

PULSE / Revit

Guida e comandi

Autodesk Revit 2023-2027

Installazione, primi passi ed esempi in linguaggio naturale. I nomi tecnici dei comandi restano invariati.

Fonte verificata: <https://github.com/LuDattilo/pulse-for-revit>

Branch: main | Commit: 2c0407f58c5266eb9466a4bdd2553a05f7da30b1

2026-10-04 | IT

Descrizioni in italiano. Nomi dei comandi, parametri e valori tecnici restano invariati. Catalogo delle sorgenti del 4 ottobre 2026; traduzioni aggiornate l'8 ottobre 2026.

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

Installazione e collegamento

Chiudi Revit, estrai completamente il pacchetto e avvia l'installer EXE ricevuto dal sito. Riavvia Revit, apri un progetto e accendi PULSE dalla scheda dedicata. Usa la configurazione del client AI prevista dal pacchetto.

Collega un client AI compatibile con MCP: Claude, Codex, opencode o un altro client, anche con modelli locali.

Apri il progetto e descrivi l'obiettivo. Verifica il risultato e approva le modifiche quando richiesto.

Lavora su una copia del progetto. Chiedi una lettura o un'anteprima prima delle modifiche, controlla la selezione e approva solo le operazioni attese.

Alla prima registrazione idonea, Trial Premium personale di 14 giorni. Una nuova richiesta non rinnova una trial già attivata.

Installa PULSE, collega il tuo assistente AI (Claude, Codex, opencode o qualsiasi altro client MCP - anche con modelli locali open-weight) e parla al modello Revit in linguaggio naturale. Questa documentazione copre installazione, configurazione, il catalogo completo dei comandi (Free e Premium) ed esempi pratici.

1. Installazione

PULSE è un plugin per Autodesk® Revit® 2023-2027. L'installazione richiede pochi minuti e non serve alcuna competenza tecnica.

Scarica PULSE Free dal sito: compila il modulo su pulse-ai.it e ricevi via email il link di download (valido 48 ore). Il file ZIP contiene Pulse-Setup.exe e la guida.

Chiudi Revit e Claude Desktop : l'installer deve poter sostituire i file del plugin e aggiornare la configurazione MCP.

Estrai completamente lo ZIP , avvia Pulse-Setup.exe e accetta la richiesta UAC. L'installer rileva automaticamente le versioni di Revit presenti (2023-2027) e installa il plugin per ciascuna.

Riavvia Revit. Comparirà la scheda PULSE nella barra multifunzione, con i pulsanti «Pulse Switch» (avvia/ferma il server, spento all'avvio) e «Settings».

Avvia il server: apri un progetto, clicca Pulse Switch per accendere il server locale. Da questo momento il tuo assistente AI può operare sul modello.

CONFIGURAZIONE MCP

Il tuo assistente di sviluppo AI non vede PULSE?

PULSE è installato correttamente. In alcuni casi l'installer potrebbe non riuscire a configurare automaticamente il tuo assistente di sviluppo AI - Claude Code, Codex, opencode o Cursor - perché percorsi, permessi, versioni e file di configurazione MCP variano da PC a PC.

Incolla il prompt dedicato nell'assistente che usi: gli chiede di completare solo la propria configurazione MCP , senza reinstallare PULSE e senza modificare il modello Revit.

Claude Code

Configura PULSE come server MCP per Claude Code su questo PC Windows. PULSE è già installato: non reinstallarlo e non modificare alcun modello Revit.

Verifica che esista %USERPROFILE%\pulse\server\Pulse.Server.exe. Se esiste, registra o aggiorna in modo idempotente il server MCP con nome pulse-revit usando il comando nativo Claude Code, preservando tutti gli altri server. Riavvia Claude Code se necessario e verifica che pulse-revit sia disponibile. Riportami file controllati, modifiche effettuate ed eventuali blocchi.

Codex

Configura PULSE come server MCP per Codex su questo PC Windows. PULSE è già installato: non reinstallarlo e non modificare alcun modello Revit.

Verifica che esista %USERPROFILE%\pulse\server\Pulse.Server.exe. Configura pulse-revit con il comando nativo Codex quando disponibile; altrimenti aggiorna %USERPROFILE%\codex\config.toml. Prima crea un backup, conserva gli altri server e ripara eventuali blocchi duplicati [mcp_servers.pulse-revit]: alla fine deve esserne uno solo. Verifica che Codex veda pulse-revit e riassumi l'esito.

opencode

Configura PULSE come server MCP per opencode su questo PC Windows. PULSE è già installato: non reinstallarlo e non modificare alcun modello Revit.

Verifica che esista %USERPROFILE%\pulse\server\Pulse.Server.exe. Individua il file di configurazione opencode realmente usato (globale o del progetto), creane un backup e aggiungi/aggiorna una sola voce MCP locale chiamata pulse-revit, senza rimuovere altre configurazioni. Riavvia opencode se necessario, verifica la connessione e riporta l'esito.

Cursor

Configura PULSE come server MCP per Cursor su questo PC Windows. PULSE è già installato: non reinstallarlo e non modificare alcun modello Revit.

Verifica che esista %USERPROFILE%\pulse\server\Pulse.Server.exe. Individua la configurazione MCP realmente usata da Cursor, crea un backup e aggiungi/aggiorna il server locale pulse-revit senza rimuovere altri server. Non duplicare la voce. Riavvia Cursor se necessario, verifica che PULSE sia disponibile e riporta file controllati, modifiche ed eventuali blocchi.

2. Collegare l'assistente AI

PULSE espone un server MCP (Model Context Protocol): un ponte standard che collega Revit all'assistente AI che preferisci. L'installer configura automaticamente Claude Desktop e Codex quando rilevati; Claude Code e gli altri client si registrano con i comandi qui sotto. Sostituisci sempre <TUO_UTENTE> con il tuo nome utente Windows.

Claude Desktop

Apri Settings -> Developer -> Edit Config : è il modo sicuro per aprire il file realmente letto dall'app. Le installazioni classiche usano %APPDATA%\Claude\claude_desktop_config.json ; alcune installazioni MSIX usano invece %LOCALAPPDATA%\Packages\Claude_*\LocalCache\Roaming\Claude\claude_desktop_config.json . L'installer rileva e aggiorna entrambi i percorsi preservando preferenze e altri server MCP.

```
{
  "mcpServers": {
    "pulse-revit": {
      "command": "C:\\Users\\<TUO_UTENTE>\\.pulse\\server\\Pulse.Server.exe"
    }
  }
}
```

Riavvia Claude Desktop dopo la modifica. I server MCP locali sono disponibili nelle chat di Claude Desktop, non nella versione web o in Cowork.

Per IT Manager

Claude Code

Da terminale, un solo comando:

```
claude mcp add pulse-revit -- "C:\\Users\\<TUO_UTENTE>\\.pulse\\server\\Pulse.Server.exe"
```

Codex (OpenAI) - app ChatGPT desktop o CLI

Codex è integrato nell'app ChatGPT desktop (Windows e macOS): il tab Codex, la Codex CLI e l'estensione IDE condividono la stessa configurazione MCP e supportano i server locali. Aggiungi a %USERPROFILE%\\.codex\\config.toml :

```
[mcp_servers.pulse-revit]
command = "C:\\Users\\<TUO_UTENTE>\\.pulse\\server\\Pulse.Server.exe"
```

In alternativa, da terminale: `codex mcp add pulse-revit -- "C:\\Users\\<TUO_UTENTE>\\.pulse\\server\\Pulse.Server.exe"`. La versione web (chatgpt.com) invece non supporta i server MCP locali.

opencode

Client open source e gratuito, utilizzabile anche con modelli locali open-weight (Ollama, LM Studio o qualsiasi endpoint compatibile OpenAI): PULSE funziona così anche senza cloud. Aggiungi a %USERPROFILE%\\.config\\opencode\\opencode.json (o al file opencode.json del progetto):

```
{
  "mcp": {
    "pulse-revit": {
      "type": "local",
      "command": ["C:\\Users\\<TUO_UTENTE>\\.pulse\\server\\Pulse.Server.exe"],
      "enabled": true
    }
  }
}
```

Altri client

Cursor (%USERPROFILE%\\.cursor\\mcp.json), Cline , Continue e Gemini CLI (%USERPROFILE%\\.gemini\\settings.json) usano lo stesso blocco mcpServers mostrato per Claude Desktop. In generale: qualsiasi client MCP desktop con trasporto stdio funziona con PULSE.

Checklist di connessione

Revit aperto con un documento attivo;

Pulse Switch acceso (icona ambra);

l'assistente AI riavviato dopo la modifica della configurazione.

3. Primi passi

Con il server acceso, scrivi all'assistente quello che vuoi ottenere come lo diresti a un collega. Parti da un comando di lettura, senza rischi: « Fai un health check del modello » oppure « Mostrami tutti i muri ». PULSE traduce la richiesta in operazioni verificate sul modello e ti risponde con i risultati.

5. Esempi pratici

Ecco richieste reali in linguaggio naturale. Quelle di sola lettura funzionano già nella versione Free; quelle di scrittura richiedono Premium.

6. Lingua del modello

PULSE rileva automaticamente la lingua di Revit (italiano, inglese, francese, tedesco) e usa i nomi corretti di parametri e categorie. Puoi scrivere all'assistente nella tua lingua: la traduzione verso le operazioni Revit è gestita da PULSE.

Catalogo dei comandi

Descrizioni in italiano. Nomi dei comandi, parametri e valori tecnici restano invariati. Catalogo delle sorgenti del 4 ottobre 2026; traduzioni aggiornate l'8 ottobre 2026.

CAD

`get_cad_geometry`

Legge la geometria annidata di un'importazione o collegamento CAD (import/link): curve, polilinee, mesh, facce dei solidi e punti, in millimetri nel modello Revit. Per impostazione predefinita restituisce un riepilogo; il dettaglio include vertici, indici dei triangoli e, quando disponibili, archi e piani analitici (arcs/planes). I filtri usano limiti campionati, non un ritaglio esatto; il campionamento uniforme delle curve non garantisce un errore massimo sulla corda. Parte da una geometria a basso dettaglio: aumentare detailLevel/curveSamples quando necessario. Il limite di scansione di 10 secondi e l'annullamento vengono verificati tra le chiamate native. Una singola chiamata nativa di geometria o triangolazione non può essere interrotta; maxVertices limita l'output, non il costo di quella chiamata. Controllare scanComplete, truncated e warnings. Le primitive dei solidi sono singole facce; i percorsi sono validi solo per la scansione corrente e non sono handle DWG persistenti. Operazione di sola lettura, senza estrazione di testo o classificazione BIM.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CadTools.cs

Documenti di computo

`export_computo_pdf`

Rende un computo .xpwe già materializzato (o listino prezzi) in un documento di consegna PDF. Funziona interamente lato server e NON richiede l'esecuzione di Revit o del plug-in: legge .xpwe dal disco, controlla localmente lo stato della licenza Pulse e scrive PDF. Prima l'anteprima (dryRun predefinito true): chiama una volta per vedere rowCount/sectionsIncluded/sectionsOmitted, poi di nuovo con dryRun:false per scrivere effettivamente il file. Eccezione deliberata al resto di questo project: ogni altro strumento Pulse.Server termina con revit.ExecuteAsync, un passthrough al plugin sul bridge - questo non tocca mai il bridge, motivo per cui continua a funzionare con Revit chiuso.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ComputoDocumentTools.cs

`export_computo_word`

Rende un computo .xpwe già materializzato (o un listino prezzi) in un documento di consegna Word (.docx). Funziona interamente lato server e NON richiede l'esecuzione di Revit o del plug-in: legge .xpwe dal disco, controlla localmente lo stato della licenza Pulse e scrive .docx. Prima l'anteprima (dryRun predefinito true): chiama una volta per vedere rowCount/sectionsIncluded/sectionsOmitted, poi di nuovo con dryRun:false per scrivere effettivamente il file. Stessa pipeline di export_computo_pdf (ComputoDocumentExporter): differiscono solo il renderer e l'estensione .docx richiesta. Eccezione deliberata al resto di questo project: ogni altro strumento Pulse.Server termina con revit.ExecuteAsync, un passthrough al plugin sul bridge - questo non tocca mai il bridge, motivo per cui continua a funzionare con Revit chiuso.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ComputoDocumentTools.cs

Creazione

`batch_export`

Esportare views/sheets nei formati DWG, DXF, DGN, PDF, immagine (PNG) o Navisworks (NWC).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`change_element_type`

Modificare il tipo di uno o più elementi in un tipo di destinazione specificato da ID o dal nome.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`color_elements`

Colora gli elementi della vista attiva di una categoria raggruppandoli in base al valore di un parametro o reimpostando (cancellando) le sostituzioni del colore. `action=color|reset`. Funziona sulla vista del modello ATTIVA (non su un foglio).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`create_array`

Creare una matrice lineare o radiale. Costruisce un vero e proprio associativo Revit `ArrayElement` (conteggio modificabile) a meno che `associative=false`. lineare utilizza `spacingX/Y/Z`; radiale utilizza `centerX/Y` e `totalAngle`.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`create_color_fill_scheme`

Colora stanze, aree o spazi MEP in base al valore del parametro utilizzando `ColorFillScheme` + `ColorFillLegend` NATIVO di Revit (un'alternativa alle sostituzioni manuali + legenda disegnata a mano di `create_color_legend`, che non è un vero elemento della legenda Revit e rimane l'unica opzione per le altre categorie). Duplica sempre uno schema esistente della categoria di destinazione e imposta il suo `ParameterDefinition` -- Revit non ha alcuna creazione `ColorFillScheme` da zero API. Una vista ha un solo slot di riempimento colore condiviso per tutte le categorie spaziali (Rooms/Areas/Zones/Spaces): l'applicazione di questo schema sostituisce silenziosamente lo schema attivo di qualsiasi altra categoria spaziale su quella vista a meno che nessuno sia attivo: la risposta riporta quali schemi verrebbero (o sono stati) sostituiti. Preview-first: `dryRun` per impostazione predefinita è `true` ed esegue l'intera sequenza in una transazione sandbox, quindi esegue il rollback.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`create_color_legend`

Colora gli elementi in base al valore del parametro e, facoltativamente, crea una vista della legenda.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`create_dimensions`

Crea annotazioni di quota nella vista attiva. Passa una serie JSON di specifiche dimensionali. Elemento mode: `[{viewId, referenceIds:[...], linePoint:{x,y,z}, dimensionStyleId?}]`. Punto a punto mode: `[{viewId, startPoint:{x,y,z}, endPoint:{x,y,z}, linePoint?, dimensionStyleId?}]`. `dimensionStyleId` viene onorato in entrambe le modalità.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`create_filled_region`

Creare una regione riempita in una vista da un contorno chiuso, facoltativamente con fori (circuiti interni).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`create_floor`

Creare un pavimento architettonico da un confine (o da una stanza), eventualmente con fori. Fornire boundaryPoints O roomId.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`create_grid`

Crea griglie dritte con nome a qualsiasi angolo del piano utilizzando le specifiche [{name,startMm:[x,y,z],endMm:[x,y,z]}] o il conteggio + spaziatura legacy X/Y; È supportato anche rename/delete. Tutte le coordinate sono modello mm. Le specifiche esplicite richiedono punti finali con la stessa Z, rifiutano assi names/coincident infiniti XY duplicati (0.1 mm) e non possono essere combinati con count/spacing. Opzionale bottomMm/topMm imposta estensioni verticali. Per impostazione predefinita, le specifiche esplicite sono un'anteprima della transazione Revit che esegue il rollback; imposta dryRun=false per eseguire il commit atomico. Le azioni legacy mantengono la loro applicazione predefinita quando dryRun viene omissso; dryRun=true li visualizza in anteprima senza scrivere.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`create_line_based_element`

Creare muri, travi o controventi. Specifiche: [{category, locationLine:{p0:{x,y,z},p1:{x,y,z},pMid?:{x,y,z}},typeId?,height?,baseLevel?,baseOffset?,structural?,structuralRole?,startOffsetMm?,endOffsetMm?,crossSectionRotationDegrees?}]. Coordinates/heights mm. baseLevel è un'elevazione, NON un id. structural=true crea un muro strutturale. L'inquadratura structuralRole è Trave (predefinita) o Controvento; gli offset dell'endpoint sono relativi al livello risolto e sovrascrivono il comune baseOffset. La rotazione della sezione è in gradi. No dryRun.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`create_point_based_element`

Crea elementi basati su punti (colonne, mobili, lavandini, rubinetti). Passa un array JSON di creazione specs: [{category, locationPoint:{x,y,z}, typeId|typeName|familyName, baseLevel, baseOffset?, rotation?, hostWallId?, hostElementId?, faceReference?, referenceDirection?, facingFlipped?, mirror?, params?}]. hostElementId ospita su QUALSIASI elemento (non solo pareti); faceReference (una stringa di rappresentazione stabile, e.g. da hostFaceReference di get_instance_transform o faceReference di create_reference_plane) colloca invece una famiglia WorkPlaneBased su un piano face/work - mutuamente esclusivo con hostWallId/hostElementId. mirror rispecchia l'istanza creata sul posto. params imposta i parametri di istanza su di esso nella stessa chiamata. OBBLIGATORIO per le famiglie WorkPlaneBased (controllare con isWorkplaneBased o familyPlacementType di get_instance_transform): la chiamata viene rifiutata senza faceReference, perché posizionandone una in qualsiasi altro modo la si fa atterrare silenziosamente su un position/orientation arbitrario che nessun successivo movimento, rotazione, specchio o chiamata set_instance_flip può correggere - deve essere creato a destra face/plane dall'inizio.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

create_revision

Elenca, crea, aggiorna o assegna revisioni ai fogli. action=list|create|set|add_to_sheets. 'set' aggiorna una revisione esistente (richiede revisionId).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

create_room_separation_lines

Creare linee di separazione della stanza (OST_RoomSeparationLines) da un poligono di confine chiuso, in modo che create_room possa avere successo in un'area a pianta aperta senza muri di cinta: un atrio, un atrio o una zona divisa il cui confine non è un muro. L'elevazione del confine deriva dal livello della vista data, mai da un livello superato separatamente, quindi i due non possono mai essere in disaccordo. Anteprema-first: dryRun assume per impostazione predefinita true e convalida il poligono senza creare elementi. Restituisce awaitingUser e un planId monouso. Dopo l'approvazione dell'utente, ripetere lo stesso input con dryRun:false e planId entro cinque minuti. I documenti modificati richiedono una nuova anteprema. La conferma nativa e l'elaborazione degli errori si verificano al momento dell'applicazione; l'anteprema non può garantire l'accettazione.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

create_structural_framing_system

Creare un sistema di travi su un livello sopra un'area rettangolare. Di default costruisce un vero e proprio associativo Revit BeamSystem (layout modificabile); impostare associative=false per raggi indipendenti sciolti.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

create_surface_based_element

Realizza pavimenti, platee di fondazione, soffitti o tetti piani. Specifiche: [{category, boundary:{outerLoop:[{p0:{x,y,z},p1:{x,y,z}},...]}, typeId?, baseLevel?, baseOffset?, structural?}]. XYZ e elevations/offsets in millimetri; baseLevel è un'elevazione, NON un ElementId. strutturale si applica ai pavimenti; OST_StructuralFoundation richiede una soletta di fondazione FloorType. I tetti sono piatti; solo anello esterno. No dryRun.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

create_text_note

Crea note di testo in una vista. Passa un JSON array: [{text, position:{x,y,z}, viewId?, textNoteTypeId?, width?, horizontalAlignment?, verticalAlignment?, rotation?, leader?}]. la rotazione è in gradi; il leader è left|right|leftArc|rightArc.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

export_elements_data

Esporta i dati degli elementi per categoria come JSON o CSV, con filtraggio dei parametri e un filtro di inclusione opzionale. Utilizzare per semplici filtri a parametro singolo; utilizzare ai_element_filter per query complesse multicriterio.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`export_families`

Esporta le famiglie caricate come file .rfa in una directory di destinazione.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`export_room_data`

Esporta locale data: numero, nome, livello, area, volume e perimetro. (Non restituisce elementi di delimitazione: utilizzare `get_room_openings` o `get_elements_in_spatial_volume` per ciò che contiene una stanza.)

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`export_schedule`

Esportare le righe della pianificazione come JSON e facoltativamente scriverle in un file di testo delimitato passando `exportPath`.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`export_to_excel`

Esporta i dati degli elementi da una categoria Revit in un file Excel.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`import_from_excel`

Importa i valori dei parametri da un file Excel negli elementi Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

`import_table`

Importa un file CSV/TSV come tabella formattata in una vista di disegno o legenda.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\CreationTools.cs

Scritture differite

`modelsnapshot_capture_read`

PULSEplusO Core snapshot capture: legge ProjectInformation PROJECT_STATUS del documento aperto la cui chiave del documento è uguale a `enrolled_model_id` e restituisce un frame `core-modelsnapshot-capture-1.0`. Sola lettura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\DeferredWriteTools.cs

`set_project_status_conditional`

PULSEplusO Core differito write: imposta ProjectInformation PROJECT_STATUS nel documento aperto identificato da `documentKey` SOLO se il suo valore corrente è uguale al valore atteso con segno, al massimo una volta per `executionIdentity`. Restituisce un frame di ricevuta. Passa solo `Grant+Signature+Payload` o gli altri argomenti. Non per use interattivo: utilizzare `set_project_info` per modificare le informazioni sul progetto.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\DeferredWriteTools.cs

Dynamo

dynamo_generate_graph

Generare e salvare un grafico Dynamo .dyn valido incentrato su Python. Utilizzare SOLO quando nessuno strumento Pulse nativo copre l'attività E l'utente ha approvato un approccio Dynamo/Python. Richiede EnableDynamo=true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\DynamoTools.cs

dynamo_get_status

Segnala Dynamo per lo stato Revit (presente, versione, disponibilità CPython3) e se è impostato EnableDynamo.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\DynamoTools.cs

dynamo_list_graph_io

Elenca inputs/outputs di un grafico .dyn (analizza il file, non lo esegue).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\DynamoTools.cs

dynamo_run_graph

Esegue un grafo Dynamo .dyn senza interfaccia grafica all'interno di Revit. Usare solo quando nessuno strumento nativo Pulse copre l'operazione e l'utente ha approvato un approccio Dynamo/Python. Richiede EnableDynamo=true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\DynamoTools.cs

Elementi

add_curtain_grid_line

Aggiungere una linea di griglia ad una griglia di facciata continua, suddividendo una facciata. Anteprema-first: dryRun per impostazione predefinita è true e l'anteprema crea VERAMENTE la riga in una transazione sandbox, misura l'effetto, quindi esegue il rollback. Tutto quanto riportato di seguito è stato misurato su un modello reale prima che lo strumento fosse scritto. UN RIFIUTO HA DUE CAUSE DISTINTE ED ENTRAMBI GETTANO CON UN MESSAGGIO VUOTO, quindi l'eccezione TYPE è l'unico segnale - misurato tramite MCP: ArgumentException = la posizione è FUORI DALL'HOST; InvalidOperationException = ESISTE GIÀ UNA LINEA GRIGLIA in quella posizione. Entrambi restituiscono un errore strutturato e il payload trasporta il tipo. (Una versione precedente di questo testo affermava che la collisione ha restituito null come risultato benigno 'notCreated'; che è stato misurato come FALSO - genera. Il ramo 'notCreated' viene mantenuto perché AddGridLine può restituire null, ma nessun input noto lo raggiunge.) IN CASO DI RIFIUTO I CONTEGGI 'after' SONO DICHIARATI NON MISURATI anziché riportati, perché Revit muta lo stato in-transation prima di throwing: leggendoli mostrava i pannelli 28 -> 32 per una chiamata che non creava nulla. LA POSIZIONE È NELLE COORDINATE DEL MODELLO, in mm: spazio misurato, non host o normalizzato. E ATTENZIONE ALL'ASIMMETRIA: l'aggiunta di una linea aumenta il conteggio dei pannelli, mentre RemoveSegment di add_remove_grid_segment NON lo diminuisce (ridimensiona invece i pannelli) - quindi non dare per scontato che le due operazioni siano speculari. oneSegmentOnly non crea una line più corta: misurata, crea la linea con metà dei suoi segmenti già saltati.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

`add_remove_grid_segment`

Aggiungi o rimuovi segmenti su una linea di griglia di facciata continua: aprendo e chiudendo la griglia di una facciata. Anteprima-first: `dryRun` per impostazione predefinita è `true` e l'anteprima esegue REALMENTE l'operazione in una transazione `sandbox`, misura l'effetto, quindi esegue il `rollback`. DUE COSE QUI CONTRADDICONO LA DOCUMENTAZIONE DI REVIT, entrambe misurate su un modello reale prima che questo strumento fosse written: (1) `RemoveSegment` NON UNISCE I PANNELLI. Le osservazioni promettono che "i due pannelli vicini verranno fusi in un unico pannello"; misurato, il conteggio dei pannelli rimane identico (28 -> 28) e così anche il pannello IDS (0 è scomparso, 0 nuovo - controllato differenziando `GetPanelIds`, poiché uno eliminato più uno creato darebbe anche delta 0). Ciò che cambia è l'AREA del pannello (222.29 -> 222.78): i pannelli vengono RIDIMENSIONATI. Pertanto l'anteprima riporta `panelAreaM2` come effetto reale e `panelCount` solo per completezza: NON aspettatevi un pannello in meno. (2) IL BLOCCO NON PROTEGGE: `RemoveSegment` riesce su una linea della griglia con `Blocco == true`, nessuna eccezione. I documenti Revit si bloccano come proprietà scrivibile, non come garanzia, quindi questo strumento rifiuta una linea bloccata a meno che `allowLockedLines: true` non venga passato deliberatamente: misurato, le linee 4 di 9 sull'host di riferimento sono bloccate. (3) IT DISTRUGGE I MONTANTI E 'add' NON LI RIPORTA RETRO - misurato e non documentato da Revit: rimuovendo UN segmento ha portato la griglia di riferimento dai montanti 53 a 50, e ripristinando lo stesso segmento con 'add' l'ha lasciata a 50, i.e. la campata restaurata resta nuda. Non 1:1 con segmenti (i montanti sono condivisi tra campate adiacenti, quindi -3 per un segmento). L'anteprima riporta `mullionCount` quindi questo è visibile prima del commit, perché l'azione opposta non lo annulla. Inoltre: un segmento già nello stato richiesto è un NO-OP silenzioso nello API (documentato su `AllSegmentCurves`), quindi viene segnalato come saltato con il motivo anziché conteggiato come fatto - mentre una curva Revit non può individuare THROWS, e le due vengono tenute separate. I segmenti vengono indirizzati dall'INDICE basato su 0 nelle curve esistenti (per rimuovere) o saltate (per aggiungere), i cui conteggi `get_curtain_grid` riporta per riga.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs`

`ai_element_filter`

Trova o elenca gli elementi per categoria, classe di elemento, simbolo di famiglia, riquadro di delimitazione o livello: muri, porte, finestre, colonne, pavimenti, tetti, stanze, telaio strutturale, ecc. (tramite codici `OST_`). Lo strumento principale per "find/list/how molti <categoria>" o "tutti i <categoria> al livello X". I filtri si combinano con AND (predefinito) o OR e possono essere invertiti (NOT); supporta il filtraggio di tipi e istanze. Per il filtraggio VALORE parametro utilizzare `filter_by_parameter_value`.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs`

`change_curtain_panel_type`

Cambia il tipo di pannelli per tende. Anteprima-first: dryRun per impostazione predefinita è true e l'anteprima esegue VERAMENTE la modifica in una transazione sandbox, legge il tipo risultante, quindi esegue il rollback, quindi typeAfter e changedCount vengono MISURATI, non previsti. Quattro cose decidono se una chiamata funzionerà e tutte vengono controllate PRIMA dell'apertura della transazione in modo che un'ambiguità non possa mai lasciare un facade scritto a metà: (1) UN PANNELLO DI TENDA NON È SEMPRE UN Pannello: GetPanelIds restituisce "Pannello o Elementi di parete" ed eseguendo get_curtain_grid su un modello reale sono state trovate anche famiglie DOOR caricate nelle facciate (6 di host 65), che riportano come FamilyInstance. Ridigitare un vetro, un muro annidato e una porta a facciata continua sono tre operazioni diverse, quindi ogni obiettivo viene segnalato per quello che effettivamente È (elementClass) e le combinazioni incompatibili con la categoria vengono rifiutate con il motivo, perché ChangePanelType genera ArgumentException 'se il simbolo di input non può essere utilizzato per il pannello'. (2) I NOMI DEI TIPI SONO AMBIGUI: il modello misurato contiene un tipo di pannello chiamato letteralmente 'Wall'. Quindi targetTypeNames si risolve tramite il contratto AmbiguousInput: una partita si risolve, diverse restituiscono i candidati e CHIEDONO, nessuna restituisce ciò che È disponibile. Preferisco targetTypeIds. (3) Un pannello già del tipo richiesto viene segnalato come rifiutato con tale motivo anziché conteggiato come modificato. (4) ChangePanelType lancia anche InvalidOperationException 'quando il cambio di tipo non è riuscito' - imprevedibile, quindi viene rilevato PER PANNELLO: un errore non interrompe il batch e non viene mai conteggiato come un successo. Limitato ai pannelli 200 per chiamata perché il metodo si rigenera per pannello.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

`compare_instance_state`

Riporta lo stato di posizionamento di un'istanza di famiglia (trasformazione, FacingFlipped/HandFlipped/Mirrored, host, riquadro di delimitazione, conteggi di relazioni di taglio) e, quando viene fornito il valore previsto, lo confronta campo per campo con le tolleranze. Cattura senza aspettative prima di un'operazione, quindi confronta il risultato in seguito: funziona anche quando l'elemento originale è stato eliminato e ricreato (e.g. da rehost_instance), poiché il confronto avviene con un'istantanea catturata, non con un secondo ID elemento attivo. Sola lettura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

`connect_mep_elements`

Collega o scollega due connettori MEP: il rimedio per le estremità aperte segnala get_mep_connectivity. Identificare ciascun lato con elementId + connectorId da get_mep_connectivity. Anteprima-first: dryRun viene impostato per impostazione predefinita su true e l'anteprima ESEGUE l'operazione in una transazione sandbox, quindi la ripristina, quindi elementsCreated/elementsRemoved vengono MISURATI, non previsti (entrambi i metodi Revit restituiscono void, quindi un diff è l'unico modo per sapere cosa è successo). LEGGI fittingCaveat prima di concludere qualsiasi cosa su fittings: fittingsAddedByRevit:0 NON promette che Revit non ne aggiungerà mai uno: è documentato che ConnectTo potrebbe creare un raccordo "se necessario" e la fedeltà di rollback è stata misurata per una disconnessione e una riconnessione diretta ma non per un ConnectTo che crea. PRIMA della transazione avvengono tre rifiuti, ciascuno supportato da un exception documentato: domini diversi, già connessi, e stesso connettore su entrambi i lati. Le differenze di forma e dimensione NON vengono deliberatamente rifiutate: Revit aggiunge un raccordo di transizione "se necessario", quindi vengono segnalate come campi (shapeA/shapeB, sizeMmA/sizeMmB, gapMm) e la chiamata è tua. fingerprint opzionale: passa expectedDomainA/B e expectedOriginMmA/B e lo strumento verifica che il connettore sia quello che intendi prima di scrivere: vale la pena utilizzarlo, perché Connector.Id è un indice POSIZIONALE all'interno dell'elemento (misurato), quindi un ID salvato da una lettura precedente può puntare a un connettore diverso.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

copy_elements

Copia elementi con mm opzionale offset. Può scegliere come target una vista diversa (sourceViewId+targetViewId) o un altro documento OPEN (targetDocumentTitle).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

create_area

Crea un'area all'interno di un poligono con confine chiuso esplicito (mm), su uno schema di area esistente. Revit NON ha API per creare un NUOVO schema di area (è solo UI, tramite calcoli di area e volume) - utilizzane uno esistente, e.g. "Area edificabile lorda" o "Area affittabile", che ogni progetto viene spedito per impostazione predefinita. A differenza di una Stanza, un'Area non viene mai rilevata automaticamente racchiudendo walls: questo strumento crea gli elementi AreaBoundaryLine per il poligono stesso dato, quindi posiziona l'Area in interiorPoint. Le aree possono vivere solo in una vista della pianta dell'area con lo stesso ambito (schema, livello): una viene riutilizzata se esiste già, altrimenti creata e segnalata (areaPlanViewWasCreated), poiché l'aggiunta di una vista completamente nuova è un effetto collaterale visibile, non silenzioso. Antepima-first: dryRun per impostazione predefinita è true ed esegue l'intera creazione in una transazione sandbox, quindi esegue il rollback.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

create_mep_curve

Crea un condotto, un tubo, un condotto o una passerella portacavi tra due punti (coordinate in MILLIMETRI). Utilizza SOLO il sovraccarico a due punti di Revit, e questa è una decisione di sicurezza, non una limitazione al funzionamento di around: è documentato che i sovraccarichi basati su connettore creano comunque l'elemento quando la connessione fallisce, lanciando InvalidOperationException e perdendo il nuovo ElementId - in modo che possano lasciare un orfano irraggiungibile. Questo strumento non può. Revit può aggiungere raccordi propri ("sono inclusi raccordi aggiuntivi per effettuare una connessione valida"), quindi la risposta riporta totalElementsCreated e fittingsAddedByRevit, non solo la curva. Lunghezza minima 1/10 pollici (~2.54 mm), controllata nella parte anteriore. conduit/cabletray NON hanno alcun concetto di tipo di sistema in Revit API (a differenza di duct/pipe) - omettere systemTypeId per questi due tipi; passarlo è semplicemente inutilizzato, mai un errore. Antepima-first: dryRun per impostazione predefinita è true e l'antepima ESEGUIsce la creazione in una transazione sandbox, quindi la ripristina, quindi i conteggi degli elementi vengono misurati anziché previsti.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

create_path_of_travel

Crea elementi Percorso di viaggio (percorsi di uscita) tra coppie di punti, in MILLIMETRI, nella vista ATTIVA, che deve essere una vista in pianta. LEGGI calculationStatus prima di concludere anything: Crea di Revit non restituisce nulla per almeno quattro ragioni diverse, e solo lo stato separa "non esiste davvero alcun percorso" (NoPathOfTravel) da "la VISUALIZZA è il problema" (PointOutsideActiveCrop, SplitView) e "l'input è" (StartAndEndPointsTooClose). Un percorso restituito con ResultAffectedByCrop ESISTE ma potrebbe non essere il percorso più breve: viene segnalato come un successo con isOptimalityLimited:true, non come un errore. Misurato su un vero model: le planimetrie di 10 di 11 avevano un ritaglio attivo, quindi questo è il caso normale. Ogni risultato porta il proprio startIndex/ endIndex ed entrambi i punti, quindi l'accoppiamento non deve mai essere dedotto dall'ordine dell'array (l'ordinamento mappato di Revit varia prima START, che è l'opposto della lettura intuitiva). viewId NON è accepted: attiva prima il piano che desideri: non è documentato se Create si comporta in modo identico su una vista non attiva, quindi il parametro viene rifiutato anziché ignorato silenziosamente. Anteprima-first: dryRun per impostazione predefinita è true e crea effettivamente i percorsi in una transazione sandbox per registrarne gli stati, quindi esegue il rollback.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

create_reference_plane

Crea un piano di riferimento da tre punti (bubbleEnd, freeEnd, thirdPoint). Il terzo punto fissa l'orientamento dell'aereo e non deve trovarsi sulla linea bubbleEnd-freeEnd. Anteprima-first: dryRun ha come valore predefinito true, ma Revit non documenta in che modo i punti normali risultanti, quindi l'anteprima non può mostrarli: la normale reale viene restituita solo dopo la creazione (dryRun:false). Controllatelo prima di posizionare qualsiasi cosa sullo plane: un piano di riferimento è economico e reversibile (delete_element), una famiglia posizionata su quello sbagliato no. Il codice faceReference del piano creato (una stringa di rappresentazione stabile) può essere passato a create_point_based_element per posizionare su di esso una famiglia basata sul piano di lavoro.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

delete_selection

Elimina un filtro di selezione salvato per nome

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

divide_mep_system

Suddividere un sistema MEP i cui componenti formano più di una rete fisica disconnessa in un sistema per rete: il rimedio per get_mep_systems' isMultipleNetwork:true. SOLO HVAC e tubazioni (documentato Revit API limit); l'elettricità viene rifiutata con un errore denominato. IMPORTANTE, misurato non documented: il sistema originale è CONSUMATO, quindi N reti producono N nuovi sistemi e il conteggio dei sistemi del modello cresce di N-1, non N. L'anteprima riporta physicalNetworkCount (cosa sa API) e netSystemCountDelta (cosa farà il modello) - deliberatamente non promette un conteggio di creazioni che API non garantisce mai. Non un undo pulito: ricollegando gli elementi successivamente si uniscono nuovamente i sistemi ma si lascia un nome con suffisso come "X (1)". Anteprima-first: dryRun viene impostato automaticamente su true; richiamare con dryRun:false per candidarsi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

find_undimensioned_elements

Trova elementi a cui non fanno riferimento le dimensioni

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

find_untagged_elements

Trova elementi senza tag in una vista

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

get_cad_lines

Leggi le righe di un'importazione o di un collegamento DWG. Predefinito: un riepilogo per livello+colore. Passare il colore del livello and/or per ottenere le singole curve (punti finali in mm, coordinate del modello), ciascuna contrassegnata come costruibile (diritta e abbastanza lunga per un MEP duct/pipe). Sola lettura: il primo passaggio per trasformare le righe DWG in MEP. Omettere importInstancelId per selezionare automaticamente la singola importazione o per elencare i candidati quando ne esistono diversi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

get_curtain_grid

Descrive le griglie di facciate continue, sistemi di facciata continua e coperture vetrate: linee U/V con stato dei segmenti, pannelli e montanti. Sola lettura, solo documento attivo. Quattro vincoli API sono essenziali. (1) Un pannello può essere una parete: GetPanelIds può restituire elementi Panel o Wall. Nel modello misurato, 41 su 681 pannelli (6%) erano pareti, ossia facciate annidate. Il totale è in panels.byElementClass e panels.byFamily, non in un semplice panelCount: non interpretare 681 come altrettanti Panel quando ci sono 464 vetrate, 152 pannelli fotovoltaici e 41 pareti, che richiedono operazioni diverse. (2) Ogni faccia di un sistema ha una griglia: CurtainSystem.CurtainGrids contiene più CurtainGrid; il risultato è host -> grids[] e una parete ne fornisce una. Non assumere una griglia per elemento. (3) I segmenti hanno tre stati: allSegmentCount, existingSegmentCount, skippedSegmentCount. Lo stato determina le operazioni ammesse: AddSegment ripristina un segmento omesso; RemoveSegment/AddMullions agiscono su uno esistente. (4) Le celle non sono calcolate per impostazione predefinita: il risultato dichiara cells.status='omitted_for_cost'. GetCurtainCells restituisce geometria CurveLoops dal costo non documentato dall'API Revit; usare verbose=true per calcolarla. I dati non leggibili sono dichiarati 'not_measured', non sostituiti da uno zero non misurato.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

get_element_parameters

Ottieni tutti i parametri di elementi specifici in base ai relativi ID elemento Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

get_element_solid_geometry

Ottieni la geometria solida REALE di un elemento (riquadro di delimitazione, baricentro, volume m3, conteggi face/edge E sezione trasversale dedotta shape: circular/rectangular/complex) in mm e le coordinate del modello. A differenza di get_BoundingBox questo riflette il solido effettivo DOPO unisce e taglia, e riporta la sezione FORMA - essenziale per rebar: un bbox 613x613 può essere una pila circolare Ø610 dove i tiranti d'angolo bars/rectangular cadono all'esterno. Utilizzare sempre questo, non il riquadro di delimitazione, per posizionare l'armatura all'interno di un host.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

`get_elements_in_spatial_volume`

Trova elementi all'interno di un riquadro di delimitazione 3D o del volume di una stanza. `volumeType`=utilizzo camera `volumeIds`; `volumeType`=usi personalizzati `customMinX..customMaxZ`.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs`

`get_instance_transform`

Legge la geometria di posizionamento dell'istanza della famiglia senza altri strumenti `exposes`: la trasformazione di posizionamento (base X/Y/Z + origine), `FacingFlipped/HandFlipped/Mirrored`, l'elemento `host` (`id`, categoria e `hostFamilyName` quando l'`host` è esso stesso un'istanza della famiglia), `levelId`, `familyPlacementType` e `isWorkplaneBased` (il controllo autorevole prima di ricreare questa istanza - `WorkPlaneBased` le famiglie DEVONO passare attraverso `faceReference` di `create_point_based_element`, altrimenti la chiamata viene rifiutata), se l'istanza si trova su un piano `face/work` in questo momento (con una stringa `faceReference` riutilizzabile) e i flag di capacità `CanFlipFacing/CanFlipHand/CanRotate`. Sola lettura.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs`

`get_linked_elements`

Elementi di query dai modelli Revit collegati con filtro opzionale. `parameterNames` è additivo: senza di esso vengono restituiti solo i campi di base.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs`

`get_mep_connectivity`

Riportare i connettori MEP di elementi specifici (o un'intera categoria di portanti del connettore): tipo di connettore, dominio, posizione in mm, se ciascuno è fisicamente collegato e gli elementi di riferimento. Solo documento attivo di sola lettura. Questo è il dettaglio per connettore dietro il conteggio `get_mep_systems'`
`unusedPhysicalConnectorCandidates`: usalo per trovare QUALE connettore su QUALE elemento è aperto. Due contratti danno forma alla risposta, entrambi degni di fiducia su intuizione: (1) `Connector.AllRefs` è documentato per includere ENTRAMBE le connessioni fisiche e logiche, quindi i riferimenti vengono restituiti suddivisi in `physicalReferences` / `logicalReferences` / `unclassifiedReferences` (classificati per coppia con `IsConnectedTo`) e NON devono essere sommati per significare 'connected' - su un circuito elettrico a cui appartiene l'insieme logico, non un giunto di condotto. (2) `isOpenPerPlumbingApi` deriva da `PlumbingUtils.HasOpenConnector`, che il Revit API prevede SOLO TUBAZIONE (curva tubo, raccordo, accessorio); altrove è null anziché false, perché per HVAC ed elettrico non esiste tale autorità e la risposta è la scansione del connettore. `observedConnectorTypes` elenca i valori `ConnectorType` in realtà present: non viene applicato alcun filtro di uguaglianza, poiché i valori enum sono composti (`Physical=19=End|Curve|Surface`) senza [Flags], quindi il filtraggio `by ==` esclude silenziosamente i connettori reali.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs`

`get_mep_systems`

Elenca i sistemi MEP (HVAC, tubazioni, elettrici) con dominio, tipo, apparecchiatura di base, numero di terminali e stato di connettività. Sola lettura, solo documento attivo: non include i sistemi dei modelli collegati. È quindi previsto trovare 0 sistemi in un modello architettonico ospitante il cui MEP risiede in un collegamento; questo non dimostra l'assenza di sistemi. `terminalElementCount` conta i terminali ed esclude l'apparecchiatura o quadro di base (equipment/panel), secondo il contratto API Revit. Non confondere i campi: `productType` identifica il prodotto Revit della sessione (mep/revit/architecture/structure/lt/unknown, da `Application.Product`); `productReportsMep` indica se quel prodotto può riportare dati MEP; `hasMepContent` indica se questo modello contiene sistemi o elementi MEP. Un modello architettonico in Revit completo può avere `productReportsMep:true` e `hasMepContent:false`: un risultato realmente vuoto, non una mancata capacità di lettura.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs`

`get_paths_of_travel`

Elenca i percorsi di esodo Path of Travel del documento attivo: punti iniziale, finale e intermedio (start/end/midpoint) in mm, lunghezza sommata delle curve, numero di punti intermedi, vista di appartenenza e stato del suo ritaglio. Sola lettura, solo documento attivo. `curveLengthMm` deriva da `GetCurves()`: l'API non espone una proprietà di lunghezza sull'elemento (31 proprietà verificate nella DLL distribuita). È quindi una lunghezza geometrica, non un parametro Revit che potrebbe arrotondare diversamente. Il percorso è calcolato nella propria vista: un ritaglio attivo può renderlo non ottimale; per questo viene riportato `ownerViewCropBoxActive`. Un risultato vuoto include `emptyResultNote`: i percorsi appartengono a una vista e devono essere creati esplicitamente, quindi anche un modello completo può legittimamente non contenerne.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs`

`get_room_openings`

Restituisce porte e finestre (doors/windows) adiacenti ai locali, con le dimensioni. Filtra tramite `roomIds`, `roomNumbers` o `levelName`. Usare prima `summaryOnly` per ottenere i conteggi per locale.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs`

`get_selected_elements`

Ottieni gli elementi attualmente selezionati in Revit.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs`

`import_from_powerbi`

Legge un Power BI CSV precedentemente esportato (o modificato manualmente) e riscrive i valori dei parametri negli elementi Revit. Accetta sia CSV delimitati da ',' (push_to_powerbi) che ';' (export_elements_data): il separatore viene rilevato automaticamente dalla riga di intestazione. Identifica gli elementi tramite la colonna `ElementId`. I campi incorporati e i parametri di sola lettura vengono ignorati. Il valore predefinito è `dryRun=true` in modo che i chiamanti possano prima visualizzare l'anteprima.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs`

inspect_selection

Legge la selezione corrente di Revit. Esattamente un elemento selezionato restituisce parametri completi di istanza e tipo. Più di uno, o nessuno, restituisce un riepilogo compatto id/name/category senza dettagli sui parametri. Sola lettura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

load_selection

Elenca le selezioni salvate o caricane una per nome. Ometti il nome nell'elenco; pass name per caricare quella selezione (e selezionarla nella vista attiva a meno che selectInView=false).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

match_element_properties

Copia i valori dei parametri da un elemento di origine a uno o più elementi di destinazione. Sovrascrive i valori esistenti dei target, quindi per impostazione predefinita viene visualizzata un'anteprima che mostra ciascuno currentValue -> newValue: passa dryRun=false per scrivere.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

measure_between_elements

Misura la distanza tra due elementi o due punti in mm. Fornire elementId1/elementId2 o point1/point2 (come array JSON [x,y,z]).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

modify_element

Sposta, ruota, specchia o copia elementi. I vettori sono oggetti {"x":mm,"y":mm,"z":mm} JSON. lo spostamento necessita di traduzione; la rotazione richiede rotationCenter + rotationAngle (gradi; rotationAxis predefinito è Z); lo specchio necessita di mirrorPlaneOrigin + mirrorPlaneNormal; la copia richiede copyOffset.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

operate_element

Seleziona, evidenzia, isola, nascondi o esegui lo zoom sugli elementi. Azioni: seleziona, casella di selezione, imposta colore, imposta trasparenza, nascondi, nascondi temporanea, isola, mostra, reimposta isola, elimina. ANTEPRIMA PRIMA SOLO SU 'delete': è l'unica azione irreversibile qui (chiama doc.Delete), quindi per impostazione predefinita è dryRun e riporta ciò che DOVREBBE eliminare più eventuali ID che non esistono - passa dryRun:false per eliminare effettivamente e Revit chiederà quindi conferma. Le altre nove azioni ignorano dryRun: sette sono sostituzioni grafiche annullate con un clic e due modificano solo la selezione UI, quindi eliminarle costituirebbe rumore. Nota sullo count dell'anteprima: Revit potrebbe eliminare PIÙ degli ID elencati, poiché l'eliminazione di un host elimina anche ciò che ospita.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

pbi_check_auth

Segnala lo stato di accesso Power BI dell'utente. Con signIn=true, avvia un flusso di codice dispositivo MSAL all'interno di Revit (un TaskDialog mostrerà il codice URL +; l'utente completa l'accesso nel proprio browser). Il token viene memorizzato nella cache con crittografia DPAPI in %LOCALAPPDATA%\pulse\msal_cache.bin.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

pbi_create_dataset

Crea un VUOTO Pulse v2 push dataset: una tabella piatta 'Data' con le colonne specificate (default: le colonne dell'elemento base 9). Se il nome esiste termina con awaitingUser. Solitamente non necessario: pbi_publish_* con mode:crea crea e riempie il set di dati. Il valore predefinito di dryRun è true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

pbi_get_binding

Restituisce la potenza BI ProjectBinding memorizzata per il documento Revit attivo (workspaceId, datasetId, datasetName, docKey, updatedAt). Restituisce bound=false con un suggerimento se non esiste ancora alcuna associazione. Sola lettura: chiamalo per verificare a quale workspace/dataset è collegato un documento prima della pubblicazione.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

pbi_list_datasets

Elenca i set di dati push in un'area di lavoro Power BI. Utile per trovare un set di dati Pulse esistente prima della pubblicazione. Sola lettura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

pbi_list_workspaces

Elenca gli spazi di lavoro (gruppi) Power BI accessibili all'utente che ha effettuato l'accesso. Sola lettura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

pbi_publish_elements

Pubblica elementi Revit delle categorie specificate in un set di dati push Power BI con UNA tabella piatta 'Data' (ElementId, UniqueId, Categoria, Famiglia, Tipo, DocumentTitle, DocumentPath, EpisodeId, ProjectId + una colonna digitata per parametro scelto, unità di visualizzazione del progetto, null quando vuoto). Chiede (awaitingUser con candidati) quando mancano categoryFilter, parameterNames o modalità; non indovina mai. mode: crea (nuovo set di dati, si interrompe in caso di conflitto di nomi) | aggiornamento (dataset esistente: le colonne devono corrispondere, altrimenti si ferma con la differenza). Il valore predefinito di dryRun è true: anteprema columns/rows senza scrittura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

pbi_publish_schedules

Pubblica pianificazioni Revit su Power BI: UN set di dati push per pianificazione, UNA tabella piatta 'Data' (ElementId, UniqueId, DocumentTitle, DocumentPath, EpisodeId, ProjectId + campi visibili della pianificazione, una riga per elemento). Chiede (awaitingUser) quando mancano scheduleIds o modalità. Calculated/combined i campi senza parametro vengono omessi con un avviso; gli orari chiave vengono rifiutati. mode: crea | aggiornamento. Il valore predefinito di dryRun è true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

pbi_publish_selection

Pubblica la selezione corrente Revit in un set di dati push Power BI con UNA tabella piatta 'Data' (stesse colonne di pbi_publish_elements; la categoria distingue le righe). Un set di dati per documento ('{Project}_Selezione'), svuotato e riscritto ad ogni pubblicazione. Chiede (awaitingUser) quando non è selezionato nulla o quando mancano parameterNames/mode. mode: crea | aggiornamento. Il valore predefinito di dryRun è true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

pbi_query

Esegue una query DAX sul set di dati Power BI associato e seleziona gli elementi corrispondenti in Revit. Utilizza i parametri del modello (categoria, livello, parameterName+parameterValue, exportRunId) per i filtri comuni o un daxQuery non elaborato per le query avanzate. action=isola nasconde altri elementi invece di selezionare. workspaceId/datasetId può essere omissso quando esiste un ProjectBinding.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

pbi_sign_out

Disconnette l'alimentazione da BI revocando tutti i token MSAL memorizzati nella cache. Dopo questa chiamata pbi_check_auth(signIn=false) restituisce signedIn=false. Nella risposta viene riportato l'account a cui si è effettuato l'accesso in precedenza. Utilizzare pbi_check_auth(signIn=true) per autenticarsi nuovamente.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

pbi_trigger_refresh

Attiva un aggiornamento su richiesta di un set di dati Power BI in modalità importazione dopo che Pulse ha aggiornato la propria origine OneDrive/CSV. Solo premio. Richiede l'accesso a Power BI oltre ai GUID dell'area di lavoro e del set di dati.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

powerbi_visual_action

Esegue un'azione visiva Pulse Power BI dopo powerbi_visual_preflight e richiede le conferme dell'utente. Azioni: selezionare | isolare | colore | reimpostare | createView. Effettuare sempre il preflight prima.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

powerbi_visual_preflight

Controllo di disponibilità specifico dell'azione di sola lettura per un comando copiato dall'oggetto visivo Pulse Power BI. Chiamalo sempre prima di powerbi_visual_action.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

push_table_to_powerbi

Scrive una tabella arbitraria (intestazioni + righe) in un CSV nella cartella Pulse/OneDrive per Power BI. Da utilizzare quando i dati sono stati calcolati dall'utente (analisi LLM, aggregazione multi-doc, ecc.) e NON è un dump diretto di elementi Revit. Passa le intestazioni come un array di stringhe JSON; passare le righe come un array JSON in cui ogni elemento è un array (posizionale) o un oggetto chiave tramite intestazione. Il file si trova accanto alle uscite push_to_powerbi e può essere prelevato dalla stessa sorgente della cartella Power BI.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

push_to_powerbi

Esporta i parametri degli elementi o le pianificazioni esistenti in CSV in OneDrive\Pulse\<DocumentName> per l'aggiornamento Power BI. categorie+parameterNames -> uno elements.csv; scheduleIds -> uno CSV per programma. L'ambito può essere limitato alla visualizzazione o selezione attiva.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

rehost_instance

Riassegna un'istanza di famiglia basata su punti a un nuovo elemento host. FamilyInstance.Host non ha setter in Revit API, quindi elimina l'istanza e la ricrea sul nuovo host nella stessa posizione, preservando type/rotation/FacingFlipped/HandFlipped, ma NON i parametri dell'istanza, che vengono persi con l'istanza eliminata (riapplicare con set_element_parameters se necessario). Solo istanze basate su punti. Anteprema-first: dryRun ha il valore predefinito true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

renumber_elements

Rinumera rooms/doors/windows per posizione o nome. Scrive nel parametro specificato; supporta prefix/suffix e start/increment.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

save_selection

Salva la selezione dell'elemento come filtro con nome. La sovrascrittura di un insieme di selezione esistente ne sostituisce l'appartenenza, quindi il valore predefinito è preview: passa dryRun=false per scrivere.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

section_box_from_selection

Crea un riquadro di sezione 3D dagli elementi selezionati

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

select_from_powerbi

Elementi Selects/zooms/isolates nella vista attiva Revit da un drill-through Power BI. Azione: selezionare | evidenziare | isolare.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

set_element_phase

Assegna la fase created/demolished agli elementi. Passa un array JSON di requests: [{elementId, phaseCreatedId?, phaseDemolishedId?}].

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

set_element_workset

Sposta gli elementi in un workset diverso. Passa un array JSON di requests: [{elementId, worksetName}]. I workset vengono risolti solo in base al nome.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

set_instance_flip

Imposta FacingFlipped and/or HandFlipped su uno stato esatto su istanze di famiglia già posizionate. flipFacing()/flipHand() sono interruttori senza setter diretto in Revit API, quindi legge prima lo stato corrente e si attiva/disattiva solo quando non è d'accordo con la richiesta. Diverso dallo specchio action di modify_element: il mirroring trasforma la geometria (Mirrored=true) ma non imposta questi due flag. Anteprema-first: dryRun ha il valore predefinito true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ElementTools.cs

Famiglie

family_abort

Documenti familiari stretti SENZA saving: quello in fase di redazione, oppure tutti i documenti familiari presenti nella sessione con all=true. L'unico modo per chiudere una famiglia creata o aperta da una chiamata API: tale documento non ha finestra, quindi il menu Finestra non può raggiungerlo. Riporta cosa è stato chiuso, cosa presenta modifiche non salvate e cosa rimane aperto. Prima l'anteprema.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\FamilyTools.cs

family_artifact

Dimensioni e sha256 di un .rfa salvato, letti una volta che il file ha smesso di cambiare: l'identità verificata di un artefatto. family_save non può riportare it: Revit termina la scrittura dopo la restituzione della chiamata di salvataggio, quindi pubblica invece sha256_observed. Di sola lettura, non necessita di documenti.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\FamilyTools.cs

family_capabilities

Cosa può fare il modulo di creazione della famiglia su questo Revit: versione e build, quali documenti della famiglia Revit sono aperti (tutti, compresi quelli senza finestra) e quali sono in fase di creazione, dove risiedono i modelli .rft installati (prima N) e la funzionalità di ogni funzionalità di creazione, comprese le famiglie di sistema e sul posto, che API non espone. Sola lettura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\FamilyTools.cs

family_constraints_apply

Etichetta la distanza tra due piani di riferimento con una famiglia parameter: il collegamento parametrico. Il parametro quindi GUIDA la distanza, quindi i piani si spostano al suo valore corrente, riportato come planes_after. Solo le coppie possono essere viste da una vista in pianta (normali lungo x o y). Le lunghezze utilizzano l'unità (predefinita mm). Prima l'anteprima.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\FamilyTools.cs

family_create

Crea un documento di famiglia da un modello .rft installato e rendilo la famiglia in fase di creazione. templatePath è obbligatorio; family_capabilities elenca i modelli installati. Facoltativo: la categoria sovrascrive quella del modello, il nome imposta il nome della famiglia. La categoria e il posizionamento del modello vengono segnalati prima e dopo. Non viene scritto nulla sul disco fino a family_save. Sostituisce qualsiasi famiglia già in fase di creazione (segnalata come avviso).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\FamilyTools.cs

family_geometry_apply

Estrudere un rettangolo i cui quattro bordi sono allineati a quattro piani di riferimento denominati (x_min, x_max, y_min, y_max) su un piano di schizzo in sketch_z, con la profondità DRIVEN da una famiglia di lunghezze parameter: i bordi dell'estrusione sono allineati ai piani, quindi li segue quando un tipo cambia. L'opzione material_parameter lega il materiale del solido a un parametro Materiale. Una estrusione per famiglia. Le lunghezze utilizzano l'unità (predefinita mm). Prima l'anteprima.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\FamilyTools.cs

family_inspect

Esamina la famiglia in authoring: titolo, categoria, tipo di posizionamento, parametri (nome, tipo, istanza o tipo, formula), tipi e relativi valori, un inventario degli elementi e gli avvisi Revit. Sola lettura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\FamilyTools.cs

family_open

Apri un .rfa esistente come famiglia in fase di creazione. Un file già aperto viene adottato, non riaperto. L'apertura non scrive nulla. Prima l'anteprima.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\FamilyTools.cs

family_parameters_apply

Dichiara famiglia parameters: nome, tipo (Lunghezza, Numero, Testo, Intero, YesNo, Materiale, Angolo, Area, Volume), gruppo, istanza o tipo, formula opzionale. Idempotente; un tipo esistente diverso viene rifiutato, non costretto. I valori sono per type: utilizzare family_types_apply. Prima l'anteprima.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\FamilyTools.cs

family_references_apply

Dichiara i piani di riferimento per nome come (asse, offset) - lo scheletro su cui è quotata una famiglia parametrica. Idempotente per name: un piano esistente viene segnalato, mai spostato, perché la posizione di un piano etichettato è determinata dalla sua dimensione. Le lunghezze utilizzano l'unità (predefinita mm). Prima l'anteprima.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\FamilyTools.cs

family_save

Salvare la famiglia in fase di creazione su .rfa all'interno delle directory consentite dall'utente (o in una condivisione di rete). sovrascrivi (default false) sostituisce un file esistente e l'utente deve confermarlo. Prima l'anteprima. Riporta dimensioni e sha256.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\FamilyTools.cs

family_tests_run

Flex test: per ogni tipo, selezionalo, rigenera e MISURA il riquadro di delimitazione del solido rispetto a quello che dovrebbe essere quel tipo (x, y, z). Dimostra che la geometria segue i parametri: gli allineamenti registrati da family_geometry_apply non possono essere dimostrati senza spostare qualcosa. Modifiche nothing: viene eseguito il rollback della transazione. Sola lettura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\FamilyTools.cs

family_types_apply

Crea tipi di famiglia e imposta values: [{name, values:{parameter: value}}]. Le lunghezze utilizzano l'unità (predefinita mm), gli angoli sono gradi. I nomi dei parametri sconosciuti vengono rifiutati per nome. I tipi esistenti vengono riutilizzati, non duplicati. Prima l'anteprima.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\FamilyTools.cs

Relazioni geometriche

get_element_attachments

Leggere gli obiettivi di attacco superiore e base di un wall/column; le colonne restituiscono anche cutStyle, giustificazione e offsetMm. Le pareti richiedono Revit 2025.2+. Sola lettura; non deduce gli allegati dalle quote altimetriche corrispondenti.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\GeometryRelationshipTools.cs

get_geometry_join_state

Leggere lo stato dell'unione, l'elemento di taglio nell'unione e la cappetta direzionale in acciaio per una coppia esplicita. La coping non supportata restituisce null, non false. I tagli Solid/void sono independent: utilizzare check_steel_cut_eligibility, get_steel_cut_data o get_instance_void_cut_relationships per le API di taglio Revit generiche.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\GeometryRelationshipTools.cs

set_element_attachment

Attach/detach un'estremità del muro o della colonna verso una destinazione esplicita nel progetto attivo. Il muro API richiede Revit 2025.2+. Il valore predefinito dryRun esegue la convalida di sola lettura e restituisce awaitingUser con un planId monouso. Dopo aver chiesto all'utente, applicare gli stessi input con dryRun:false e planId entro cinque minuti; un documento modificato richiede una nuova anteprima. La conferma nativa e l'elaborazione degli errori si verificano al momento dell'applicazione. Non sostituisce mai automaticamente un allegato di colonna esistente; staccare prima il suo bersaglio esplicito. Le richieste ripetute nello stesso stato non sono operative. Le impostazioni delle colonne devono essere omesse per i muri e scollegate. Ispezionare il conteggio degli avvisi soppressi e il solido effettivo dopo l'applicazione.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\GeometryRelationshipTools.cs

set_geometry_join

Geometria Join/unjoin, impostare l'ordine di taglio delle giunzioni desiderato o la cappetta apply/remove. Una coppia esplicita. Coping: elementId è la trave in acciaio da tagliare; targetId la sua taglierina in acciaio beam/column. L'anteprima predefinita esegue la convalida di sola lettura e restituisce awaitingUser con un planId monouso. Dopo aver chiesto all'utente, applicare gli stessi input con dryRun:false e planId entro cinque minuti; un documento modificato richiede una nuova anteprima. La conferma nativa e l'elaborazione degli errori si verificano al momento dell'applicazione. set_order richiede cuttingElementId e non commuta mai alla cieca; lo stato desiderato ripetuto è un no-op. Non si tratta di un taglio solido-solido o di un'istanza vuota cut: utilizzare gli strumenti add/remove_steel_solid_cut e add/remove_steel_instance_void_cut esistenti per tali operazioni. Nessun accoppiamento automatico o inferenza materiale.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\GeometryRelationshipTools.cs

Gruppi e assieme

create_assembly

Creare un assieme Revit dagli ID elemento (Crea assieme di Revit). Validates/derives la categoria di denominazione e controlla i membri con le porte proprie di API. Anteprima-first: dryRun viene impostato automaticamente su true; dryRun=false crea. Una creazione di massa (più di elementi 20) richiede conferma.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\GroupAssemblyTools.cs

create_assembly_views

Creare una vista di assieme

(viewType=3d_orthographic|detail_section|material_takeoff|part_list|single_category_schedule|sheet) o copiare viste da un'altra istanza (viewType=acquire_views + sourceAssemblyId). detail_section prende la direzione (HorizontalDetail, DetailSectionA/B, ElevationTop/Bottom/Left/Right/Front/Back); il foglio necessita di titleBlockId; single_category_schedule necessita di scheduleCategoryId; opzionale viewTemplateId (+assignTemplate) assegna un modello. Anteprima-first: dryRun viene impostato automaticamente su true; dryRun=false crea. Additivo, quindi nessuna finestra di dialogo di conferma.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\GroupAssemblyTools.cs

`create_group`

Creare un gruppo di modelli dagli ID degli elementi (comando Gruppo di Revit). Il nome facoltativo imposta il nome del tipo di gruppo. Anteprema-first: `dryRun` viene impostato automaticamente su `true`; `dryRun=false` crea. Una creazione di massa (più di elementi 20) richiede conferma.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\GroupAssemblyTools.cs`

`dissolve_assembly`

Sciogliere un assembly: l'istanza dell'assembly viene eliminata, i suoi membri rimangono nel modello e vengono restituiti (`AssemblyInstance.Disassemble`); con esso vengono cancellati i punti di vista dell'assemblea, così come avviene la dissolvenza di Revit. Anteprema-first: `dryRun` viene impostato automaticamente su `true`; `dryRun=false` chiede conferma.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\GroupAssemblyTools.cs`

`get_assembly_data`

Leggere un'istanza di assembly Revit o un assembly type: nome, ID membro (facoltativamente con dettagli per membro ed espansione di gruppi nidificati), center/origin/transform in mm, categoria di denominazione, `allowsAssemblyViewCreation` e per un tipo le relative istanze. Sola lettura.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\GroupAssemblyTools.cs`

`get_element_containers`

Per ogni element id, riportare l'istanza del gruppo e l'assembly che lo contiene (`Element.GroupId` / `Element.AssemblyInstanceId`), con i relativi nomi. Sola lettura.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\GroupAssemblyTools.cs`

`get_group_data`

Leggi un Revit group: ID membro (`Group.GetMemberIds`), facoltativamente espanso in modo ricorsivo attraverso gruppi nidificati e con dettagli per membro, oltre a gruppi di dettagli allegati (disponibili; mostrati e compatibili per una determinata vista). Passa `groupId` (istanza) o `groupTypeId`: i membri di un tipo di gruppo sono leggibili solo attraverso una delle sue istanze. Sola lettura.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\GroupAssemblyTools.cs`

`modify_assembly`

Modificare un assembly esistente: `action=add_members|remove_members|set_members` (`elementIds`), `action=rinomina` (nome), `action=set_naming_category` (`namingCategoryId`). Utilizza i gate di API (`CanRemoveElementsFromAssembly`, `AreElementsValidForAssembly`, `IsValidNamingCategory`). Anteprema-first: `dryRun` viene impostato automaticamente su `true`; Si applica `dryRun=false`. Le modifiche di massa dei membri (più di elementi 20) richiedono conferma.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\GroupAssemblyTools.cs`

`modify_group`

Modifica un gruppo senza toccare il suo members: `action=rename` (`groupTypeId+name` o `groupId+name`) rinomina il tipo condiviso da tutte le istanze; `action=show_attached_details|hide_attached_details` (`groupId`, `viewId`, `detailGroupId`) o `show_all_attached_details|hide_all_attached_details` (`groupId`, `viewId`) attiva/disattiva i gruppi di dettagli allegati in una vista. Anteprema prima (`dryRun` predefinito `true`).

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\GroupAssemblyTools.cs`

`place_group`

Posiziona un'altra istanza di un tipo di gruppo di modelli esistente in `x/y/z` in millimetri. Anteprema-first: `dryRun` viene impostato automaticamente su `true`; `dryRun=false` posiziona l'istanza. Additivo, quindi nessuna finestra di dialogo di conferma.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\GroupAssemblyTools.cs`

`ungroup_group`

Separa un'istanza di gruppo e mantieni i suoi membri nel modello (`Group.UngroupMembers`). Elimina l'elemento del gruppo e restituisce gli ID dei membri rilasciati. I gruppi di dettagli allegati non possono essere separati. Anteprema-first: `dryRun` viene impostato automaticamente su `true`; `dryRun=false` chiede conferma.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\GroupAssemblyTools.cs`

IFC

`ifc_analyze_rebuildability`

Analizzare IFC `DirectShapes` e valutare la fattibilità di ricostruirli come elementi Revit nativi.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs`

`ifc_compare_original_vs_rebuilt`

Confronta il volume e la geometria (`volume/geometry`) del `DirectShape` originale con quelli della sua ricostruzione nativa.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs`

`ifc_export_basic`

Esporta il documento attivo in IFC. Le bandiere di prima classe coprono le opzioni comuni; utilizzare le sostituzioni per qualsiasi altra chiave `IFCExportOptions`.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs`

`ifc_export_with_configuration`

Esporta utilizzando una configurazione denominata (incorporata o personalizzata) con sostituzioni `key/value` opzionali.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs`

`ifc_get_capabilities`

Rileva il supporto della versione IFC e la presenza del componente aggiuntivo Revit-ifc

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

`ifc_get_export_configuration`

Ottieni i dettagli completi di una specifica configurazione di esportazione in base al nome.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

`ifc_link`

Collega un file IFC al documento attivo (crea un file collaterale .ifc.RVT gestito da Revit).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

`ifc_list_export_configurations`

Elenca le configurazioni di esportazione integrate disponibili

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

`ifc_list_rebuild_candidates`

Elenca gli elementi al di sopra di una soglia di confidenza di ricostruzione.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

`ifc_open_or_import`

Aprire o importare un file IFC come progetto Revit nativo (actions: apri | importa).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

`ifc_rebuild_family_instances`

Posiziona le istanze della famiglia (porte, finestre, mobili) da IFC DirectShapes.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

`ifc_rebuild_floors`

Ricostruisci i pavimenti nativi da IFC DirectShapes. Il valore predefinito di dryRun è true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

`ifc_rebuild_openings`

Tagliare le aperture in walls/floors ricostruito in base all'apertura IFC DirectShapes.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

`ifc_rebuild_roofs`

Ricostruisci i tetti nativi da IFC DirectShapes. Il valore predefinito di dryRun è true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

ifc_rebuild_structural_members

Ricostruisci colonne e travi da IFC DirectShapes. Il valore predefinito di dryRun è true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

ifc_rebuild_walls

Ricostruisci i muri nativi da IFC DirectShapes. Il valore predefinito di dryRun è true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

ifc_reload_link

Ricarica un collegamento IFC esistente, facoltativamente da un nuovo file.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

ifc_set_family_mapping_file

Imposta il file di mappatura della famiglia utilizzato dalle successive esportazioni IFC.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

ifc_tag_unreconstructable_elements

Tag IFC DirectShapes che non può essere ricostruito scrivendo un parametro marker.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

ifc_validate_request

Convalida il percorso del file IFC, l'estensione e la versione dello schema.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\IfcTools.cs

Collegamenti

add_linked_file

Aggiunge un nuovo file collegato Revit da un percorso di file e facoltativamente inserisce un'istanza nella posizione specificata.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\LinkTools.cs

align_link_to_host

Allinea un'istanza di collegamento all'origine interna del progetto host, alle coordinate condivise o al punto base del progetto.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\LinkTools.cs

cad_link_cleanup

Analizza e ripulisci i file imported/linked CAD. action=list|delete.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\LinkTools.cs

`get_coordination_models`

Elenco di sola lettura dei modelli di coordinamento Autodesk Revit con metadati del tipo e istanze facoltative.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\LinkTools.cs

`get_link_transform`

Restituisce la trasformazione completa di un'istanza di file collegato.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\LinkTools.cs

`get_linked_file_instances`

Elenca i modelli Revit COLLEGATI (collegamenti RVT) in questo progetto, raggruppati per tipologia, con stato di caricamento e percorso. Utilizzare per "quali collegamenti/modelli collegati sono presenti in questo progetto". La risposta snella predefinita fornisce gli ID delle istanze e lo stato bloccato; per la trasformazione del posizionamento di un collegamento utilizzare `get_link_transform` o passare `verbose=true`.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\LinkTools.cs

`get_selected_linked_elements`

Restituisce informazioni sulle istanze di collegamento attualmente selezionate.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\LinkTools.cs

`highlight_linked_element`

Evidenzia un elemento all'interno di un modello collegato con un riquadro di sezione opzionale.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\LinkTools.cs

`manage_links`

Elenca, ricarica, scarica o rimuovi file collegati. Per aggiungere un NUOVO collegamento utilizzare invece `add_linked_file`.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\LinkTools.cs

`move_link_instance`

Sposta un'istanza di file collegato. `mode=delta` si applica (x,y,z) come offset; `mode=assoluto` posiziona l'origine in (x,y,z). I valori sono espressi in mm.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\LinkTools.cs

`pin_unpin_link_instance`

Blocca o sblocca istanze di file collegati.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\LinkTools.cs

`reload_linked_file_from`

Ricarica un file Revit collegato da un percorso file diverso.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\LinkTools.cs

`show_cross_model_elements`

Seleziona gli elementi host più gli elementi nei modelli Revit collegati. La strategia predefinita crea marcatori rossi DirectShape attorno a ciascun elemento collegato (sincrono, robusto). `usePostCommandIsolate=true` utilizza invece `IsolateElements` nativo di Revit: asincrono e non può essere combinato con box di sezione/sostituzioni nella stessa chiamata.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\LinkTools.cs`

Materiali

`create_material`

Crea un nuovo materiale nel progetto Revit.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\MaterialTools.cs`

`delete_material`

Elimina un materiale dal progetto tramite ID o nome.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\MaterialTools.cs`

`duplicate_family_type`

Duplica un tipo di famiglia caricabile con un nuovo nome e sostituzioni di parametri facoltativi.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\MaterialTools.cs`

`duplicate_material`

Duplica un materiale esistente con un nuovo nome.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\MaterialTools.cs`

`get_compound_structure`

Ottieni la struttura dei livelli wall/floor/roof/ceiling per tipo ID o nome.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\MaterialTools.cs`

`get_material_catalog_status`

Confronta un file del catalogo dei materiali con i materiali nel modello aperto e segnala la differenza, per CAMPO. Read-only: non viene scritto nulla, quindi è sicuro eseguirlo prima di qualsiasi importazione. Accetta un CSV con una colonna 'name' più uno qualsiasi degli `manufacturer/model/description/url/cost/mark/keynote/comments` (',' o ';' separati, entrambi rilevati) o JSON con un array 'materials'. Leggi `fieldSummary`first: `change`` conta i valori che il modello ha GIÀ che differiscono dal catalogo - quelli che un'importazione sostituirebbe - mentre `fill`` conta i campi vuoti. Una cella di catalogo vuota significa "non descritto", mai "cancellarlo". I nomi corrispondenti a diversi materiali sono riportati in `ambiguous`` e saltati anziché indovinati.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\MaterialTools.cs`

`get_material_properties`

Ottieni proprietà dettagliate del materiale per materiale ID o name: aspetto, risorse strutturali e termiche, oltre agli otto campi di identità commerciale (descrizione, produttore, modello, URL, costo, marchio, nota chiave, commenti) sotto le stesse chiavi scrive `set_material_properties`, in modo che la risposta possa essere restituita invariata. Un campo vuoto è null; un campo Revit non contiene questo materiale è elencato in `identityUnavailable`.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\MaterialTools.cs`

`get_material_quantities`

Calcola l'area del materiale e il volume degli elementi, opzionalmente filtrati per categoria o limitati alla selezione corrente.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\MaterialTools.cs`

`get_materials`

Elenca i materiali del documento attivo Revit, opzionalmente filtrati per classe o nome.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\MaterialTools.cs`

`import_material_catalog`

Applicare un catalogo materiali (CSV/JSON) ai materiali del modello, scrivendo produttore, modello, descrizione, URL, costo, marchio, nota chiave, commenti. LA MODALITÀ È A SCELTA DEL CHIAMANTE: `mode='fillEmpty'` (default) scrive solo dove il campo modello è vuoto e quindi non può perdere dati; `mode='overwrite'` sostituisce anche i valori già presenti nel modello. Il valore predefinito è `fillEmpty` a meno che l'utente non abbia affermato che il catalogo vince - misurato case: un modello conteneva la nota chiave "09 29 00.D1" dove il catalogo aveva la meno specifica "09 29 00", quindi il catalogo non è automaticamente la fonte migliore. Una cella di catalogo vuota non scrive mai in nessuna delle due modalità. ``cost`` NON viene scritto di default anche quando il catalogo riporta it: un prezzo finisce nel parametro Costo Revit e da lì nei palinsesti e nei fogli consegnati, quindi deve essere richiesto - passa `fields:["cost", ...]`. Utilizzare ``fields`` per restringere il resto. Anteprema-first: `dryRun` per impostazione predefinita è `true` e `get_material_catalog_status` mostra la stessa differenza gratuitamente. I nomi che corrispondono a più materiali vengono rifiutati, mai indovinati.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\MaterialTools.cs`

`list_material_parameters`

Elenca i parametri effettivamente trasportati dai materiali del modello (una riga per parametro, sia nativo che SHARED) con il tipo di archiviazione, se può essere scritto e su quanti materiali è vuoto. Usalo per decidere QUALI parametri arricchire invece di assumere un set fisso: misurato su un modello reale un materiale porta parametri 44, quindi gli otto commerciali che restituisce `get_material_properties` sono un sottoinsieme. Vengono riportati due conteggi di vuoto fianco a fianco: ``blankOn`` (test di questo strumento: testo vuoto o zero numerico) e ``hasValueOn`` (il verdetto di Revit) - perché i documenti Revit non dicono cosa restituisce `HasValue` per un parametro di testo che contiene "", e un disaccordo tra i due è informazione, non rumore. Sola lettura. Quindi utilizzare `get_parameter_values_by_category` (OST_Materials, `includeEmpty`) per vedere QUALI materiali lasciano vuoto un parametro scelto.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\MaterialTools.cs`

set_compound_structure

Modifica la struttura composta su un tipo wall/floor/roof/ceiling. action=replace|add|remove|modify|set_wrapping. set_wrapping imposta openingWrapping (none|exterior|interior|both), endCap (none|exterior|interior) e layerWrapping per strato.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\MaterialTools.cs

Informazioni del server

analyze_send_code_gaps

Raggruppa ogni chiamata send_code_to_revit registrata nel registro di controllo in base al relativo campo `purpose`, in modo che un divario di capacità che continua a ricorrere venga visualizzato come conteggio invece di rimanere invisibile nelle chiamate una tantum. Il raggruppamento avviene solo con la corrispondenza esatta del testo (rifilato, senza distinzione tra maiuscole e minuscole): non suppone che due scopi formulati in modo diverso descrivano lo stesso divario; leggi le voci grezze di quella sentenza. Di sola lettura, nessun documento richiesto.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\MetaTools.cs

clear_cache

Cancella ogni voce dalla cache dei risultati dello strumento lato plug-in.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\MetaTools.cs

flow_step

Dichiara in quale fase di un flusso registrato ti trovi. Restituisce l'involuppo del flusso (flusso, flowStep, flowStepCount) E rende visibile il passaggio nella barra multifunzione Revit, l'unico livello che nessun client MCP può rifiutare. Chiamalo all'inizio di OGNI passaggio compreso l'ultimo, prima che lo strumento esegua il lavoro di quel passaggio; passare awaitingUser:true con la domanda ad un cancello. Quindi, quando il flusso termina - applicato, niente da fare, rifiutato o abbandonato - chiamalo ANCORA ANCORA con complete:true, in modo che l'indicatore non rimanga acceso su una decisione presa. Quella chiamata di chiusura NON inserisce un step: inviare complete:true INVECE di annunciare l'ultimo passaggio nasconderebbe l'indicatore prima ancora che il lavoro di quel passaggio inizi. Due chiamate all'ultimo passaggio, non una. Non tocca i dati del modello e non necessita di documenti aperti.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\MetaTools.cs

get_cache_stats

Restituisce la telemetria diagnostica hit/miss dalla cache dei risultati dello strumento lato plug-in.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\MetaTools.cs

get_project_info

Ottieni nome, indirizzo, livelli, fasi, workset e collegamenti del progetto dal documento Revit attivo.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\MetaTools.cs

`get_session_info`

Quali pezzi Pulse sono in realtà running: timestamp di build di plug-in e server, conteggio degli strumenti, profilo (sviluppo vs produzione e relativa porta), locale e se un documento è rilegato. Chiamalo PRIMA quando uno strumento manca dall'elenco o si comporta come se una correzione fosse absent: un divario nel timestamp significa una distribuzione obsoleta, non uno strumento danneggiato. Non ha bisogno di documenti aperti.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\MetaTools.cs

`say_hello`

Testare la connessione di MCP a Pulse. Visualizza un saluto in Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\MetaTools.cs

`set_project_info`

Imposta campi modificabili delle informazioni sul progetto. Vengono modificati solo i campi che attraversi; altri rimangono intatti.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\MetaTools.cs

Parametri

`add_prefix_suffix`

Aggiungere un suffisso and/or ai valori dei parametri nel modello o in una selezione. Viene eseguito come anteprima di prova per impostazione predefinita; impostare `dryRun=false` per applicare le modifiche.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ParameterTools.cs

`add_shared_parameter`

Aggiungi un parametro condiviso alle categorie di progetto. Viene rispettato il tipo di dati di una definizione appena creata (un parametro condiviso digitato, non sempre testo). Viene eseguito come anteprima di prova per impostazione predefinita; impostare `dryRun=false` per applicare la rilegatura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ParameterTools.cs

`bulk_modify_parameter_values`

Imposta o modifica il valore di un parametro su MOLTI elementi contemporaneamente (modifica in blocco/trova e sostituisci) - e.g. impostare Commenti, Contrassegni o un parametro condiviso in un'intera categoria o nella selezione. Per un singolo elemento utilizzare `set_element_parameters`.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ParameterTools.cs

`clear_parameter_values`

Cancella i valori dei parametri sugli elementi per categoria o ambito

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ParameterTools.cs

`export_shared_parameter_file`

Esporta il parametro condiviso file: restituisce i suoi gruppi e definizioni e ne scrive una copia in filePath quando ne viene fornito uno.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ParameterTools.cs

`filter_by_parameter_value`

Trova o elenca gli elementi in base al parametro VALORE: una condizione o più condizioni combinate con AND/OR tramite l'array delle condizioni. Condizioni: uguale, not_equals, contiene, not_contains, begins_with, ends_with, greater_than, less_than, is_empty, is_not_empty. Da utilizzare per 'porte tagliafuoco', 'muri di tipo X', 'elementi privi di Marchio', 'locali con area > N'. Per i parametri stringa a livello di tipo (e.g. nome tipo) impostare parameterType="type" - parameterType="both" potrebbe non risolverli. Per trovare per CATEGORIA utilizzare solo ai_element_filter.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ParameterTools.cs

`manage_global_parameters`

Gestire i parametri globali (valori denominati a livello di progetto). Azioni: elenco | ottenere | creare | impostare | elimina | rinominare | set_formula | move_up | move_down | ordinare. A differenza dei parametri project/shared, i parametri globali POSSONO essere rinominati. Per impostazione predefinita, le azioni di scrittura sono preview: passare dryRun=false per applicare.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ParameterTools.cs

`manage_project_parameters`

Gestire i parametri del progetto. Azioni: elenco | creare | elimina | modificare | set_group | set_binding_type | rinominare. 'delete' rimuove anche i parametri non condivisi (risolve il bug Revit REVIT-136670). Revit consente a due parametri di progetto di condividere un nome visualizzato, quindi delete/modify/set_binding_type restituisce AmbiguousInput con i candidati invece di selezionare one: passa parameterId (da 'list' o da error.context.candidates) per nominarne esattamente uno. 'create' crea un parametro SHARED e lo dice - Revit API non può crearne uno nuovo non condiviso - con un GUID casuale il cui file di supporto viene scartato, quindi nessun FILE di parametro condiviso contiene quella definizione - il modello does: misurato, il guid sopravvive save/close/reopen e action:list lo riporta, ma niente portatile lo trasporta. Mantieni il guid se un altro modello deve mai assumere la stessa identità, perché add_shared_parameter per lo stesso nome ne conia uno diverso e divide quello che sembra un singolo parametro. Utilizzare add_shared_parameter dall'inizio quando deve essere riutilizzabile su tutti i modelli. Chiamando 'create' due volte con un nome si creano DUE parametri con quel nome, ed è così che nasce l'ambiguità di cui sopra; la risposta avvisa quando il nome è già associato. Ogni azione di scrittura viene visualizzata in anteprima per impostazione predefinita: vedere il parametro dryRun per sapere esattamente quali la leggono. 'rename' NON è supportato per un progetto associato parameter: Definition.Name è di sola lettura, quindi l'azione fallisce sempre con PermissionDenied e restituisce l'unica soluzione alternativa esistente: creare il nuovo nome, leggere i vecchi valori con export_elements_data, riscriverli sotto il nuovo nome con sync_csv_parameters, verificarli e solo dopo eliminare il vecchio parametro; l'ultimo passaggio è irreversibile, quindi non eseguirlo sulla base del successo della segnalazione di scrittura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ParameterTools.cs

set_element_parameters

Imposta i valori dei parametri su uno o più elementi. Passa le richieste come una stringa di array con codifica JSON. Supporta parameterName per nome visualizzato e builtInParameter per Revit API nome enum, e.g.

REBAR_SYSTEM_SPACING_TOP_DIR_1. UNITÀ: un valore numerico su length/area/etc. Il parametro è scritto in unità interne (piedi) Revit - per impostare un valore visualizzato, passare una STRINGA CON UN'UNITÀ (e.g. "3000 mm", "3 m") e Revit lo analizza in base all'unità e alla locale. Un valore null cancella il parametro. Nota: i parametri a livello di tipo possono essere impostati tramite un'istanza quando il tipo è risolvibile.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ParameterTools.cs

sync_csv_parameters

Sincronizza i valori dei parametri dai dati CSV negli elementi Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ParameterTools.cs

transfer_parameters

Copia i valori dei parametri dall'elemento di origine a uno o più elementi di destinazione.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ParameterTools.cs

Nuvole di punti

pointcloud_clear_filter

Rimuovi il filtro di selezione di una nuvola di punti e imposta la sua azione su nessuno, in modo che smetta di essere evidenziata o isolata. Lo stato viene riletto. Antepima-first: dryRun viene impostato automaticamente su true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs

pointcloud_delete

Elimina un'istanza della nuvola di punti. Irreversibile. Anche deleteUnusedType (predefinito false) elimina il tipo, ma solo quando nessun'altra istanza lo utilizza. Un'istanza bloccata viene rifiutata a meno che unpinTemporarily non sia true, perché Revit non eliminerà un elemento bloccato. Antepima-first: dryRun viene impostato automaticamente su true e l'antepima riporta gli elementi che verrebbero rimossi con esso.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs

pointcloud_detect_openings

Trovare le aperture (finestre, porte) in un wall rilevato: regioni di una facciata in cui non sono stati restituiti punti mentre la muratura attorno ad esse è coperta. Necessita di averageDistance più referenceNormal/referencePoint (piano del muro); regione casella opzionale, slabThickness (0.3), cellSize (0.05), minOpeningWidth/minOpeningHeight (0.4). Ogni apertura restituisce il proprio rettangolo nel telaio del muro (u orizzontale, w UP) e i propri angoli nelle coordinate del modello. Il rettangolo è il NUCLEO vuoto: fino a una cella più piccola per lato. Di sola lettura, basato su campioni.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs

`pointcloud_detect_planes`

Rileva superfici planari in una regione campionata (RANSAC, deterministico). Richiede `averageDistance` (la risoluzione di campionamento) e `tolleranza` (quanto vicino deve essere un punto per contare come sul piano); opzionale `minPointsPerPlane` (50), `maxPlanes` (5), iterazioni (500) e una regione `box` (default: estensione della nuvola). Ogni piano viene fornito con la sua normale, inclinazione, conteggio e frazione inlier e l'estensione degli inlier. Sola lettura, campione-based: la risposta indica la busta all'interno della quale sono validi i numeri.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs`

`pointcloud_floor_levelness`

Misurare il livello di un pavimento o di un solaio da una regione campionata. Richiede `averageDistance` e `tolleranza`; l'opzione `designElevation` misura rispetto al progetto anziché al piano adattato e `outlierTolerance` segnala la quantità di campione che si trova oltre tale piano. Regione del riquadro opzionale (default: estensione della nuvola). Riporta l'inclinazione e l'elevazione del piano adattato oltre alle statistiche di deviazione con segno. Di sola lettura, basato su campioni.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs`

`pointcloud_get`

Leggi una nuvola di punti in full: tipo, motore, origine, trasformazione di posizionamento, trasformazione totale, estensione, scansioni, regioni. Entrambe le trasformazioni portano `uniform_scale`, che su una nuvola ReCap è la scala del tipo, non 1. Selezionare per `elementId`, `uniqueId` o nome; ai candidati viene rifiutato un nome ambiguo. Sola lettura, nessuna transazione.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs`

`pointcloud_list`

Elenca le istanze e i tipi di nuvola di punti nel documento attivo, con posizionamento opzionale e conteggi totali di trasformazioni, limiti, scansioni e regioni. Sola lettura, nessuna transazione; un file di origine mancante è un avviso per elemento, non un errore. Le lunghezze sono in `lengthUnit`.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs`

`pointcloud_list_regions`

Elenca le regioni ReCap di una nuvola di punti. I nomi delle regioni rappresentano l'handle utilizzato dagli strumenti di override per regione. Sola lettura, nessuna transazione; un elenco vuoto significa che il cloud non definisce regioni.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs`

`pointcloud_list_scans`

Elenca le scansioni all'interno di una nuvola di punti, facoltativamente con le origini. Revit identifica una scansione in base al nome, quindi `name` è l'handle restituito. Sola lettura, nessuna transazione; un'origine illeggibile è un avviso su quella scansione.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs`

pointcloud_load

Carica un'origine nuvola di punti (percorso ReCap .rcp/.rcs o una stringa di identificazione per un motore non di file) e posiziona un'istanza. Il motore è verificato rispetto al registro di Revit. Anteprima-first: dryRun viene impostato automaticamente su true. I percorsi locali devono trovarsi nelle directory dell'utente; sono consentite condivisioni di rete.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs

pointcloud_recap_project_info

Leggi la visione di ReCap SDK del progetto sorgente di una nuvola di punti attraverso un adattatore riflettente OPZIONALE che carica nothing: riporta il tipo SDK trovato, l'inventario dei membri che espone realmente e i valori che potrebbe estrarre in modo inequivocabile (ciascuno nominando il proprio membro). Se SDK non è installato, la risposta lo dice e nomina l'assembly. Sola lettura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs

pointcloud_reset_overrides

Ripristina la sostituzione grafica di una nuvola di punti in una vista ai valori predefiniti per un scope: visibile, nessuna sostituzione del colore. viewId è obbligatorio; l'ambito è l'istanza (predefinita), la scansione o la regione, con scopeName per gli ultimi due. Viene toccato solo l'ambito richiesto. Anteprima-first: dryRun viene impostato automaticamente su true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs

pointcloud_sample

Punti campione da una nuvola di punti a una spaziatura target, facoltativamente all'interno di un volume indicato come piani. Restituisce un SAMPLE, mai l'intero cloud. output: riepilogo (count/bounds/centroid), in linea (punti, fino a limit in linea), artefatto (CSV/PLY/JSONL nella sandbox Pulse). Sola lettura, nessuna transazione.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs

pointcloud_sample_around_element

Punti campione di una nuvola di punti attorno a uno o più elements: l'unione delle loro estensioni allineate all'asse, espanse mediante riempimento. Restituisce un SAMPLE, mai l'intero cloud. Gli elementi senza estensione utilizzabile vengono segnalati come avvisi. Sola lettura, nessuna transazione.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs

pointcloud_sample_box

Punti campione di una nuvola di punti all'interno di un volume box (centro, dimensione, rotazione opzionale attorno a Z). Restituisce un SAMPLE, mai l'intero cloud. output: riepilogo, in linea (fino a limit in linea) o artefatto (CSV/PLY/JSONL nella sandbox Pulse). Sola lettura, nessuna transazione.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs

pointcloud_set_filter

Applica un filtro di selezione ad una nuvola di punti e imposta action: evidenzia i punti all'interno del volume, oppure isolali. il filtro è {kind:'box',box:{...}} o {kind:'planes',planes:[...]} nelle coordinate del modello e lengthUnit - accetta gli stessi volumi pointcloud_sample_box. Revit NON preserva l'azione del filtro quando il documento viene salvato. Antepima-first: dryRun viene impostato automaticamente su true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs

pointcloud_set_override

Imposta l'override grafico di una nuvola di punti PER UNA VISTA: visibilità e colorazione, per l'intera istanza, una scansione o una regione. È richiesto viewId (una vista del modello, non un modello). color_mode è no_override, fisso, elevazione, normali o intensità; esigenze fisse fixedColor {r,g,b}; elevation/intensity necessita del gradiente {color1,color2}. Non viene toccato nulla al di fuori dell'ambito richiesto. Antepima-first: dryRun viene impostato automaticamente su true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs

pointcloud_set_transform

Sposta e ruota un'istanza della nuvola di punti. la modalità assoluta sostituisce il posizionamento, relativa applica la trasformazione sopra quella corrente. trasformazione è il modulo dell'oggetto o l'array principale di riga del numero 16 restituito da pointcloud_get, traduzione in lengthUnit. La sua scala uniforme deve essere 1 o quella dell'istanza (il viaggio di andata e ritorno): un'istanza di nuvola di punti non può essere ridimensionata o specchiata. Un'istanza bloccata richiede unpinTemporarily. Antepima-first: dryRun viene impostato automaticamente su true. Il posizionamento viene riletto dopo la modifica e ripristinato se Revit non è d'accordo.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs

pointcloud_wall_deviation

Misura la distanza di una regione campionata da un riferimento plane: il muro modellato quando vengono forniti referenceNormal e referencePoint (un punto SU quel piano, in lengthUnit), altrimenti un piano adattato ai punti. Richiede averageDistance e tolleranza; outlierTolerance e regione box opzionali. Riporta le statistiche di deviazione con segno e su quale lato cadono i punti. Di sola lettura, basato su campioni.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\PointCloudTools.cs

Progetto

analyze_model_statistics

Analizzare i conteggi degli elementi per categoria nel documento Revit attivo.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`analyze_warning_fixability`

Avvertenze sul modello di triage per la correzione MANUALE. Per gruppo di avvisi aggiunge a `get_warnings`: una classificazione di possibilità di correzione (automatizzabile | `needs_design_decision` | bloccato) CON il motivo, più per elemento la categoria, `family/type`, livello e cosa blocca una modifica (appuntato, appartenenza al gruppo, `CanDeleteElement`, super componente). NON RISOLVE NULLA: dice all'utente dove andare in Revit e quali avvisi sono decisioni di progettazione che solo lui può prendere. La classificazione corrisponde al testo descrittivo in INGLESE; un Revit localizzato riporta 'unknown' anziché indovinare. Costo: un passaggio di avviso più una lettura del parametro per elemento. `includeViews=true` individua anche le viste di ciascun elemento ed è MOLTO più lento (un raccoglitore con ambito vista per vista, che può forzare una ricostruzione della geometria) - disattivato per impostazione predefinita.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`audit_families`

Famiglie di audit nel progetto Revit. Elenca le famiglie caricabili (.rfa) per impostazione predefinita; impostare `includeSystemFamilies=true` per elencare anche i tipi di famiglie di sistemi (tipi wall/floor/roof/ceiling, incluso solaio di fondazione).

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`batch_rename`

Rinominare in batch elementi o tipi di sistema nel progetto Revit. Supporta sia elementi della famiglia caricabile che tipi di sistema (tipi wall/floor/ceiling/roof).

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`check_model_health`

Esegui un controllo dello stato del modello e restituisci un punteggio di integrità.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`clash_detection`

Rileva le collisioni tra due categorie di elementi. Utilizza l'intersezione della geometria solida true per impostazione predefinita (meno positivi false rispetto ai riquadri di delimitazione).

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`create_custom_voce`

Crea una voce personalizzata per il prezzo diretto (codice + descrizione + unità + prezzo, nessuna suddivisione) nella libreria personale 'Le mie voci', riutilizzabile su projects/computi. Anteprema-first: `dryRun` ha il valore predefinito true e non scrive nulla.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`create_level`

Crea, modifica, rinomina o elimina un livello. `action=create|set|rename|delete`. Per `set/rename/delete` identificare il livello con `levelId` o con il nome.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`create_nuovo_prezzo_analisi`

Compone un Nuovo Prezzo (NP) tramite l'analisi del prezzo (risorse elementari + spese generali% + profitto%, profitto su costi+spese generali) e lo salva come voce personalizzata in 'Le mie voci'. Strumento di scrittura; dryRun default true (anteprima della ripartizione e del codice che verrebbe assegnato, non scrive nulla).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`create_room`

Crea una nuova stanza nel progetto Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`delete_custom_voce`

Cancella una voce dalla libreria personale 'Le mie voci' tramite codice esatto. Anteprima-first: dryRun imposta automaticamente true e non elimina nulla. Non influisce sulle assegnazioni già registrate tramite `record_computo_voce`.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`delete_element`

Elimina elementi dal progetto Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`duplicate_system_type`

Duplicare, rinominare o eliminare un tipo di sistema (muro, pavimento, tetto, soffitto).
`action=duplicate|rename|delete`. `action=delete` è l'anteprima prima (dryRun per impostazione predefinita è true) e riporta gli elementi dipendenti - istanze del tipo - che andrebbero con esso; duplicare e rinominare applicare immediatamente.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`export_elements_xpwe`

Costruisci un computo PriMus XPWE direttamente da model: una riga di misurazione per istanza, raggruppata per parametro (il tipo per impostazione predefinita), con una descrizione della riga componibile e le quattro dimensioni PriMus mappate su parametri o costanti. Legge lo stesso Revit (categoria o selezione corrente). Compila i campi dei tag WBS/BIM. Anteprima-first: dryRun ha il valore predefinito true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`export_quantity_mapping`

Esporta la mappatura approvata tra le voci del listino prezzi e Revit types/materials su un supporto durevole CSV. Le righe riportano il parametro di confronto scelto e un valore approvato per target mappato. Prima l'anteprima.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

export_schedule_xpwe

Costruisci un computo PriMus XPWE da una SCHEDULA ESISTENTE: legge le sue colonne e righe visibili e le mappa sui ruoli XPWE. I nomi di colonna convenzionali (Codice/Descrizione/Unità di misura/Quantità/WBS, IT o EN) vengono rilevati automaticamente, ovvero il modello di pianificazione 'XPWE-ready'; altrimenti passare columnMapping. Viene segnalata una colonna mappata mancante nella pianificazione, mai ignorata. Anteprima-first: dryRun ha il valore predefinito true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

export_xpwe

Esporta le righe assemblate del chiamante in un file PriMus XPWE (documentType 'pricelist' o 'computo'). Questo strumento serializza ciò che passi; NON interroga Revit né decide associazioni. Compila i campi tag WBS (codiceWBSCAP/codiceWBS) e tag BIM (tagBIM). Anteprima-first: dryRun ha il valore predefinito true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

get_available_family_types

Elenca i tipi di famiglia disponibili nel progetto Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

get_computo_cam_summary

Calcola la quota CAM (Criteri Ambientali Minimi) di un computo già scritto: aggrega per importo (quantità × prezzo) in bucket CAM / non CAM / non classificati, dal capitolo del codice articolo. Sola lettura; NON necessita di documenti aperti (legge il file .xpwe). Non decide la conformità.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

get_computo_status

Leggere il sidecar dell'archivio di lavoro Computo (<nome>.computo.json) per RIPRENDERE un calcolo metrico tra sessioni senza perdere la ricetta di misurazione o decisions: riporta la ricetta (unità, ambito, parametri di dimensione, flag derivati), progresso per ambito (done/pending), conteggi delle decisioni e righe bloccate, oltre a un suggerimento di ripresa. Sola lettura; NON necessita di documenti aperti (legge un file).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

get_computo_totals

Legge i quattro livelli monetari di un computo (costo diretto, prezzo con ricarico applicato, totale imponibile, totale compreso VAT) da un .xpwe già esportato più le percentuali impostate tramite set_computo_percentages. Solo file, sola lettura; nessun documento Revit richiesto.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

get_element_ownership

SOLO MODELLI CONDIVISI: chi possiede, chi ha creato e chi ha modificato per ultimo un insieme di elementi. Per elemento restituisce checkoutStatus (NotOwned | OwnedByCurrentUser | OwnedByOtherUser), il proprietario, modelUpdatesStatus rispetto al centrale e, a meno che statusOnly, il creatore, l'ultimo utente che ne ha salvato una modifica nel centrale e chiunque abbia una richiesta di modifica in sospeso. Inoltre esegue la configurazione per utente (byOwner/byCreator/byLastChangedBy), quindi "chi lavora in quest'area" è una chiamata. IT RISPONDE CHI, MAI QUANDO: il worksharing Revit API non riporta alcun timestamp di alcun tipo, quindi non è possibile dedurre alcuna data da questa risposta - lo dice il payload in limits.timestampNote. I valori sono la cache locale di Revit central e potrebbero essere obsoleti; servono per riferire a una persona, non per decidere se una scrittura avrà successo. È NECESSARIO un ambito (elementIds o categoryFilter, non entrambi) (non spazza mai l'intero modello) e maxElements è limitato a 1000 perché il livello di dettaglio completo alloca un oggetto nativo per elemento senza costi documentati. Rifiuta un modello non condiviso, dove la proprietà non esiste. Sola lettura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

get_incidenza_manodopera

Legge un computo .xpwe già esportato e aggrega l'incidenza del costo del lavoro (art. 41 cc.13-14 D.Lgs 36/2023) per voce e sul totale dei lavori. Sola lettura; NON necessita di documenti aperti (legge un file). Il valore predefinito di groupBy è "codice" (provvisorio: vedere la descrizione dello strumento).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

get_parameter_values_by_category

Istogramma dei valori di un parametro in una categoria in un singolo call: ogni valore distinto con il relativo conteggio e se è stato letto dall'istanza o dal tipo. Sostituisce la sequenza get_element_parameters + export_elements_data + filter_by_parameter_value. La scansione è esaustiva quindi i conteggi sono esatti; viene limitato solo l'ELENCO dei valori distinti e viene segnalato il troncamento.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

get_phases

Elenca tutte le fasi del progetto nel documento Revit attivo.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

get_project_coordinates

Dove si trova il modello nello world: punto base del progetto, punto di rilevamento e origine interna, ciascuno con la sua posizione relativa all'origine interna E la sua posizione condivisa, più le coordinate condivise dell'origine interna, l'angolo rispetto al nord reale, la posizione del progetto attivo e il sito latitude/longitude. Risposte "dov'è questo modello" per un passaggio di rilevamento, una federazione di modelli o un controllo georeferenziato IFC. Lunghezze in mm (coordinate condivise anche in metri e piedi interni grezzi), angoli in GRADI: Revit API restituisce piedi decimali e radianti, e questo strumento li converte. Sola lettura. Le tre quantità NON sono intercambiabili e vengono restituite separatamente: posizione (relativa all'origine interna), sharedPosition (sistema condiviso) e le coordinate condivise dell'origine interna, che equivalgono alla posizione condivisa del punto base solo quando il punto base si trova sull'origine.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`get_project_units`

Dichiara il valore units del documento: secondo le specifiche misurabili, Forge specTypeId/unitTypeId, precisione, simbolo di unità e flag di formattazione, oltre a unitSystem (Metric|Imperial) e i simboli decimali e di raggruppamento delle cifre. Chiamatelo PRIMA di interpretare qualsiasi numero dimensionale: la stessa suite restituisce millimetri, piedi interni e metri quadrati a seconda del campo.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`get_shared_parameters`

Elenca tutti i parametri del progetto con i relativi vincoli e categorie, facoltativamente filtrati per categoria.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`get_warnings`

Ottieni avvisi sui modelli dal documento Revit attivo. Gli avvisi sono RAGGRUPPATI per tipo di errore in base a default: warningGroups fornisce una riga per classe di problema con warningCount e affectedElementCount, ordinati per primi, quindi gli avvisi 49 vengono letti come ~25 problemi distinti. Viene comunque restituito l'elenco semplice 'warnings'. matchedWarnings è il numero true che supera i filtri e 'truncated' indica se la pagina piatta è stata ricoperta.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`get_worksets`

Elenca i workset del documento Revit attivo, con stato open/editable, proprietario e conteggio degli elementi. Solo workset utente a meno che non sia impostato includeSystemWorksets.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`ingest_pricelist`

Inserisci un listino prezzi regionale (PriMus XPWE) una volta in un indice locale ricercabile in ~/.pulse/pricelists. Quindi utilizzare search_pricelist. Non legge il modello Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`lines_per_view_count`

Conta le linee di dettaglio per vista (passaggio di un singolo documento, sicuro su qualsiasi dimensione del modello) più un conteggio delle linee del modello a livello di progetto. Le linee del modello non hanno una vista proprietaria, quindi vengono segnalate una volta a livello di progetto, non per vista.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`list_custom_voci`

Elenca la libreria personale "Le mie voci" di voci a prezzo diretto scritte a mano, facoltativamente filtrate da una query di sottostringa. Sola lettura; non legge il modello Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`list_family_sizes`

Elenca le famiglie caricate con conteggi type/instance e, quando includeSize=true, la dimensione del file della famiglia in KB misurata esportando ciascuna famiglia in un file temporaneo. Disabilitato per impostazione predefinita perché l'esportazione è lenta sui modelli con molte famiglie.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`list_pricelists`

Elenca tutti i listini prezzi già acquisiti tramite ingest_pricelist (ID, regione, anno, conteggio articoli, quando è stato creato). Usalo per scoprire un prezzariold valido prima di chiamare search_pricelist. Sola lettura; non legge il modello Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`list_schedulable_fields`

Scopri i campi pianificabili disponibili per una categoria.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`load_family`

Carica una famiglia nel progetto Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`manage_additional_settings`

Gestisci impostazioni aggiuntive (scheda Gestisci): stili di linea, spessori di linea, modelli di linea, modelli di riempimento, halftone/underlay.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`manage_phase_filters`

Elenca, imposta o crea filtri di fase Revit. 'set' modifica una presentazione (Nuovo | Demolito | Esistente | Temporaneo) su un filtro, preservando gli altri tre. Presentazioni: Nessuna | ByCategory | Sostituito | NotDisplayed.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`manage_project_units`

Ottieni o imposta le unità di progetto (lunghezza, area, volume, angolo, ecc.). Azioni: ottieni, imposta, list_valid_units.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`manage_worksets`

Creare, rinominare, eliminare o impostare il workset attivo (solo modelli condivisi). Per LISTare i workset utilizzare get_worksets.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

`modify_schedule`

Modificare i campi della pianificazione, l'ordinamento, i filtri, i totali generali, la formattazione delle colonne o rinominare la pianificazione. actions supportato: `add_field`, `remove_field`, `set_sorting`, `clear_sorting`, `set_filter`, `clear_filter`, `set_totals`, `set_field_format`, `rinomina`.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`purge_unused`

Elimina dal progetto famiglie e tipi non utilizzati (`families/types`), materiali non utilizzati e, se richiesto, modelli di vista e filtri di vista senza riferimenti.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`read_pricelist`

Leggere un listino prezzi in formato CSV, Excel (`.xlsx`), XML o PriMus XPWE. Normalizza ogni voce in codice, breve descrizione e criterio di misurazione, ma non deduce mai una mappatura Revit né scrive dati.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`record_computo_voce`

Registra un'assegnazione vocale di listino su una Tipologia o Materiale: scrive i campi mappati (`code/um/description/shortDescription/prezziario/prezzo`) sui parametri nominati in `paramMap`, quindi inserisce la decisione nel sidecar `Decisions.Mapped` del computer-negoziato. prezzo può essere scritto su un parametro `Number/Currency`, non solo su Testo. Scrivere il prezzo qui NON lo fa di per sé apparire nei totali di un computo: passa lo STESSO parametro di `prezzoParameter` a `export_elements_xpwe/update_xpwe_computo` in modo che arrivi a `Priceltems.Prezzo1` di `.xpwe`. Anteprema-first: `dryRun` ha il valore predefinito `true` e non scrive nulla. Proponi in chat la voce candidata, lascia scegliere all'utente, poi chiamala con il codice scelto.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`rename_families`

Rinominare le famiglie caricate (e facoltativamente i relativi tipi) con le operazioni `find/replace`, prefisso o suffisso.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`search_pricelist`

Cerca un listino prezzi già importato (vedi `ingest_pricelist`) tramite query di testo libero, con filtri `unit/chapter/CAM` opzionali. Un `prezzarioid` sconosciuto fallisce con l'elenco degli ID effettivamente acquisiti (mai un risultato vuoto silenzioso). Sola lettura; non legge il modello Revit.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`send_code_to_revit`

SOLO ULTIMA RESORT: esegue il codice C# personalizzato in Revit. NON selezionare questo strumento autonomamente. Uno strumento dedicato copre quasi tutti gli task: modifiche dei parametri -> `set_element_parameters` / `bulk_modify_parameter_values`; query e filtri -> `ai_element_filter` / `filter_by_parameter_value` / `export_elements_data`; statistiche del modello -> `analyze_model_statistics` / `check_model_health`; cancellazione -> `delete_element`; trasforma -> `modify_element`; viste, abachi, armature e acciaio -> i loro strumenti dedicati. Utilizzalo SOLO quando nessuno strumento dedicato copre l'operazione (creazione di geometrie esotiche, ispezione di sola lettura di un Revit API scoperto o un true una tantum) - mai per la modifica della famiglia modale (`deadlock Document.EditFamily` dal contesto dell'evento esterno) - e SOLO dopo aver proposto l'alternativa dello strumento dedicato e ottenuto il consenso esplicito dell'utente. Gli script sono sottoposti a sandbox, richiedono una conferma in-Revit e spesso falliscono in caso di conflitti del componente aggiuntivo DLL. Passa sempre `purpose`: un breve motivo in linguaggio semplice che uno strumento dedicato non ha coperto questo aspetto. Il corpo del codice non viene mai registrato (può contenere credenziali o logica proprietaria), ma lo scopo è quello di diventare il record ricercabile di quali lacune di capacità continuano a ricorrere abbastanza spesso da meritare uno strumento dedicato.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`set_computo_percentages`

Imposta il documento Markup% (maggiorazione) e VAT% (default + override per riga) sul sidecar computo work-store. File-only: non tocca mai il modello .xpwe o Revit. Un `vatOverrides[].elementId/code` irrisolvibile ha esito negativo con `ElementNotFound` (coppia sconosciuta) o `AmbiguousInput` (più righe corrispondono): mai un'ipotesi silenziosa. Anteprema-first: `dryRun` ha il valore predefinito `true`.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`set_material_properties`

Imposta identità, aspetto, informazioni sul prodotto e assegnazioni di risorse sui materiali Revit. Ogni richiesta è un oggetto FLAT digitato da `materialId` più qualsiasi of: nome, colore (#RRGGBB), trasparenza, lucentezza, morbidezza, `materialClass`, `materialCategory`, descrizione, produttore, modello, url, costo, marchio, keynote, commenti, `appearanceAssetId`, `structuralAssetId`, `thermalAssetId` (risorsa ids: 0/-1 cancella).

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`sync_computo_model_state`

Proietta lo stato per elemento del computo (Non computato/Computato/Da aggiornare) sullo model: legge il sidecar del negozio di lavoro, rimisura con la sua ricetta registrata, confronta con il precedente .xpwe e timbra di conseguenza due parametri di istanza nascosti (`Pulse_StatoComputo/Pulse_NotaComputo`). I calcoli di origine pianificata vengono rifiutati (nessuna identità di elemento su cui proiettarsi). Anteprema-first: `dryRun` ha il valore predefinito `true` e non scrive nulla.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

`tag_rooms`

Tagga le stanze nella vista attiva. Funziona solo sulla vista attiva: attiva prima la vista corretta.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs`

tag_walls

Etichettare i muri nei loro punti medi nella vista attiva. Funziona solo sulla vista attiva. Etichetta tutti i muri per impostazione predefinita o un sottoinsieme tramite wallIds.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

update_xpwe_computo

Aggiorna un computo XPWE ESISTENTE per un nuovo phase: rimisura il modello, accoppia le righe di misurazione in base al loro ID elemento e segnala aggiunto/aggiornato (con il delta)/invariato/rimosso. NON DECIDE NULLA: ogni categoria di modifica necessita di un'azione esplicita prima che il file venga riscritto, altrimenti la chiamata viene rifiutata con l'elenco delle decisioni pendenti. Anteprema-first: dryRun ha il valore predefinito true.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

verify_computo_preflight

Check deterministico pre-consegna del computo: completezza (elementi in-scope non mappati), coerenza interna (voci senza codice, qty<=0, pinned disallineate), UM incoerente e doppio-conteggio 1:N; riconciliazione vs abaco se passi scheduleId. Read-only, non decide la consegna.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

verify_xpwe_computo

Verificare un computo XPWE rispetto a un programma Revit che il progettista ha già convalidato: un secondo testimone indipendente (rimisurare il modello confermerebbe solo che lo strumento concorda con se stesso). Riporta le voci corrispondenti, le discrepanze con delta e percentuale e ciò che esiste solo su un lato. Nomi quando i totali concordano mentre gli elementi no (una modifica compensativa si nasconde in un totale). Non scrive nulla.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

wipe_empty_tags

Trova e rimuovi tag vuoti o orfani

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

workflow_clash_review

Rileva le collisioni tra due categorie e crea una vista riquadrata in sezione 3D per la revisione visiva.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

workflow_data_roundtrip

Esporta i parametri in Excel per la modifica esterna, quindi reimporta una volta salvato il file.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

workflow_model_audit

Esegui un flusso di lavoro di controllo del modello completo.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

workflow_room_documentation

Genera automaticamente viste di dettaglio (e facoltativamente sezioni) per ogni stanza su un livello.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

workflow_sheet_set

Crea automaticamente un set di fogli con cartigli da una definizione list: [{number, name, viewIds?}].

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ProjectTools.cs

Armature

convert_rebar_system_to_rebars

Convertire un sistema di armatura su area o su percorso in armature autonome (distruttive). Fornire systemId. Restituisce gli ID delle armature autonome risultanti.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

create_area_reinforcement

Crea un sistema di armatura d'area su un elemento ospitante: parete, solaio o fondazione (wall/floor/foundation). majorDirection è un oggetto JSON {x,y,z}; il parametro facoltativo curves è un array JSON di {type:line|arc, start{x,y,z}, end{x,y,z}, mid?{x,y,z}} in millimetri, per definire esplicitamente il contorno. memberCount/memberIds descrivono le barre dopo la rigenerazione; se è presente memberCountNote, il conteggio era ancora 0: rileggere i dati con get_area_reinforcement_data.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

create_fabric_area

Creare un sistema di aree del tessuto su un host (wall/floor/foundation). majorDirection è JSON {x,y,z}; le curve opzionali sono una serie JSON di {type:line|arc, start{x,y,z}, end{x,y,z}, mid?{x,y,z}} in mm che formano un anello chiuso.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

create_fabric_sheet

Crea un singolo foglio di tessuto in un host. Fornire hostId e fabricSheetTypeId o fabricSheetTypeName; opzionale bendProfile è una matrice JSON di {type:line|arc, start{x,y,z}, end{x,y,z}, mid?{x,y,z}} in mm che forma un anello chiuso (fa piegare la lamiera).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

create_free_form_rebar

Creare un'armatura a forma libera non vincolata da anelli di curva (mm) in un host. loops è un array di loop JSON, ciascuno un array JSON di specifiche della curva {type, start, end, mid?}.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

create_path_reinforcement

Creare un sistema di rinforzo del percorso su un host. curves è un array JSON di {type:line|arc, start{x,y,z}, end{x,y,z}, mid?{x,y,z}} in mm (obbligatorio). Ribaltamento opzionale, pathTypeId, startHookId, endHookId. memberCount/memberIds riflettono le barre dopo la rigenerazione; se è presente memberCountNote il conteggio era ancora 0 - rileggere con get_path_reinforcement_data.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

create_rebar_bending_detail

Creare un dettaglio di piegatura dell'armatura per un'armatura in una vista (Revit 2024+). Fornire rebarId e viewId (una vista drafting/detail); opzionale bendingDetailTypeId, posizione JSON {x,y,z} in mm, rotationDegrees. Restituisce un errore di versione su Revit 2023.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

create_rebar_coupler

Creare un accoppiatore dell'armatura che collega due estremità della barra o chiuderne una. Fornire couplerTypeId o couplerTypeName (un tipo di famiglia della categoria accoppiatore), end1 (JSON {rebarId,end}) e end2 opzionale (JSON {rebarId,end}). 'end' è 0 o 1.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

create_rebar_from_curves

Creare un'armatura da curve complanari esplicite (mm) in un host. curve è un array JSON di {type:line|arc, start{x,y,z}, end{x,y,z}, mid?{x,y,z}}; normale è JSON {x,y,z}. Opzionale hooks/layout.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

create_rebar_from_shape

Creare un'armatura guidata dalla forma in un host da una forma dell'armatura. origin/xVec/yVec sono JSON {x,y,z} in mm. Disposizione opzionale JSON.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

get_area_reinforcement_data

Leggere un'armatura su area system: direzione principale (vettore mm), tipo id/name, ID host, ID dell'armatura dell'elemento, ID della curva di confine, conteggio degli elementi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

get_fabric_area_data

Leggere un'area del tessuto system: digitare id/name, ID host, ID fogli, conteggio fogli, direzione principale (vettore mm).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

get_fabric_rounding

Leggere il documento sulle regole di arrotondamento della lunghezza del tessuto (applicare bandierina, arrotondamento della lunghezza in mm segment/total e metodo Nearest|Up|Down).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_fabric_sheet_data`

Leggere un tessuto sheet: digitare id/name, isBent, fabricNumber, tagliare lunghezza e larghezza totale (mm).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_fabric_wire_data`

Leggere gli elementi metallici di un foglio di tessuto in una direzione. Fornire fabricSheetId e direzione (major|minor); opzionale maxWires (predefinito 200). Restituisce diametro per filo (mm), distanza (mm), lunghezza filo (mm).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_path_reinforcement_data`

Leggere un'armatura del percorso system: tipo id/name, ID host, ID dell'armatura dell'elemento, ID dell'elemento curva, offset aggiuntivo (mm), orientamento della barra primaria.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_rebar_api_capabilities`

Riporta quale rinforzo con controllo della versione presenta i supporti Revit in esecuzione.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_rebar_bending_detail_data`

Leggere un dettaglio di piegatura dell'armatura (Revit 2024+): ID dell'armatura host, ID vista del proprietario, posizione (mm) e rotazione (gradi). Fornire bendingDetailId. Restituisce un errore di versione su Revit 2023.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_rebar_constraint_candidates`

Elenca i vincoli candidati per una maniglia dell'armatura. Fornire rebarId e handleIndex (da manage_rebar_constraints action=list_handles). Sola lettura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_rebar_constraints`

Elenca le maniglie vincolate di un'armatura e se i relativi vincoli possono essere modificati.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_rebar_coupler_data`

Leggere un'armatura coupler: couplerMark, quantità, tipo id/name e ciascun descrittore di armatura collegato {rebarId, end}. Fornire couplerId.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_rebar_element_data`

Leggere il nucleo data di una singola barra d'armatura: tipo di barra, host, forma, regola di layout, numero di barre, lunghezza totale (mm), volume.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_rebar_geometry`

Restituisce le curve della linea centrale (mm) di una barra d'armatura in corrispondenza di un indice di posizione della barra (predefinito 0). Opzionalmente sopprimi il raggio hooks/bend.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_rebar_host_data`

Rinforzo del report ospitato da un element: validità e lo rebar/area/path/fabric in esso contenuto, più copertura comune.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_rebar_numbering`

Leggi la numerazione delle armature. Con rebarId restituisce il contrassegno di pianificazione di quella barra; senza di esso restituisce il contrassegno di ogni abaco dell'armatura più il conteggio dei contrassegni vuoti (un proxy per la numerazione degli spazi vuoti). L'elenco dell'intero documento è terminato da maxResults (truncated/returnedCount contrassegna un taglio) e summaryOnly restituisce solo i conteggi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_rebar_rounding`

Leggi le regole di arrotondamento della lunghezza delle armature. Senza rebarId restituisce il documento predefinito; con rebarId restituisce l'arrotondamento effettivo di quella barra. Il metodo è Nearest|Up|Down. Non è previsto alcun arrotondamento del volume in Revit API.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_rebar_splice_candidates`

Ripartire le geometrie di giunzione candidate per un'armatura in base alle regole (Revit 2025+, sola lettura). Fornire rebarId e facoltativamente spliceTypeId; la posizione deve essere 'Middle' - il regolamento API documenta End1/End2 come non valido. spliceTypeId deve essere un ElementType di OST_RebarSpliceType, ottenuto da list_rebar_splice_types - un accoppiatore FamilySymbol viene rifiutato. Restituisce un errore di versione sui target più vecchi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_rebar_splice_data`

Leggere i dati di giunzione dell'armatura (Revit 2025+): per ciascuna estremità della barra, lunghezza sovrapposizione (mm), sfalsamento (mm), posizione di giunzione, armatura collegata id/end, più la catena di giunzione. Fornire rebarId. Restituisce un errore di versione sui target più vecchi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_rebar_varying_data`

Lettura dello stato dell'armatura a lunghezza variabile (Revit 2025+, sola lettura): canHaveVaryingLengthBars, varyingEnabled e lunghezza della linea centrale per posizione (mm). Fornire rebarId, opzionale includeBarLengths (predefinito true). Restituisce un errore di versione sui target più vecchi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`get_reinforcement_settings`

Leggere le impostazioni di rinforzo a livello di documento.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`include_exclude_rebar_bars`

Mostra o nascondi una singola barra di un set di armature in una vista. Fornire rebarId, viewId, barPositionIndex, nascosto (true=nascondi).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`list_rebar_bar_types`

Elenca tutti i tipi di barre d'armatura (ID, nome, modello e diametro nominale in mm).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`list_rebar_cover_types`

Elenca tutti i tipi di copriferro dell'armatura (id, nome, distanza del copriferro in mm).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`list_rebar_fabric_types`

Elencare i tipi di rinforzo del tessuto (tipi di fogli di tessuto e tipi di aree di tessuto).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`list_rebar_hook_types`

Elenca tutti i tipi di ganci dell'armatura (ID, nome, angolo del gancio in gradi).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`list_rebar_shapes`

Elenca tutte le forme dell'armatura (ID, nome).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`list_rebar_splice_types`

Elenca i tipi di giunzione dell'armatura (Revit 2025+; restituisce un errore di versione sulle destinazioni più vecchie).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`manage_fabric_rounding`

Imposta le regole di arrotondamento della lunghezza del tessuto del documento. Campi: applyRules (bool), lengthRoundingMm (doppio), lengthRoundingMethod (Nearest|Up|Down). volumeRounding non è supportato da API ed è segnalato negli avvisi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

manage_rebar_constraints

Vincoli dell'armatura Inspect/edit. Fornire rebarId e action: list_handles | list_candidates (con handleIndex) | set_preferred (con handleIndex, candidateIndex) | remove_preferred (con handleIndex) | ricalcolare (2025+). Le azioni list_* sono di sola lettura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

manage_rebar_numbering

Gestire la numerazione delle armature. action=set_number scrive il contrassegno di pianificazione di una singola battuta (richiede rebarId + newNumber). action=renumber|remove_gaps non vengono esposti da Revit API e restituiscono un risultato strutturato 'unsupported'. Una destinazione di sola lettura evidenzia il problema negli avvisi anziché in errori.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

manage_rebar_rounding

Impostare le regole di arrotondamento della lunghezza dell'armatura. Senza rebarId modifica il documento predefinito; con rebarId modifica quella barra. Campi: applyRules (bool), lengthRoundingMm (doppio), lengthRoundingMethod (Nearest|Up|Down). volumeRounding non è supportato da API ed è segnalato negli avvisi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

modify_rebar_bending_detail

Modificare un dettaglio di piegatura dell'armatura (Revit 2024+). Fornire bendingDetailId e qualsiasi posizione JSON {x,y,z} in mm, rotationDegrees. Restituisce un errore di versione su Revit 2023.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

move_rebar_in_set

Spostare una singola barra all'interno di un'armatura impostata da un vettore di traslazione (mm). Fornire rebarId, barPositionIndex, traduzione JSON {x,y,z}. Supera reset:true per cancellare una mossa precedente.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

place_fabric_sheet

Posiziona un foglio di tessuto esistente in un host. Fornire fabricSheetId e hostId; la trasformazione opzionale è JSON {translation:{x,y,z}} in mm (identità predefinita).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

propagate_rebar

Propaga l'armatura a host simili. NOTA: Revit API non espone alcun metodo di propagazione dell'armatura su nessuna versione supportata; questo restituisce un risultato strutturato 'unsupported'. Fornire l'array rebarId, opzionale targetHostIds JSON.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

remove_fabric_reinforcement_system

Rimuovere un sistema di rinforzo dell'area del tessuto (distruttivo). Fornire fabricAreaId.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`remove_rebar_splice`

Rimuovere una giunzione dell'armatura all'estremità della barra (Revit 2025+). Fornire rebarId e barEnd opzionale (0 o 1; predefinito 0). Per impostazione predefinita viene visualizzata un'anteprima che segnala anche se esiste una giunzione in quel end: passare dryRun=false per applicare. Restituisce un errore di versione sui target più vecchi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`remove_rebar_system`

Rimuovere un'area o un sistema di rinforzo del percorso (distruttivo). Fornire systemId.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`set_area_reinforcement_layers`

Attivare o disattivare uno strato di un sistema di rinforzo su area. Fornire areaReinforcementId, livello (top_major|top_minor|bottom_major|bottom_minor) e attivo.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`set_fabric_sheet_bend_profile`

Imposta il profilo di piega di un foglio di tessuto piegato. Fornire fabricSheetId e bendProfile (un array JSON di {type:line|arc, start{x,y,z}, end{x,y,z}, mid?{x,y,z}} in mm che forma un anello chiuso). Valido solo quando il foglio è piegato.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`set_path_reinforcement_options`

Imposta le opzioni su un sistema di rinforzo del percorso. Fornire pathReinforcementId e uno qualsiasi tra additionalOffsetMm, primaryBarOrientation (TopOrExterior|BottomOrInterior|NearSide|FarSide), alternatingBarOrientation. Le chiavi non supportate vengono segnalate negli avvisi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`set_rebar_coupler_visibility`

Impostare un accoppiatore non oscurato (solido) o oscurato in una vista. Fornire couplerId, viewId, non oscurato (bool).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`set_rebar_hooks`

Impostare il tipo di gancio alle estremità dell'armatura. Fornire rebarId e startHookId and/or endHookId (passare 0 per liberare il gancio di un'estremità). Funziona su tutte le versioni Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

`set_rebar_host`

Riassegnare un'armatura a un nuovo host. Fornire rebarId e newHostId (deve essere un host di barre d'armatura valido).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

set_rebar_layout

Impostare il layout di distribuzione di un'armatura basata sulla forma. il layout è JSON {rule, number?, arrayLengthMm?, spacingMm?, barsOnNormalSide?, includeFirstBar?, includeLastBar?}. Numeri richiesti per rule: fixed_number -> numero 1..1002 + arrayLengthMm>0; maximum_spacing / minimum_clear_spacing -> spacingMm>0 + arrayLengthMm>0; number_with_spacing -> numero 1..1002 + spacingMm>0; singolo -> nessuno. I valori non validi restituiscono InvalidInput prima della transazione.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

set_rebar_shape

Modificare la forma di un'armatura guidata dalla forma. Fornire rebarId e shapeld o shapeName.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

set_rebar_terminations

Impostare le terminazioni finali dell'armatura (orientation/rotation). Solo Revit 2027+; restituisce un errore di versione sui target più vecchi. Fornire rebarId, fine (0|1), orientamento, rotationDegrees.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

set_rebar_varying

Enable/disable un set di barre d'armatura di lunghezza variabile (il comando 'Set di barre d'armatura variabile', Revit 2025+). Fornire rebarId e abilitato (bool): quando true i vincoli del set determinano le lunghezze per battuta. Il set deve essere basato sulla forma ed essere idoneo (CanHaveVaryingLengthBars). Restituisce un errore di versione sui target più vecchi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

set_rebar_visibility

Imposta la presentazione della vista dell'armatura. Fornisci rebarId, viewId e non oscurato (mostralo davanti all'host).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

set_reinforcement_settings

Configurare le impostazioni di rinforzo a livello di documento. Fornire uno qualsiasi tra hostStructuralRebar, rebarShapeDefinesHooks, rebarShapeDefinesEndTreatments (bool). Alcune opzioni si applicano solo quando il documento non ha rinforzi; le modifiche bloccate vengono segnalate negli avvisi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

splice_rebar

Giuntare un'armatura in base alle regole (Revit 2025+). Fornire rebarId e facoltativamente spliceTypeId; la posizione deve essere 'Middle': le regole base API documentano End1/End2 come non valide e appartengono a sovraccarichi manuali basati su linea che questo strumento non espone. spliceTypeId deve essere un ElementType di OST_RebarSpliceType, ottenuto da list_rebar_splice_types - un accoppiatore FamilySymbol viene rifiutato. Sostituisce la barra con diversi segmenti, quindi il valore predefinito è preview: passa dryRun=false per applicare. Restituisce gli ID dell'armatura risultanti o un errore di versione sulle destinazioni più vecchie.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

split_rebar

Dividere un set di armature basato sulla forma in due set in una determinata posizione della barra. Fornire rebarId e splitAtPosition (1..count-1). Restituisce gli ID dell'armatura originale e quella nuova.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

transfer_rebar_annotations

Trasferire le annotazioni sulle armature tra le viste ricreando MultiReferenceAnnotations sulle armature visibili nella vista di origine. Fornire sourceViewId, targetViewId. I problemi per elemento sono segnalati negli avvisi[].

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

unify_rebars

Unifica le barre indipendenti compatibili in una sola (Revit 2025+). Fornire rebarIds (array JSON di ID >=2); le barre vengono unificate a coppie in un'unica armatura. Tutti tranne il PRIMO ID vengono consumati e i relativi ID diventano non validi, quindi per impostazione predefinita viene utilizzato preview: passa dryRun=false da applicare. Restituisce un errore di versione sui target più vecchi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\RebarTools.cs

Strutture in acciaio

add_steel_fabrication_info

Aggiungere informazioni sulla fabbricazione dell'acciaio agli elementi Revit in modo che partecipino ai dettagli dell'acciaio (SteelElementProperties). Fornire elementIds come array JSON, e.g. [123,456]. Supporta dryRun. Restituisce gli ID che hanno ricevuto informazioni sulla fabbricazione più eventuali ID saltati.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

add_steel_instance_void_cut

Aggiungere un'istanza di taglio void in modo che un'istanza di famiglia void tagli un altro elemento (InstanceVoidCutUtils). Fornire voidInstanceId (l'istanza del vuoto di taglio) e targetElementId (l'elemento da tagliare). Supporta dryRun. Nota: questo è un taglio geometrico Revit generico, non specifico per l'acciaio.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

add_steel_solid_cut

Aggiungi un taglio solido in modo che un elemento ne tagli un altro (SolidSolidCutUtils). Fornire cutElementId (la taglierina) e targetElementId (l'elemento da tagliare). Opzionale splitFaces (predefinito false). Supporta dryRun. Nota: questo è un taglio geometrico Revit generico, non specifico per l'acciaio.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

analyze_structural_steel_model

Acciaio strutturale a livello di documento summary: conteggi di gestori di connessione, tipi di connessione, tipi di gestori di connessione, tipi di approvazione ed elementi strutturali framing/column che trasportano SteelElementProperties. summaryOnly conta solo i resi; altrimenti array di campioni limitati tramite maxResults.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

check_steel_cut_eligibility

Controlla se un elemento può tagliarne un altro tramite un taglio solido and/or un'istanza di taglio vuoto, senza mutare. Fornire cutElementId (la taglierina) e targetElementId (l'elemento da tagliare). Restituisce solidCutEligible (+ solidCutFailureReason quando false) e instanceVoidCutEligible.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

create_default_steel_connection_handler_type

Creare il valore StructuralConnectionHandlerType predefinito per il documento (CreateDefaultStructuralConnectionHandlerType). Restituisce il nuovo ID del tipo. Supporta dryRun.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

create_generic_steel_connection

Crea una connessione strutturale generica tra due o più elementi (funziona senza un fornitore di connessione installato: la linea di base sicura). Fornire elementIds (array JSON di ID elemento >=2); opzionale connectionName. Supporta dryRun.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

create_steel_connection

Creare una connessione strutturale tipizzata tra due o più elementi da un tipo di gestore connessione (connectionHandlerTypeId o connectionHandlerTypeName). Richiede una connessione installata provider/type. Fornire elementIds (array JSON di ID >=2). Supporta dryRun. inputPoints sono accettati ma non ancora cablati (Revit non espone alcun costruttore ConnectionInputPoint pubblico).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

create_steel_connection_handler_type

Crea un StructuralConnectionHandlerType. Fornire il nome; opzionale familyName (default vuoto); guid opzionale (quando omesso viene generato un nuovo GUID). Supporta dryRun. Restituisce il nuovo tipo id e la sua connessione GUID.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

create_steel_structural_connection_type

Crea un StructuralConnectionType legato a un simbolo di famiglia. Fornire familySymbolId (un simbolo di famiglia di connessione valido); applyTo = BeamsAndBraces | ColumnTop | ColumnBase | Connessione (Connessione predefinita); nome facoltativo. Supporta dryRun.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

delete_steel_connection

Elimina un gestore di connessione strutturale tramite connectionId. Distruttivo: supporta l'anteprima di dryRun. Gli elementi collegati stessi non vengono cancellati.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`get_instance_void_cut_relationships`

Leggere le relazioni di taglio istanza-void di un elemento (InstanceVoidCutUtils): `cuttingVoidInstances` (istanze void che tagliano questo elemento) e `elementsBeingCut` (elementi che questo elemento taglia quando è esso stesso un'istanza void di taglio). I conteggi vengono sempre restituiti; matrici a meno che `summaryOnly`, troncato a `maxResults` (predefinito 100). Generic Revit geometry, not steel-specific.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`get_solid_cut_relationships`

Leggere le relazioni di taglio solido-solido di un elemento (SolidSolidCutUtils): `cuttingSolids` (solidi che tagliano questo elemento) e `solidsBeingCut` (solidi che questo elemento taglia). I conteggi vengono sempre restituiti; matrici a meno che `summaryOnly`, troncato a `maxResults` (predefinito 100). Geometria generica Revit, non specifica per l'acciaio.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`get_steel_connection_applicability`

Segnala suggerimenti di applicabilità di `StructuralConnectionType`. Revit non espone alcun predicato pubblico "questo tipo si applica a questi elementi", quindi restituisce l'ID del simbolo della famiglia `ApplyTo` + del tipo e, per qualsiasi `elementIds` fornito, le relative categorie, chiaramente etichettate come consultive.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`get_steel_connection_data`

Leggere un gestore di connessione strutturale tramite id: tipo id/name, ID degli elementi connessi, origine, flag `custom/detailed`, ID del tipo di approvazione, stato di controllo del codice, flag di `override-type-params`.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`get_steel_connection_input_points`

Leggere i punti di input di una connessione strutturale handler: ID di ciascun punto (GUID) e posizione (x,y,z in mm). Fornire `connectionId`.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`get_steel_connection_settings`

Leggere il documento `StructuralConnectionSettings` (attualmente espone il flag `IncludeWarningControls`).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`get_steel_connection_type_data`

Leggere un tipo di connessione strutturale tramite ID. Restituisce i dati `StructuralConnectionType` (identificativo simbolo famiglia, `applyTo`) o `StructuralConnectionHandlerType` (connessione GUID, flag `generic/custom/detailed`) a seconda del tipo di elemento.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`get_steel_connection_validation`

Segnala avvisi di convalida per un gestore di connessione strutturale. Revit API non espone alcun produttore pubblico di `ConnectionValidationInfo` per un gestore inserito, quindi restituisce `validationAvailable=false` con lo stato di controllo del codice del gestore e una nota. Utilizzare lo strumento generale `get_warnings` per gli errori a livello di documento.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs`

`get_steel_cut_data`

Leggere le relazioni di taglio per un element: tagli solido-solido (taglio di solidi + solidi tagliati tramite `SolidSolidCutUtils`) e tagli di istanze-vuote (taglio di istanze vuote + elementi tagliati tramite `InstanceVoidCutUtils`).

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs`

`get_steel_element_fabrication_properties`

Leggere le proprietà di fabbricazione dell'acciaio di un element: se ha `SteelElementProperties` (`hasFabricationProperties`) e il suo ID univoco di fabbricazione (stringa GUID). Fornire `elementId`.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs`

`get_steel_element_properties`

Leggere le proprietà di fabbricazione dell'acciaio di un element: se trasporta `SteelElementProperties` e il suo ID univoco di fabbricazione (GUID). L'enumerazione dell'id esterno e del collegamento materiale non viene esposta da Revit SDK. Utilizzare `summaryOnly` solo per le bandiere.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs`

`get_steel_element_warnings`

Riporta gli avvisi di fabbricazione in acciaio per un elemento (o tutti gli elementi se `elementId` viene omissso). Revit SDK non espone alcun avviso specifico per l'acciaio API; questo restituisce `count 0` con una nota. Utilizzare lo strumento generale `get_warnings` per gli errori a livello di documento. `summaryOnly` conta solo i resi.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs`

`get_steel_external_id_map`

Riporta la mappa ID esterno della fabbricazione di acciaio per un elemento. Revit SDK non espone l'enumerazione degli ID esterni per elemento; questo restituisce l'ID univoco di fabbricazione (se presente) e conta 0 con una nota.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs`

`get_steel_fabrication_unique_id`

Leggere l'ID univoco della fabbricazione dell'acciaio (GUID) di un elemento dal relativo `SteelElementProperties`. Fornire `elementId`. Restituisce una nota quando l'elemento non ha proprietà di fabbricazione in acciaio.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs`

`get_steel_material_links`

Segnala i collegamenti ai materiali di fabbricazione in acciaio per un elemento. Revit SDK non espone l'enumerazione del materiale collegato su `SteelElementProperties`; questo restituisce `count 0` con una nota.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs`

`get_steel_reference_by_fabrication_id`

Risolvere l'elemento Revit a cui fa riferimento una fabbricazione in acciaio GUID. Fornire fabricationGuid (un GUID). Restituisce found=true con il riferimento elementId o found=false quando nessun elemento corrisponde.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`get_structural_connection_provider_data`

Segnalare metadata/capabilities di un fornitore di connessioni strutturali tramite id/key.

StructuralConnectionsProviderData è un buffer opaco riempito dal provider senza lettore pubblico, quindi segnala available=false con una nota.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`get_structural_connection_provider_registry`

Segnala i fornitori di connessioni strutturali registrati (Autodesk Steel Connections, IDEA StatiCa, ecc.). Revit API non espone alcuna query pubblica sul registro del provider, quindi segnala availability=false con una nota anziché creare un elenco.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`get_structural_connection_validation_info`

Dettagli di convalida del report per una connessione strutturale posizionata (connectionId). Nessun API pubblico produce un ConnectionValidationInfo popolato per un gestore esistente, quindi restituisce il CodeCheckingStatus del gestore e una nota.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`get_structural_steel_api_capabilities`

Segnala quale acciaio strutturale presenta la versione Revit in esecuzione supports: SteelElementProperties, connessioni strutturali, utenze tagliate, mutazione di connessione personalizzata API (rimossa in R27) e se è rilevabile qualsiasi fornitore di connessione strutturale.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`list_steel_approval_types`

Elenco StructuralConnectionApprovalType definitions: id, nome. Utilizzare maxResults (predefinito 100) e summaryOnly per la navigazione con i conteggi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`list_steel_connection_handler_types`

Elenco StructuralConnectionHandlerType definitions: id, nome, connessione GUID, flag generic/custom/detailed. Utilizzare maxResults (predefinito 100) e summaryOnly per la navigazione con i conteggi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`list_steel_connection_handlers`

Elenca i gestori delle connessioni strutturali nel documento con id, id/nome del tipo (id/name), numero di elementi collegati e indicatori custom/detailed. Usare maxResults (predefinito 100) e summaryOnly per iniziare dai conteggi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`list_steel_connection_providers`

Elenca i fornitori di connessioni strutturali installati. Il pubblico Revit API non espone alcun registro dei provider interrogabile; questo restituisce il conteggio 0 con una nota esplicativa.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`list_steel_connection_types`

Elenca le definizioni `StructuralConnectionType` nel documento: id, nome, id del simbolo di famiglia e `applyTo`. Usare `maxResults` (predefinito 100) e `summaryOnly` per iniziare dai conteggi.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`manage_custom_steel_connection_type`

Modificare una connessione strutturale personalizzata (gestore). `action` = `add_references` | `remove_references` | `add_elements` | `remove_subelements`. NOTA: la mutazione della connessione personalizzata di Revit necessita di `References/Subelements` selezionati in modo interattivo che non possono essere creati da JSON, quindi questo strumento convalida gli input e restituisce un errore strutturato anziché indovinare. Le API `add/remove legacy` sono state rimosse in Revit 2027.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`manage_steel_approval_type`

Amministrare le definizioni `StructuralConnectionApprovalType`. `action` = `crea` (richiede nome) | `lista`. Revit API non espone `rename/delete` per i tipi di approvazione, quindi tali azioni restituiscono un errore strutturato.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`modify_steel_connection_inputs`

Aggiungere o rimuovere elementi connessi su un gestore di connessione strutturale. `action` = `add_element_ids` | `remove_element_ids` (fornire `elementIds[]`). `add_references/remove_references` non sono supportati tramite questo strumento (i riferimenti Revit non possono essere creati dagli ID JSON). Restituisce i conteggi `accepted/skipped`.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`remove_steel_instance_void_cut`

Rimuovere un taglio vuoto di istanza tra un'istanza di famiglia vuota e un altro elemento (`InstanceVoidCutUtils`). Fornire `voidInstanceId` e `targetElementId`. Geometria Revit generica, non specifica per l'acciaio.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`remove_steel_solid_cut`

Rimuovere un taglio solido tra due elementi (`SolidSolidCutUtils`). Fornire `cutElementId` e `targetElementId`. Geometria Revit generica, non specifica per l'acciaio.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

`set_steel_connection_approval`

Imposta il tipo di approvazione di un gestore di connessione strutturale. Fornire `connectionId` e `approvalTypeId` o `approvalTypeName` (verificato rispetto alle definizioni `StructuralConnectionApprovalType` del documento).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

set_steel_connection_default_order

Reimpostare un gestore connessione strutturale sull'ordine degli elementi predefinito (SetDefaultElementOrder). Fornire connectionId.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

set_steel_connection_status

Imposta lo stato di controllo del codice di un gestore di connessione strutturale. status = NotCalculated | OkChecked | CheckingFailed.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

set_steel_connection_type

Modificare il tipo di connessione strutturale. Revit non espone alcun setter di tipo sul posto, quindi ricrea connection: legge gli elementi connessi, elimina il vecchio gestore e ne crea uno nuovo con connectionHandlerTypeId|connectionHandlerTypeName. Richiede una connessione installata provider/type. Supporta dryRun. I punti di input esistenti non vengono conservati.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

set_steel_connection_type_family_symbol

Riassocia un StructuralConnectionType a un simbolo di famiglia diverso. Fornire connectionTypeId e familySymbolId. Il nuovo simbolo viene convalidato rispetto al tipo ApplyTo esistente. Supporta dryRun.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

set_steel_fabrication_unique_id

Imposta l'ID univoco della fabbricazione dell'acciaio (GUID) del SteelElementProperties di un elemento. Fornire elementId e uniqueId (un GUID). L'elemento deve già avere proprietà di fabbricazione in acciaio (eseguire prima add_steel_fabrication_info).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

set_steel_solid_cut_face_splitting

Impostare se le facce del solido di taglio vengono divise in un taglio solido esistente (SolidSolidCutUtils.SplitFacesOfCuttingSolid). Fornire cutElementId, targetElementId e dividere (bool, obbligatorio). Richiede un taglio solido esistente tra i due elementi. Supporta dryRun. Geometria Revit generica, non specifica per l'acciaio.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\StructuralSteelTools.cs

Flussi di lavoro

manage_user_flows

Creare, ottenere, elencare o eliminare flussi Pulse definiti dall'utente (memorizzati per utente). creare necessita di un oggetto 'flow' JSON {id,title,description,triggers[],steps[{name,gate,tools[]}]}; ottenere ed eliminare la necessità 'id'.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\UserFlowTools.cs

Viste

`activate_view`

Aprire e attivare una vista o un foglio tramite `viewId` o esattamente `viewName`. Necessario prima degli strumenti che agiscono SOLO sulla vista attiva (`tag_rooms`, `tag_walls`, `color_elements`, `create_dimensions`). UI only: il modello non viene modificato.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs`

`align_viewports`

Allinea le finestre tra i fogli. 'placement' corrisponde ai centri dei riquadri; 'model' corrisponde all'angolo minimo del contorno del riquadro in modo che le visualizzazioni in scala uguale della stessa regione siano allineate.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs`

`apply_view_template`

Elenca, applica o rimuovi modelli di vista dalle viste. `action=list|apply|remove`. Un `templateId` che non è effettivamente un modello di vista viene RIFIUTATO con quello che è realmente (measured: un normale FloorPlan e un tipo di porta entrambi utilizzati per restituire successo), quindi ottieni gli ID da `action=list` anziché da un elenco di viste. LA COMPATIBILITÀ È PER VISTA TARGET: `View.IsValidViewTemplate` di Revit viene controllato per ogni vista prima dell'assegnazione, perché un modello valido per una vista può non essere valido per un'altra e l'assegnazione grezza genera un errore: le viste che non possono accettarlo vengono segnalate in ``skipped`` con il motivo invece di essere conteggiate silenziosamente. Entrambi applicano e rimuovono il report `skippedCount`, quindi un'esecuzione parziale è distinguibile da una completa.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs`

`batch_create_sheets`

Crea più fogli con cartigli e posizionamento della vista opzionale. fogli è un JSON array: `[{number, name, titleBlockName?, viewIds?}]`.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs`

`batch_modify_view_range`

Modifica gli offset dell'intervallo di visualizzazione (superiore, piano di taglio, inferiore, profondità di visualizzazione) per più viste. Gli offset sono in mm.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs`

`close_views`

Chiudi vista aperta windows: gli `viewIds` elencati, oppure tutti tranne quello attivo. Non chiude mai l'ultima finestra. UI only: le visualizzazioni rimangono nel modello e `activate_view` le riapre.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs`

`create_placeholder_sheets`

Crea, elenca, converti o elimina fogli segnaposto. `action=create|list|convert|delete`.

Ambito: | `sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs`

`create_preset_schedule`

Creare un programma da un modello predefinito (e.g. RoomFinish, DoorHardware, WallQuantities, WindowSchedule).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`create_schedule`

Crea una nuova visualizzazione della pianificazione in Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`create_sheet`

Crea un nuovo foglio nel progetto Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`create_view`

Crea una nuova vista in Revit: pianta del pavimento, pianta del soffitto, sezione, prospetto, disegno o vista 3D. Può andare in timeout se eseguito in parallelo con altre scritture: è meglio eseguirlo da solo.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`create_view_filter`

Creare, applicare o elencare filtri di visualizzazione basati su parametri (action=create|apply|list). Un filtro trasporta una regola (parameterName/filterRule/filterValue) o più tramite l'array di regole combinato con AND/OR. Per apply: superare filterId+viewId, opzionalmente overrideR/G/B.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`create_views_from_rooms`

Crea viste di dettaglio, sezione o prospetto da locali con uno schema di denominazione.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`delete_schedule`

Elimina un programma tramite ID o nome

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`duplicate_schedule`

Duplica una pianificazione con un nuovo nome. duplicateOption e filterBySheet opzionali (le pianificazioni non supportano le viste dipendenti: as_dependent viene rifiutato con il suggerimento di utilizzare invece filterBySheet).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`duplicate_sheet_with_content`

Duplica un foglio includendo annotazioni ed elementi di dettaglio

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`duplicate_sheet_with_views`

Duplica un foglio N volte con le opzioni di duplicazione della vista configurabili.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`duplicate_view`

Duplica una vista esistente in Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`get_current_view_elements`

Elenca gli elementi visibili nella vista attualmente attiva.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`get_current_view_info`

Ottieni informazioni sulla vista attualmente attiva in Revit.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`get_schedule_data`

Esporta i dati della pianificazione come JSON da una visualizzazione della pianificazione esistente. Restituisce sempre un `config` block: filtri (con fieldName, filterType e valore digitato), sortGroupFields (ordine + raggruppamento header/footer/blankLine), flag grandTotal e nome/intestazione per colonna. I valori doppi del filtro sono unità interne Revit (piedi), non convertite.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`get_view_graphics`

Leggi cos'è una vista overriding: flag nascosto per categoria e superficie con sostituzione grafica COMPLETA (colore, spessore e motivo della linea projection/cut; ID motivo, colore e visibilità della superficie e del taglio foreground/background; mezzitoni; trasparenza; livello di dettaglio), con IsCategoryOverridable/CanCategoryBeHidden in modo da sapere cosa è impostabile. Restituisce inoltre le sostituzioni per elemento per elementIds, i filtri applicati IN ORDINE (l'ordine decide quale filtro vince) con le relative sostituzioni, visibilità e stato abilitato, visibilità del workset, impostazioni di collegamento RVT, opzioni nascoste a livello di classe, modello di visualizzazione e viewTemplateId + graphicsOverridesAllowed: una vista guidata da un modello non può essere sovrascritta direttamente, che è la causa più comune di una scrittura grafica non riuscita. Approfondisci per default: vengono elencate solo le categorie nascoste o sovrascritte. Sola lettura.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

`list_open_views`

Elenca le visualizzazioni del documento attivo che hanno una finestra aperta e quale è attiva. Ingresso per close_views e zoom_view.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

list_views

Elenca viste, fogli e modelli di vista con gli ID utilizzati dagli altri strumenti view/sheet, inclusi gli ID VIEWPORT necessari a align_viewports e gli ID delle istanze di abaco su ciascun foglio, che nessun altro strumento produce. Per view: ID, nome, viewType, scala, detailLevel, disciplina, isTemplate, MODELLO DI VISTA ASSEGNATO (viewTemplateId + viewTemplateName) e areGraphicsOverridesAllowed, più dove è posizionato. LEGGERE viewTemplateId PRIMA DI DIAGNOSTICARE UNA SCRITTURA GRAFICA NON RIUSCITA: API afferma "I parametri controllati dal modello non possono essere modificati direttamente in questa vista", quindi un modello non null è il solito motivo per cui una modifica scale/detail-level/override sembra non fare nulla. scala, detailLevel e disciplina sono null quando la vista non ha tale proprietà (la API genera tabelle, legende e fogli) - null significa non applicabile, mai 0. posizionamenti è un ARRAY perché una legenda può trovarsi su molti fogli e un abaco raggiunge un foglio attraverso un ScheduleSheetInstance anziché un Viewport, quindi ogni posizionamento dichiara placedVia. Snella per impostazione predefinita; matchingCount riporta la popolazione prima di maxResults quindi una pagina troncata non viene mai letta come l'intero modello.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

manage_unplaced_views

Elenca o elimina le viste che non sono posizionate su nessun foglio

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

manage_view_templates

Elenca, duplica, elimina, rinomina o crea modelli di visualizzazione.

action=list|duplicate|delete|rename|create_from_view. create_from_view crea un nuovo modello da una vista esistente (passaggio viewId). Le azioni di scrittura vengono impostate automaticamente su un'anteprima (l'eliminazione segnala quante visualizzazioni perderebbero il modello): passare dryRun=false per applicare.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

override_graphics

Sostituisci la grafica degli elementi in una vista (colori, trasparenza, mezzitoni, spessore della linea).

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

place_viewport

Posiziona una vista su una tavola come finestra, con rotazione e tipo di vista opzionali.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

rename_views

Rinominare in batch le viste utilizzando le operazioni find/replace, prefisso o suffisso. Applicalo con filterName and/or viewTypes: SENZA di essi l'operazione prende di mira OGNI vista non modello nel documento. L'anteprima riporta candidateCount (viste raggiunte dal filtro) e skippedUnchangedCount (già denominata correttamente), quindi "nessuna visualizzazione corrispondente" e "niente da modificare" rimangono distinguibili.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

zoom_view

Ingrandire una finestra di visualizzazione aperta (default: la visualizzazione attiva): adatta, fattore, rettangolo o sheet_size. Restituisce gli angoli incorniciati in mm. Solo UI.

Ambito: | sources\revit-official\src\Pulse.Server\Tools\ViewTools.cs

Fonte verificata

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

main / 2c0407f58c5266eb9466a4bdd2553a05f7da30b1

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