

PULSE / Solibri

Guida e comandi

Solibri Office 26.6

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

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

Branch: main | Commit: bc7fd4c5041832a48c62c1b0526fa08255ba02e5

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-solibri.html>

Installazione e collegamento

È richiesto Solibri Office con l'API supportata. Il plugin Java e il server MCP devono appartenere allo stesso pacchetto; apri il progetto e, se necessario, la vista Pulse Bridge. Collega il server nel client AI seguendo il pacchetto. Installer commerciale e licenze sono in preparazione: non sostituire il pacchetto approvato con script di sviluppo.

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.

Download e seriale automatico non ancora collegati al form. Contattaci per questo prodotto.

Richieste di esempio

- > Esegui il checking e mostrami i risultati critici della regola selezionata.
- > Isola i componenti coinvolti e salva un'immagine della vista.
- > Leggi la tabella ITO ed esporta le issue in BCF.

FUNZIONALITÀ

Regole e checking

Consulta ruleset, regole e parametri; modifica i parametri supportati ed esegui il checking del progetto.

Risultati e triage

Filtra componenti non conformi per gravità e decisione; aggiorna il triage degli esiti.

Componenti e selezioni

Cerca componenti e proprietà IFC; leggi o aggiorna la Selection Basket.

Review visiva

Gestisci camera, visibilità e colori; isola gli elementi e salva snapshot della vista 3D.

ITO e quantità

Consulta definizioni e tabelle di Information Takeoff, con ricalcolo dove richiesto.

BCF e librerie

Esporta issue in BCF; consulta e utilizza le funzioni di libreria esposte dal plugin.

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.

BCF

`solibri.bcf.export`

Esporta le issue di Solibri in un file BCF (.bcfzip) sul disco e restituisce il percorso. version: V1_0|V2_0|V2_1|V3_0 (default V2_1). scope: ALL|SELECTED|MARKED (default ALL).

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\BcfTools.cs

Verifica del modello

`solibri.checking.chosen_get`

Elenca le regole/ruleset attualmente spuntati nella vista Checking di Solibri - cosa verrebbe eseguito lanciando `solibri.checking.run` adesso. Diverso da `solibri.rulesets.list` (quello mostra il contenuto dei ruleset, questo lo stato di selezione nell'interfaccia).

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\CheckingTools.cs

`solibri.checking.chosen_rows`

Righe attualmente evidenziate nella tabella dei risultati del checking, risolte negli stessi oggetti Result/componente/issue usati dagli altri tool - per agire su "quello che l'utente ha selezionato in Solibri adesso" invece che solo su ciò che seleziona l'assistente.

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\CheckingTools.cs

`solibri.checking.chosen_set`

Imposta quali regole/ruleset sono spuntati nella vista Checking (sostituisce la selezione corrente) - usalo per "esegui solo queste regole" prima di `solibri.checking.run`. I ref che non si risolvono vengono ignorati e riportati in 'notFound'.

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\CheckingTools.cs

`solibri.checking.decision_set`

Imposta il triage (ACCEPTED/REJECTED/UNDEFINED) di un risultato di checking, dato il suo uniqueKey (da `solibri.results.get`) - es. per accettare una deviazione già discussa invece di lasciarla riapparire ad ogni check.

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\CheckingTools.cs

`solibri.checking.failed`

Componenti che falliscono una data regola, da `Checking.getFailedComponents` - filtro diretto per triage mirato (es. "solo i CRITICAL ancora UNDEFINED di questa regola") invece di scorrere tutti i risultati con `solibri.results.get`.

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\CheckingTools.cs

`solibri.checking.run`

Esegue il checking del progetto Solibri aperto e restituisce un summary (numero di risultati). `checkSelected=true` limita ai componenti selezionati.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\CheckingTools.cs`

`solibri.results.get`

Restituisce i risultati dell'ultimo checking: name, description, severity, uniqueKey, ruleName, componentGuids (i componenti coinvolti - agganciabili a `solibri.components.info`, `solibri.view.color`, `solibri.issues.create` ecc.), decision (ACCEPTED/REJECTED/UNDEFINED) e `slideId` (l'issue collegata, se esiste). Filtrabile per componente o per regola.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\CheckingTools.cs`

Interferenze

`solibri.clash.query`

Rileva interferenze/clash tra due gruppi di componenti nel modello aperto, calcolate al volo (non richiede regole caricate - funziona sempre). Broad-phase su bounding box poi intersezione precisa. Restituisce coppie target/ostacolo con dimensioni, volume e `contactPoint` (centroide preciso della mesh di compenetrazione, utile per posizionare una camera/viewpoint esattamente sul punto di interferenza - null se non calcolabile). 'matched' indica quante interferenze esistono in totale, 'truncated' se la lista è stata tagliata da limit.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\ClashTools.cs`

Componenti

`solibri.components.info`

Restituisce nome, tipo e property set di uno o più componenti dati i loro GUID IFC.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\ComponentTools.cs`

`solibri.components.query`

Cerca o trova componenti (muri, pilastri, travi, solai...) nel modello e restituisce i loro GUID (necessari per selezione, isolamento, colorazione e info). Filtri opzionali: type (es. WALL, DOOR, COLUMN - vedi `solibri.components.types`) e `nameContains`. 'matched' indica quanti ne esistono in totale, 'truncated' se la lista è stata tagliata da limit.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\ComponentTools.cs`

`solibri.components.types`

Elenca i tipi di componente presenti nel modello aperto con il conteggio di ciascuno (es. WALL: 120, DOOR: 34). Usalo per sapere cosa cercare prima di una query.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\ComponentTools.cs`

Geometria

`solibri.classifications.classify`

Esegue un sistema di classificazione (da `solibri.classifications.list`) su un componente e restituisce i codici/etichette corrispondenti - es. "a quale codice Uniclass appartiene questo muro".

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\GeometryTools.cs`

`solibri.classifications.list`

Elenca i sistemi di classificazione disponibili nel modello aperto (es. Uniclass, OmniClass) con nome, descrizione, edizione e fonte.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\GeometryTools.cs`

`solibri.components.contains`

Verifica se un punto 3D e' realmente dentro il volume di un componente (test sulla mesh reale, non sul bounding box) - es. per verificare se una coordinata rilevata (laser scanner, sensore) cade davvero dentro quell'elemento.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\GeometryTools.cs`

`solibri.components.distance`

Distanza reale (non bounding-box) tra due componenti: valori scalari rapidi (`distance/horizontalDistance/verticalDistance`) piu' il segmento esatto dei due punti piu' vicini sulle superfici reali. Per verifiche di franco/clearance anche quando i componenti NON si compenetrano - complementare a `solibri.clash.query`, che copre solo i casi di sovrapposizione.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\GeometryTools.cs`

`solibri.components.footprint`

Impronta 2D di un componente: area, perimetro, numero di vertici, centroide, area al netto dei fori. Utile per domande di superficie/copertura senza dover esportare la geometria 3D completa.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\GeometryTools.cs`

`solibri.components.line`

Componenti attraversati dalla retta infinita passante per due punti, estesa in ENTRAMBE le direzioni (a differenza di `solibri.components.ray`, che va in una sola direzione da un'origine) - utile per l'asse di allineamento di un grid o di una tubazione.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\GeometryTools.cs`

`solibri.components.mesh`

Geometria triangolata reale di un componente: area, volume, bounding box e conteggio triangoli/vertici - indipendente da qualunque quantita' IFC dichiarata. Utile per un controllo incrociato quando il software di origine non ha esportato le quantita', o per verificarle. Non restituisce la mesh grezza (nessun vertice/triangolo elencato).

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\GeometryTools.cs`

`solibri.components.ray`

Componenti attraversati da un raggio con origine e direzione arbitrarie (una sola direzione, dall'origine in poi) - per verifiche di franco verticale/orizzontale, es. "cosa incontro salendo da questo punto fino al soffitto".

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\GeometryTools.cs`

`solibri.components.region`

Componenti che intersecano un volume 3D arbitrario (min/max), indipendente da un componente esistente - a differenza di `solibri.clash.query`, che parte sempre dai limiti di un componente target. Per una zona di cantiere, un buffer attorno a un punto, o qualunque volume non modellato come Space IFC.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\GeometryTools.cs`

`solibri.components.related`

Grafo delle relazioni IFC di un componente - es. VOIDS/FILLS per apertura<->muro ospitante, SERVES per terminale MEP<->impianto che lo alimenta, DEFINES_BY_TYPE per istanza<->tipo, CONTAINS per gerarchie di composizione. Un'unica relazione alla volta.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\GeometryTools.cs`

Issue

`solibri.issues.comments_add`

Aggiunge un commento testuale a un'issue esistente - utile per registrare una verifica o una decisione presa (es. "confermata in sopralluogo del 27/08").

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\IssuesTools.cs`

`solibri.issues.comments_list`

Elenca i commenti testuali su un'issue (da `solibri.issues.list`).

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\IssuesTools.cs`

`solibri.issues.create`

Crea una nuova issue (Slide) nel progetto aperto, collegata opzionalmente a dei componenti e/o a un risultato di checking. Viene aggiunta a una Presentation (per default "Pulse Issues", riusata se esiste già) - e' quello che la rende visibile nel pannello Issues di Solibri, non solo un oggetto in memoria.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\IssuesTools.cs`

`solibri.issues.delete`

Elimina un'issue (Slide) rimuovendola da ogni Presentation a cui appartiene, dato il suo `uniqueId` (da `solibri.issues.list`).

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\IssuesTools.cs`

`solibri.issues.list`

Elenca le issue (Slide) native di Solibri nel progetto aperto - le stesse del pannello Issues. scope: all (tutte), selected (selezionate nell'interfaccia), marked (marcate manualmente dall'utente, tipicamente per un export mirato).

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\IssuesTools.cs`

`solibri.issues.update`

Aggiorna un'issue esistente dato il suo `uniqueId` (da `solibri.issues.list`). Cambia solo i campi passati.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\IssuesTools.cs`

`solibri.issues.viewpoint_add`

Cattura la vista 3D CORRENTE di Solibri (stato camera + snapshot immagine) e la allega a un'issue come prova visiva - usalo subito dopo aver inquadrato l'elemento in questione con `solibri.view.look_at` o `solibri.view.camera_set`.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\IssuesTools.cs`

Estrazione delle informazioni (ITO)

`solibri.ito.definitions`

Elenca le definizioni di Information Takeoff (computi/quantità) presenti nel progetto Solibri.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\ItoTools.cs`

`solibri.ito.table`

Restituisce la tabella ITO corrente (colonne, righe, riga totale). Con `calculate=true` ricalcola prima tutti i takeoff (operazione potenzialmente lunga).

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\ItoTools.cs`

Librerie

`solibri.library.list`

Elenca i ruleset DISPONIBILI nella libreria di Solibri (file .cset su disco), cioè quelli caricabili dal Ruleset Manager - non necessariamente già aperti nel progetto. Filtro opzionale per nome/cartella. NB: per vedere le REGOLE dentro un ruleset e la loro configurazione bisogna prima caricarlo in Solibri (azione GUI), poi usare `solibri.rules.search / solibri.rules.describe`.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\LibraryTools.cs`

Regole

`solibri.rules.get`

Restituisce i parametri di una regola dato il suo ref (dal campo 'ref' di `solibri.rulesets.list`): id, tipo, valore.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\RuleTools.cs`

Catalogo delle regole

`solibri.rules.describe`

Descrive una regola caricata: cosa controlla (descrizione ufficiale) e come è CONFIGURATA ORA (valori dei parametri: tolleranze, target, eccezioni). Funziona anche sulle regole built-in di Solibri. NB: i valori si possono leggere ma non modificare via API - la modifica va fatta nella GUI di Solibri.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\RuleCatalogTools.cs`

`solibri.rules.search`

Cerca fra TUTTE le regole caricate nel progetto quelle che corrispondono a una richiesta in linguaggio naturale (cerca in nome E descrizione ufficiale della regola). Usalo quando l'utente chiede 'che controllo posso fare per X': restituisce le regole candidate con percorso nel ruleset e descrizione.

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\RuleCatalogTools.cs

Parametri delle regole

`solibri.rules.set_parameter`

Imposta il valore di un parametro di regola. Con `confirm=false` restituisce solo un preview (valore attuale vs proposto) e NON scrive nulla. Con `confirm=true` applica `setValue` e restituisce `oldValue/newValue` (post-verifica). Tipi supportati: Boolean/Int/Double/String/Enumeration.

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\RuleParameterTools.cs

Set di regole

`solibri.rulesets.export_json`

Restituisce la serializzazione JSON nativa di un ruleset (`Ruleset.toJson`) più la sua descrizione. È l'unica superficie pubblica che potrebbe contenere i VALORI configurati dei parametri delle regole built-in.

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\RulesetTools.cs

`solibri.rulesets.get`

Restituisce il sotto-albero di un ruleset dato il suo `uniqueId`.

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\RulesetTools.cs

`solibri.rulesets.list`

Elenca i ruleset del progetto Solibri aperto, con l'albero delle regole (`uniqueId`, `name`, `isRule`, `children`).

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\RulesetTools.cs

Selezione

`solibri.selection.get`

Restituisce i GUID IFC dei componenti attualmente selezionati - la selezione corrente (Selection Basket) di Solibri. Usalo per sapere cosa è selezionato adesso.

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\SelectionTools.cs

`solibri.selection.set`

Imposta la Selection Basket ai componenti con i GUID IFC dati (sostituisce la selezione corrente).

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\SelectionTools.cs

Impostazioni

`solibri.bcf.servers`

Elenca i server BCF-API configurati (es. BIMcollab, Trimble Connect) per la sincronizzazione live delle issue - distinto da `solibri.bcf.export` che produce un file statico. Solo lettura: l'autenticazione richiede un login interattivo nel browser.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\SettingsTools.cs`

`solibri.layouts.list`

Elenca i layout di tab salvati nell'interfaccia Solibri (es. "3D + Info") con nome e numero di riquadri.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\SettingsTools.cs`

`solibri.layouts.switch`

Cambia il layout di tab attivo nell'interfaccia Solibri (da `solibri.layouts.list`).

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\SettingsTools.cs`

`solibri.settings.format`

Formatta un numero esattamente come lo vedrebbe l'utente nell'interfaccia Solibri (unita' di misura e locale configurati) - usalo per presentare lunghezze/aree/volumi/percentuali/angoli restituiti da altri tool invece di mostrare il numero grezzo.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\SettingsTools.cs`

Struttura spaziale

`solibri.components.door`

Dettagli tipizzati di una porta (componente DOOR): larghezza e altezza complessive, tipo di apertura (es. SINGLE_SWING_LEFT, SLIDING_TO_RIGHT) - dati non disponibili tramite `solibri.components.info` generico.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\SpatialTools.cs`

`solibri.components.quantities`

Quantita' IFC standard di un componente (area, volume, lunghezza, larghezza, altezza, spessore, perimetro, diametro ecc. - 23 tipi possibili), ricalcolate da Solibri stesso SENZA dover prima configurare una tabella ITO - risposta immediata a "quant'e' lo spessore di questo muro". Restituisce solo i tipi che il componente effettivamente ha.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\SpatialTools.cs`

`solibri.components.site`

Dettagli tipizzati del componente SITE: elevazione e coordinate geografiche (gradi/primi/secondi) del punto di riferimento IFC del modello - utile per verificare che modelli federati di discipline diverse condividano lo stesso riferimento geografico.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\SpatialTools.cs`

`solibri.components.window`

Dettagli tipizzati di una finestra (componente WINDOW): tipo di partizionamento (schema dei riquadri) - dati non disponibili tramite `solibri.components.info` generico.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\SpatialTools.cs`

`solibri.spaces.summary`

Riepilogo di una stanza (componente SPACE): GUID di porte/finestre/aperture collegate, area totale per tipo di elemento di contorno (pareti, finestre, porte, colonne, travi, pavimenti, soffitti), e `spaceBoundaries` - il dettaglio per-confine (area, tipo PHYSICAL/VIRTUAL, componente reale collegato) per individuare confini VIRTUAL non modellati che falsano i calcoli. Dati non disponibili tramite `solibri.components.info` generico.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\SpatialTools.cs`

`solibri.storeys.components`

Elenca i componenti presenti su un piano/livello dato (GUID da `solibri.storeys.list`) - es. "cosa c'è al piano 3". Filtro opzionale per tipo (vedi `solibri.components.types`).

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\SpatialTools.cs`

`solibri.storeys.list`

Elenca tutti i piani/livelli (BUILDING_STOREY) del modello aperto, con nome, quota ed elevazione, e quanti componenti ciascuno contiene.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\SpatialTools.cs`

Sistema

`solibri.project.scores`

Storico del punteggio di copertura/qualità del progetto (percentuale componenti passati, percentuale regole verificate), calcolato e mantenuto da Solibri stesso - per un andamento nel tempo invece di un solo valore istantaneo.

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\SystemTools.cs`

`solibri.system.info`

Stato di Solibri: se è occupato (busy, es. sta ancora eseguendo un checking - utile da controllare prima del comando successivo), l'utente collegato, il percorso e stato di salvataggio del progetto (NO_FILE/UNSAVED/SAVED/MODIFIED - controllalo prima di suggerire azioni rischiose), l'header IFC del modello (schema, autore, software di origine) e i modelli federati che lo compongono (nome, UUID, software sorgente).

Ambito: | `sources\solibri\server\Pulse.Solibri.Server\Tools\SystemTools.cs`

Viste

`solibri.view.camera.get`

Restituisce lo stato della camera 3D di Solibri: location, direction e upVector (coordinate del modello).

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\ViewTools.cs

`solibri.view.camera.set`

Imposta la camera 3D di Solibri. Si possono passare location e/o upVector (se omessi restano invariati); la direction NON e' settabile via API. Restituisce lo stato risultante (post-verifica).

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\ViewTools.cs

`solibri.view.color`

Colora nella vista 3D i componenti con i GUID dati. Colore ARGB 0-255 (default rosso opaco).

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\ViewTools.cs

`solibri.view.hide`

Nasconde nella vista 3D i componenti con i GUID dati.

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\ViewTools.cs

`solibri.view.isolate`

Isola nella vista 3D solo i componenti con i GUID dati: mostra tutto, poi nasconde tutto il resto.

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\ViewTools.cs

`solibri.view.look_at`

Inquadra nella vista 3D i componenti con i GUID dati (zoom-to-fit). Da usare dopo un isolamento: altrimenti la camera resta dov'era e può guardare il vuoto.

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\ViewTools.cs

`solibri.view.show`

Mostra nella vista 3D i componenti con i GUID dati (senza toccare gli altri).

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\ViewTools.cs

`solibri.view.show_all`

Rende visibili tutti i componenti nella vista 3D (annulla isolamenti e nascondimenti).

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\ViewTools.cs

`solibri.view.snapshot`

Cattura la vista 3D corrente in un file PNG sul disco e restituisce il percorso.

Ambito: | sources\solibri\server\Pulse.Solibri.Server\Tools\ViewTools.cs

Fonte verificata

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

main / bc7fd4c5041832a48c62c1b0526fa08255ba02e5

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