UI behavior

- **Ctrl+Z / Ctrl+Y** (also Ctrl+Shift+Z) undo/redo your **form inputs** in each plugin window — text fields, checkboxes, radio buttons, rules grids, view checklists. This works only while the plugin window has focus, and never touches the Revit model or Revit's own undo stack (model changes are undone in Revit as usual — each step is a single named Revit transaction).
- Steps finish **silently** on success (no confirmation popup); only errors and diagnostics raise dialogs.
- All settings persist automatically on close (X button included). Window position/size are remembered and clamped back on screen if your monitor setup changes; per-monitor DPI differences are compensated.
- The wizard is resizable; the Step 4 layout scales proportionally.

Tips & Best Practices

- **Adopt the two door conventions early:** entrance doors get the exact Comments text (default `entrance door`) and a filled Mark. This single habit unlocks Steps 3 and 4 completely.
- **Model room separation lines deliberately.** Hallways are detected via a *shared* separation line with the stairs room, and apartments connect across separation lines flanking the same line. Avoid one long separation line running along several unrelated rooms.

- **Keep apartments closed.** The BFS travels through doors and separation lines; an unintended opening between two apartments can merge them. The nearest-entrance arbitration mitigates this, but clean boundaries are always better.
- **Order your rules table by decisiveness.** Put the most room-defining family first (e.g. Bed before Sink if a studio contains both) — the first matching rule wins.
- **Protect manual names** with the lock suffix: enable "Preserve names ending with:" and end a room name with `.` (e.g. "Master Suite.") to keep the plugin from ever renaming it.
- **Use view selection, not level guessing.** In "All levels" mode the plugin processes exactly the views you tick — tick the working plan of each level (the one whose furniture you want scanned in Step 4).
- **Run steps in order after big changes.** Renaming stairs (Step 2) redefines the circulation boundary; re-running Step 4 afterwards re-flows apartments correctly.
- **Sheets first for tables and area plans.** Open the destination sheet and zoom where you want the table — it is dropped at the center of your current zoom.
- **Save a template** once your office naming standard is dialed in, and import it at the start of every new project.

Troubleshooting & FAQ

Step 3 and Step 4 look locked. The wizard detects progress from the model, not from a session flag: both steps unlock once rooms exist in the model (Step 1). If they still appear locked, run Step 1 "Place rooms" — or open a floor-plan view of a level that actually has placed rooms.

"No stairs found in the current view." Step 2 reads stair elements from the **active view**. Open a floor plan that shows the staircase (stairs are matched to rooms by their footprint, which is identical in XY on every floor).

"No room found within the stair core." There is no placed room over the staircase. Place rooms first (Step 1 "Place rooms") and make sure the stair space is enclosed.

Step 3 tagged no doors / "Run Step 3 (Entrance Door Tags) first." keeps appearing. Check that entrance doors' Comments **exactly equal** the configured text (case doesn't matter, but extra words do), and that the doors sit on the target levels.

Apartment numbers are missing on some rooms (ApartmentID empty). The room wasn't reached from any entrance door: it may be separated by a room with a circulation name, not connected via a door or separation line, or its entrance door has no Mark (the Mark links door → tag → number).

Two apartments got merged, or a room went to the wrong apartment. Look for an unintended door/separation-line connection between the units. Note that shared terraces are resolved automatically (terrace goes to the apartment whose door opens onto it) and contested rooms go to the nearest entrance.

A room I named manually keeps being renamed. Enable **Preserve names ending with:** and end the name with the suffix (default `.`).

"Open a SHEET first, then click Place." Tables, parking tables and area plans can only be dropped onto a sheet — make a sheet the active view.

"This table is already placed (on sheet '...')." / "This area plan is already placed..." The plugin never duplicates a per-level table or area plan. Delete the existing instance (or use the existing sheet) if you want to move it.

"No Area Scheme found." Gross tables and area plans need an Area Scheme. Create a "Gross Building" area scheme (Architecture > Area > Area Plan) — new Revit projects include one by default.

"Update lines" fails or the outline looks wrong. The footprint is traced from wall solids (with floors/columns as backup). The dialog includes a diagnostic (wall/column/floor solids found, which floor plan was used). Small gaps in the exterior wall ring are bridged automatically; large openings may need closing in the model.

Garage Naming reports "Renamed 0 rooms." The dialog lists the likely cause: no usable rule (pick the family from the search popup rather than typing it), a family-name mismatch, or unenclosed rooms (a room with an open boundary has Area = 0 and can't be matched geometrically — the count of such rooms is shown).

Hidden door tags — will Step 4 still work? Yes. "Hide door tags after creation" hides them visually only; detection and apartment seeding read them regardless.

Does Ctrl+Z in the wizard undo model changes? No — inside plugin windows Ctrl+Z/Ctrl+Y only undo your form inputs. Model changes are undone with Revit's own Undo (each step is one named transaction, e.g. "ArchPlug Rooms - Room Naming").

The window opened oversized or off-screen after changing monitors. The plugin clamps its remembered position back onto the nearest screen and re-applies per-monitor DPI scaling on first show. Simply reopen the window if your monitor layout changed mid-session.

Are the schedules static exports? No. Everything — rooms, tags, ApartmentID, schedules, area plans — is native, live Revit data that updates with the model. Re-running a step refreshes the plugin's own elements without duplicating them.
