ARCHITECTURAL SERVICES DEPARTMENT

GUIDELINE FOR ADOPTION OF OPENBIM IN PROJECT COLLABORATION

Version: 1

xxxxxx 2025

In response to the increasing significance of OpenBIM, the Development Bureau issued a Technical Circular in January 2025, emphasizing the need for Works Departments to prioritize the adoption of an OpenBIM strategy throughout various project stages. While the benefits of adopting OpenBIM and multi-software collaboration are well-understood, there is a lack of detailed guidance and resources for project teams on hands-on implementation.

Building on the insights gained from the pilot project experiences and aligning with the latest government policy, ArchSD engaged Archpolis Limited to develop and publish the Guideline for adoption of OpenBIM in project Collaboration. This guide is designed to serve as a comprehensive standard for the implementation of OpenBIM practices.

The primary aim of this Guide is to establish essential settings and data field values, particularly in terms of IFC data structures and properties. These guidelines ensure consistency across all IFC models exported from native BIM models, facilitating effective model federation and seamless information exchange across different BIM authoring tools.

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Purpose

This document serves as a comprehensive guide to executing the BIM process through openBIM workflow in project collaboration. It is designed as a practical resource for implementing openBIM, providing detailed, step-by-step instructions based on the insights gained from previous pilot projects, specifically tailored to the local industry.

To effectively demonstrate the coordination and collaboration workflow among various native BIM authoring tools, the guide highlights Graphisoft Archicad, Tekla Structures, and Autodesk Revit as exemplary tools for architectural, structural, and building services modeling. By offering clear, actionable solutions to common challenges, this guideline facilitates the adoption of best practices in the openBIM for professionals across diverse fields.

This Guide sets out all necessary settings and data field values, in terms of IFC data structures and properties, that are required to be kept consistent across all IFC models exported from native BIM models. This is to ensure successful model federation and exchange information across different BIM authoring tools.

1.2. How to read this Guide

1. This guide is written with a clear thought process to help you understand OpenBIM in our local context. Follow this structure to get the most out of it:

    a. **Introduction**: sets the stage for local context.

    b. **OpenBIM Basics:** explores the standards and services provided by buildingSMART,

    c. **IFC- Industry Foundation Classes:** digs into the IFC schema, focusing on what's relevant to our daily work. Explain what IFC is designed to do, and not to do

    d. **Exporting IFC**: gives information requirement recommendation (and explain why) based on IFC schema, not the workflow of any particular software, but with software limitation in mind.

    e. **Referencing IFC**: describes how shared IFC models should be used.

    f. **Common Data Environment**: gives out selection criteria beyond functionalities for an effective coordination on CDE what to consider beyond functionalities.

    g. **Quality Assurance**: discusses typical model audit items on individual and federated, native and IFC models, and advanced checking brought about by IFC models.

    h. **Issue Tracking**: introduces standard issue management communication protocol and workflow.

    i. **Model Federation**: is the centre stage for the implementation of the above.

    j. **Common (Mis)understanding**: discusses and provides insights to the common difficulties encountered by local practitioners on open BIM adoption.

k. **Appendices**: illustrate the above by 3 common BIM authoring softwares, plus other open BIM usage examples.

1.3. Terminology

The abbreviations and terminology / glossary should refer the CIC BIM Dictionary (2024).
The following table lists the acronyms / short forms used in this guide and their meanings:

Acronym / Terminology	Full Name
ArchSD	Architectural Services Department of the Government of the Hong Kong SAR
DEVB	Development Bureau of the Government of the Hong Kong SAR
BEP	BIM Execution Plan
AR	Architectural Discipline
LA	Landscape Discipline
ST	Structural Discipline
BS	Building Services Discipline
AC	Air Conditioning Discipline
FS	Fire Services Discipline
WS	Water Services Discipline
EL	Electrical Discipline
SD	Drainage Discipline
LOIN	Level of Information Need, as defined in CIC BIM Standards General, version adopted in the BEP
IFC2x3CV2	IFC2x3 Coordination View 2.0
IFC4RV	IFC4 Reference View
IFC4DT	IFC4 Design Transfer View
Open BIM	A general term for utilising open formats for interoperation among BIM softwares and platforms.
openBIM®	A suite of Standards and Services created by buildingSMART International promoting and implementating interoperation among BIM softwares and platforms.
Pilot Project	The Museum Project in Kowloon Park, a pilot project adopting open BIM carried out by ArchSD since 2023

1.4. Open Standard

1. In BIM and the building industry, an open standard is a publicly available, collaboratively developed specification that defines rules for data structure, exchange, and interoperability, ensuring vendor-neutral collaboration (e.g., IFC for BIM data schemas). An open format is a specific, non-proprietary file type or data structure that implements an open standard for storing and sharing BIM information across software, promoting accessibility and long-term usability (e.g., .ifc files based on IFC).

2. An open standard allows for interoperability and compatibility among different systems and applications.

3. There are numerous open standards / formats for the building industry, each serving specific purposes. Below are some notable examples:

    a. Brick, https://brickschema.org/

    b. CityGML, https://www.ogc.org/standards/citygml

    c. Digital Buildings Project, https://github.com/google/digitalbuildings

    d. Dotbim, https://www.dotbim.net/

    e. dxf, https://en.wikipedia.org/wiki/AutoCAD_DXF

    f. Green Building XML (gbXML), https://www.gbxml.org/

    g. GeoJSON, https://geojson.org/

    h. GeoPackage, https://www.geopackage.org/

    i. GL Transmission Format (glTF), https://www.khronos.org/gltf/

    j. Hypar Elements, https://hypar-io.github.io/Elements/index.html

    k. materialsdb, http://www.materialsdb.org/

    l. Structural Analysis Format (SAF), https://saf.guide/

    m. Speckle Objects Kit, https://speckle.guide/dev/objects.html

    n. STEP, http://www.steptools.com/stds/step/IS_final_p21e3.html

    o. Universal Scene Description (USD), https://graphics.pixar.com/usd/release/index.html

    p. IFC, "Industry Foundation Classes".

4. While standards like gbXML or glTF serve niche purposes, IFC is often favoured for its unmatched versatility and adoption in Building Information Modeling (BIM). Notably, IFC covers the widest range of industry domains—spanning architecture, engineering, construction, and facility management—making it the most comprehensive option available. Its key benefits include:

a. Broad Scope: Covers geometry, properties, spatial relationships, and project data, unlike specialized formats.

b. Interoperability: Works seamlessly across diverse software, reducing compatibility issues.

c. Vendor Neutrality: Not tied to specific vendors, ensuring flexibility and longevity.

d. Global Standard: As an ISO standard, it's widely recognized and trusted internationally.

e. Collaboration: Supports multidisciplinary teams by centralizing data exchange.

5. These advantages, combined with its extensive domain coverage, make IFC the go-to choice for comprehensive, efficient, and future-proof BIM workflows compared to other formats.

6. This Guide uses IFC as the primary format for exchanging information among BIM models across platforms and stakeholders.

1.5. Resources

The readers may find the following resources useful for matters set out in this Guide:

1. IFC in general
   a. https://www.buildingsmart.org/
2. Autodesk Revit
   a. Revit IFC Manual 2.0 (https://autodesk.ifc-manual.com/)
3. For Archicad:
   a. IFC-Based Information Models in openBIM Workflows - Workflow Guide
   b. IFC Reference Guide for Archicad 26
4. For Tekla Structures
   a. Product Guide - IFC (https://support.tekla.com/doc/tekla-structures/2024/int_ifc_general_info)

2. OPENBIM BASICS

2.1. BuildingSMART

buildingSMART is a global community of chapters, members, partners and sponsors led by the parent body, buildingSMART International. The buildingSMART community is committed to creating and developing open digital ways of working for built asset environment. buildingSMART standards help asset owners and the entire supply chain work more efficiently and collaboratively through the entire project and asset lifecycle.

https://www.buildingsmart.org/about/

A big part of buildingSMART's work is the development and promotion of the Industry Foundation Classes (IFC), an open, international standard (ISO 16739-1:2018) for Building Information Modeling (BIM). IFC was originally created by buildingSMART (then known as the International Alliance for Interoperability) in 1994 to provide a vendor-neutral, standardized data format for sharing information across diverse software platforms and disciplines in the building industry. Today, buildingSMART continues to maintain and update IFC for global construction and asset management. By supporting IFC for open BIM, buildingSMART enables interoperability and collaboration between tools and teams.

2.2. buildingSMART's Standards and Services

buildingSMART offers a set of standards and services to improve BIM interoperability across the industry. The following sections are the ones project teams would usually encounter in daily collaboration:

2.2.1. IFC (Industry Foundation Classes) – Open standard for vendor-neutral digital representation of built assets in BIM

1. User Needs: Standardized Digital Descriptions in BIM

   In BIM workflows, users such as architects, engineers, and infrastructure planners require a consistent, interoperable framework for digitally representing built assets—including buildings, civil structures, components, processes, and relationships—to enable seamless data exchange, minimize vendor lock-in, and support full lifecycle management from design through operation without proprietary barriers.

2. IFC as the Solution

   Industry Foundation Classes (IFC) serves as the solution, comprising a set of standardized, digital descriptions for the built asset industry, developed as an open, global, vendor-neutral standard by buildingSMART International and published under a Creative Commons Attribution-NoDerivatives 4.0 license, as well as ISO 16739.

   It provides core definitions with global consensus, while allowing extensions for non-IFC content—such as additional properties, classification references, or material lists—via custom properties or linkages to the buildingSMART Data Dictionary (bSDD), ensuring teams use agreed-upon, consistent terms.

3. What IFC Contains

   As a standard, IFC defines semantics, attributes, and relationships for entities like objects (e.g., walls, ducts), abstract concepts (e.g., performance, costing), processes (e.g., installation), and actors (e.g., owners, contractors).

   The latest version, IFC4.3 ADD2 (approved as ISO 16739-1:2024), contains over 1300 entities and types, approximately 2500 properties organized in over 750 sets, with enhancements for infrastructure domains and alignment definitions.

   Popular previous versions include IFC2x3 (2005), IFC4 (2013), and IFC4 ADD2 TC1 (2018).

   IFC is currently undergoing a major refactoring, developed under the title IFC5, with its core project plan out for Standards Committee review in 2025, aiming for simplified data exchange and next-generation improvements.

4. How Users Interact with IFC

As a format, IFC can be encoded in various electronic formats, with the recommended STEP Physical File Format (SPFF) as .ifc for exchanging data in versions like IFC2x3, IFC4, and IFC4.3; other encodings include XML or JSON.

Users interact by authoring or importing IFC models in BIM software (e.g., Revit, ArchiCAD, Tekla, or open-source tools like BlenderBIM), exporting/importing files for collaboration, validating against schemas or MVDs, and extending with bSDD-integrated terms for custom requirements in projects like building design or infrastructure planning.

2.2.2. bSDD (buildingSMART Data Dictionary) - Free repository of standardized terms and properties for consistent BIM terminology

1. User Needs: Consistent Terms in BIM

In Building Information Modeling (BIM) workflows, users such as architects, engineers, and facility managers require consistent terminology, properties, and classifications across models and projects to prevent miscommunication, data errors, and interoperability issues during design, construction, and maintenance phases.

2. bSDD as the Solution

The buildingSMART Data Dictionary (bSDD) addresses these needs by serving as a free, vendor-neutral online service provided by buildingSMART International, acting as a centralized repository for interconnected data dictionaries that ensure standardized, reliable definitions for the built environment. It promotes openBIM interoperability and data consistency without proprietary restrictions

3. What bSDD Contains

bSDD hosts definitions of terms, including classifications (e.g., objects like "wall" or "duct," materials, and concepts), properties (e.g., "diameter" or "volume" with data types, units, and values), relations (e.g., hierarchies, mappings), translations, and reference entities (e.g., countries, languages, units).

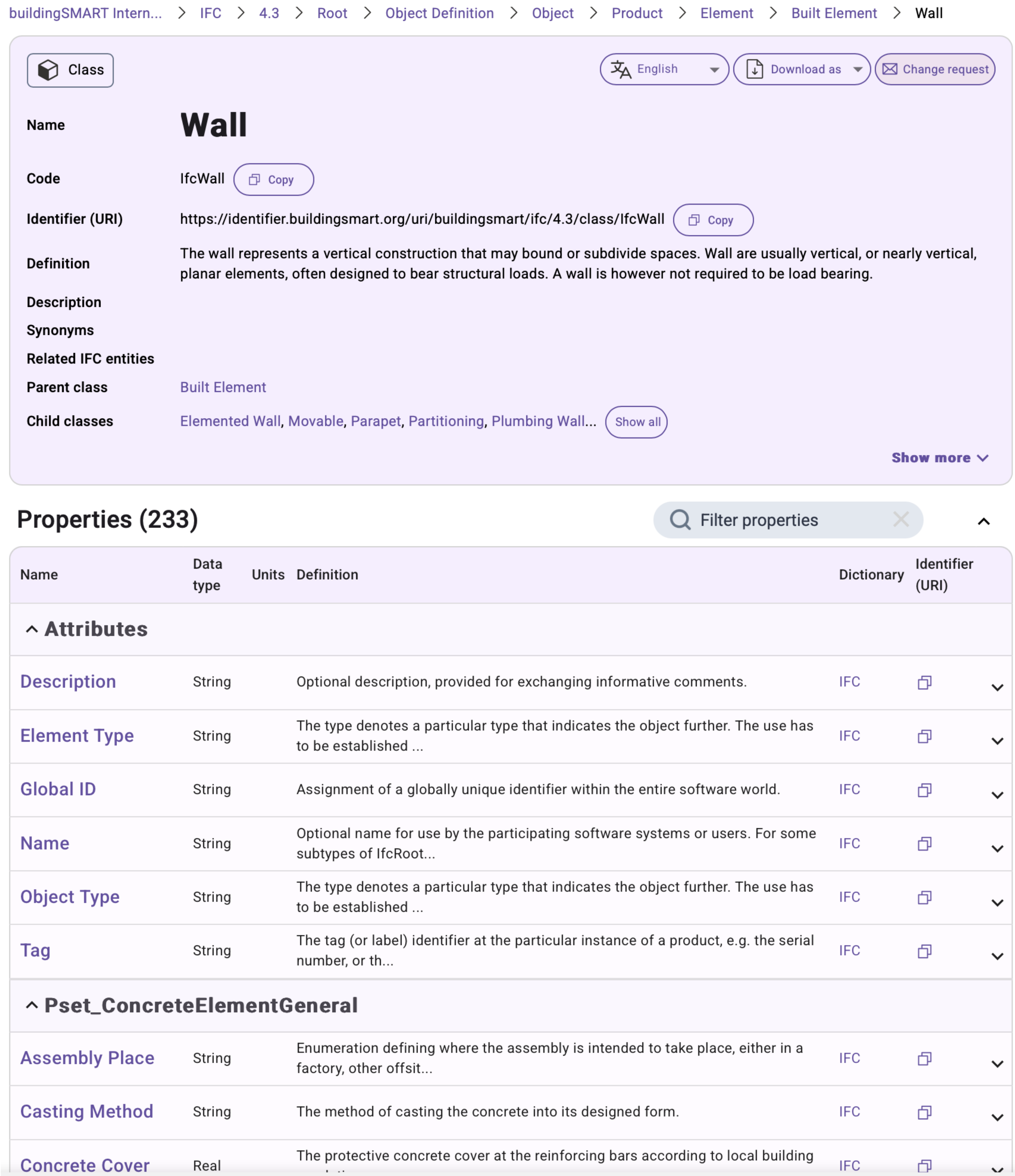
It includes

a. standard terms from international classifications (e.g., IFC), national standards (e.g., Uniclass, DIN), and industry agreements (e.g., ETIM), as well as,

b. user-defined terms from organizations or companies,

all structured per ISO 23386 and ISO 12006-3 for consistency.

As of 2025, content quality is enhanced through verification processes, earning a "verified" badge for high-quality dictionaries



bsDD documentation the definition, attributes and Pset of the IfcWall

4. How Users Interact with bSDD

a. Users search for terms via the bSDD search interface (e.g., querying "wall" or browsing dictionaries like IFC), download data, and integrate it into software via plugins (e.g., Revit, BlenderBIM) or APIs for validation in IDS or IFC models.

b. To upload content, authorized organizations register, prepare JSON or Excel files, and submit via the Manage portal or API, with previews for testing and private dictionaries as a paid feature. This enables sharing custom terms while maintaining version control and status (e.g., Preview, Active, Inactive).

c. Local authorities may create custom bSDD dictionaries to ensure alignment with local BIM standards (e.g., DEVB BIM Harmonisation Guidelines and ArchSD BIM Guides) and to enforce consistent terminology for public projects (such as facility upkeeping and code compliance).

2.2.3. IDS (Information Delivery Specification) – Open standard for defining and validating BIM data requirements in a machine-readable format

1. User Needs: Clear Information Requirements in BIM

In BIM workflows, users need to define and exchange precise information requirements—such as specific properties, classifications, or data formats for elements like walls or ducts—to avoid ambiguities, ensure compliance, and automate validation across models and stakeholders. This prevents errors in data delivery during project phases like handover or certification.

2. IDS as the Solution

The Information Delivery Specification (IDS) is a buildingSMART standard that provides a computer-interpretable format for defining these requirements, making them human-readable yet machine-checkable to promote clarity, interoperability, and automation in openBIM processes. It integrates with standards like IFC and bSDD, reducing manual checks and supporting multi-version IFC compatibility.

For example: All **walls** should have the property **FireRating** in the set **Pset_WallCommon** with a value being one of **0, 30**, **60, 120, 240**.

3. What IDS Contains

An IDS file (in XML format) includes structured specifications: applicability filters (e.g., targeting IFC entities or classifications), requirements (e.g., mandatory properties, attribute values, materials, parts), and optional elements like cardinality rules, regular expressions for values, and references to bSDD for standardized terms. It can define use-case-specific exchanges, such as facility management handover, with facets for entities, attributes, classifications, properties, materials, and parts.

4. How Users Interact with IDS

Users create IDS files using tools like Plannerly, IDS editors, or XML authoring, often starting from templates or bSDD-integrated platforms. They apply IDS to validate IFC models via software plugins (e.g., in BIM tools or validators like buildingSMART's), checking compliance and generating reports. Sharing occurs through project contracts or repositories, enabling automated audits and iterative refinements.

2.2.4. BCF (BIM Collaboration Format) – Open format for exchanging BIM issues, viewpoints, and comments independently of models

1. User Needs: Issue Communication in BIM

In BIM workflows, collaborators such as architects, engineers, and contractors need to identify, discuss, and resolve model-based issues—like clashes, design errors, or change requests—without exchanging entire IFC models, to streamline coordination, reduce data overhead, and track workflows across disciplines and software tools.

2. BCF as the Solution

The BIM Collaboration Format (BCF) is an open standard developed by buildingSMART International for exchanging contextualized issues and topics in BIM projects, enabling model-independent communication via XML/JSON files or RESTful APIs, bypassing proprietary formats and promoting openBIM collaboration. Version 3.0, the latest as of 2025, introduces API enhancements for web-based workflows, including JSON support and improved issue management.

3. What BCF Contains

A BCF file (.bcf or .bcfzip) structures issues as "topics," each in a folder with:

a. markup.bcf (topic GUID, title, description, type, status, priority, labels, assigned_to, creation/modification details, comments, related files/links);

b. viewpoint.bcfv (camera position, direction, clippings, component visibility/selection via IFC GUIDs, colors, 3D annotations);

c. snapshot.png (view images);

d. optional bim_snippet (arbitrary data like clash excerpts);

e. and project/version info at the root.

BCF 3.0 adds features like server-assigned IDs, authorizations for actions, and extended bim_snippets.

4. How Users Interact with BCF

a. For manual workflow, users generate BCF topics in BIM software (e.g., Revit, ArchiCAD, Tekla, Solibri, Navisworks) via plugins or native support, capturing the following, then export BCF files:

(i) Date and involved parties

(ii) issue description with subsequent comments

(iii) viewpoint (camera angle) with section cuts,

(iv) Snapshots,

(v) related IFC element IDs of relevant model, etc.

The receiving users import these BCF files to view issues, zoom to snapshots, select elements, and add comments for roundtripping.

b. For web-based workflows, connect to servers (e.g., BIMcollab, Bimsync) via APIs for real-time syncing and editing. In practice, issues marked in one tool can seamlessly open and locate in another for review.

c. Advantages include smaller file sizes (avoiding full IFC exchanges), faster resolution with visual context, better cross-platform coordination, and audit trails of comments separate from model data, complementing IFC with issue-based dialogue.

2.2.5. Validation Service - Cloud tool for validating IFC files against schemas and MVDs for compliance

Validation

1. User Needs: IFC Compliance in BIM

In BIM workflows, users such as software developers, modelers, and project teams require reliable validation of IFC files to ensure compliance with standards, detect errors in data structure, geometry, or semantics, and maintain interoperability during model exchange, certification, or quality assurance processes. This prevents issues like invalid exports or misaligned data in collaborative environments.

2. Validation Service as the Solution

The buildingSMART Validation Service is a free, cloud-based platform developed by buildingSMART International, in collaboration with software vendors and projects, to automatically check IFC files against official schemas and Model View Definitions (MVDs), providing detailed reports to confirm quality and standards adherence before sharing or certification. It supports multiple IFC versions (e.g., IFC2x3, IFC4, IFC4.3) and integrates with openBIM workflows.

3. What Validation Service Contains

The service performs comprehensive checks, including:

a. syntax validation (file structure and encoding); schema validation (conformance to IFC data models);

b. MVD validation (specific subsets like Coordination View or Reference View);

c. geometry validation (shape representations and spatial accuracy);

d. and optional advanced rules (e.g., property sets, entity constraints).

Results are presented with color-coded icons (e.g., green for pass, red for fail) and detailed error logs, supporting single or batch file processing.

4. How Users Interact with Validation Service

Users access the service via a web interface (validate.buildingsmart.org), log in for enhanced features (e.g., larger file uploads or history), upload IFC files, select validation options (schema, MVD), and process them to receive immediate results with summaries, error details, and downloadable reports. It also offers API integration for automated workflows in software tools, and documentation/guides for troubleshooting

2.2.6. openCDE - Open APIs for seamless data exchange between CDEs and BIM tools without proprietary limits

openCDE

1. User Needs: Interoperable Data Exchange in BIM

In BIM workflows, users including project teams, CDE providers, and software vendors need seamless, automated exchange of data—such as documents, models, and issues—between Common Data Environments (CDEs) and BIM tools to avoid manual transfers, reduce errors, data loss, and time spent on coordination across platforms and stakeholders.

2. openCDE as the Solution

The openCDE (open Common Data Environment) is a buildingSMART International initiative providing a portfolio of open API standards to enable interoperability and open communication between CDE platforms and BIM tools, automating data flows, minimizing manual interventions, and supporting efficient collaboration in the built asset industry without proprietary restrictions. Unlike proprietary CDEs, which often create data silos, lock-ins, and compatibility issues by restricting exchanges to their ecosystems, openCDE promotes vendor-neutral integration, allowing data to flow freely between diverse tools and platforms, reducing costs and enhancing flexibility in openBIM workflows.

3. What openCDE Contains

openCDE includes a family of APIs:

a. Foundation API (common elements like versioning, authentication, and user info);

b. BCF API (for issue management via BIM Collaboration Format); Documents API (for document sharing, upload/download, approved as Final Standard in December 2023);

c. and Dictionary API (linked to bSDD for terminology).

Specifications use OpenAPI for dynamic endpoints, supporting interactive and non-interactive flows, with future expansions for more features.

4. How Users Interact with openCDE

Advanced users implement the APIs in CDE platforms (e.g., Bimsync, Catenda) and BIM tools (e.g., Solibri, ArchiCAD) via GitHub repositories and OpenAPI specs for code generation. Interaction involves authenticating, discovering endpoints, and performing actions like uploading/downloading documents or syncing BCF issues directly within tools, enabling real-time access and automation in projects like asset management at Schiphol Airport. Developers join working groups or test prototypes via buildingSMART contacts.

3. IFC - Industry Foundation Classes

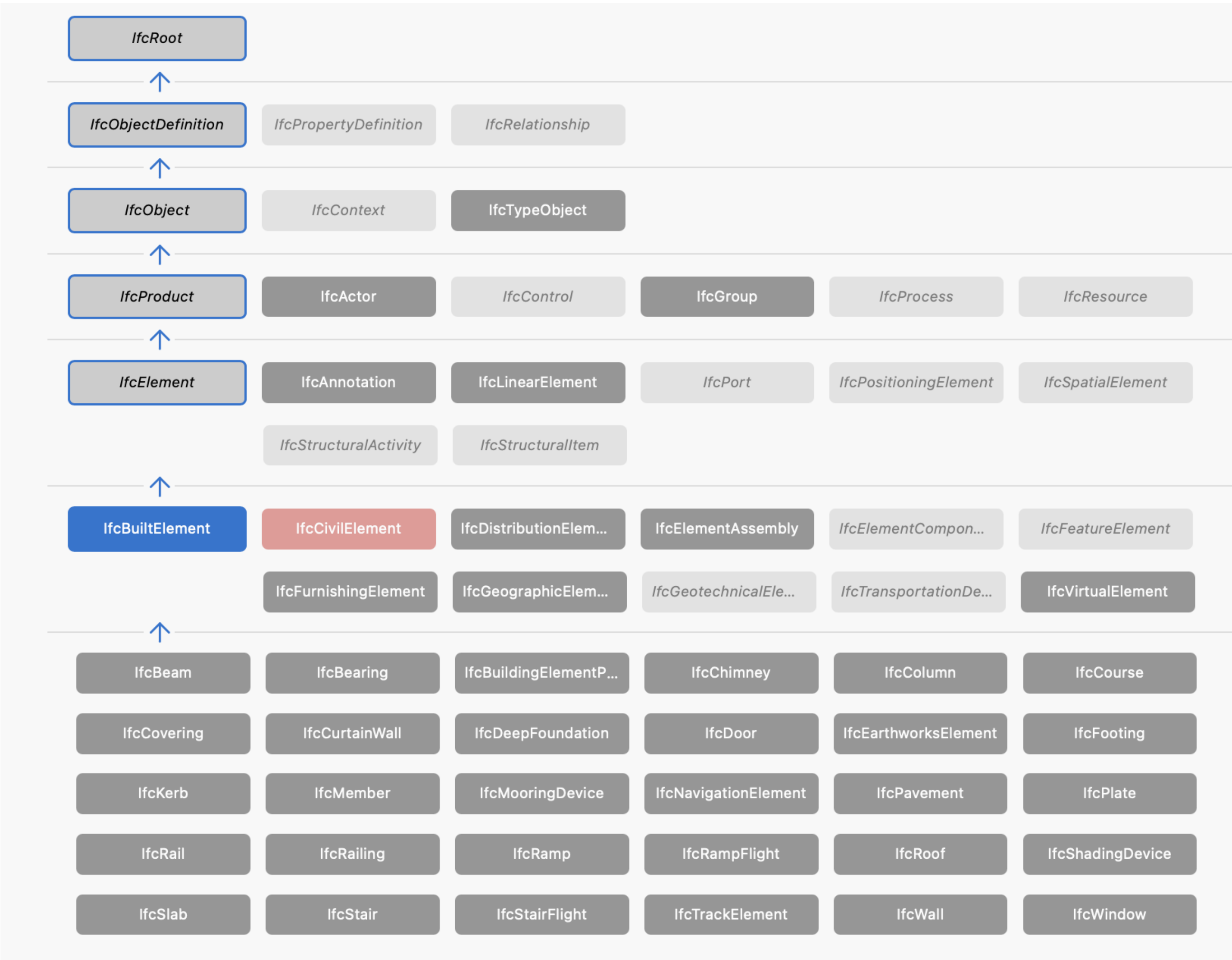
3.1. IFC Entities and Hierarchy

1. IFC entities are the core pieces of an IFC model. Each entity is defined by a name starting with "Ifc" (e.g., IfcWall) and represents something in the project—physical objects like walls, systems like pipes, or concepts like projects.

2. Many entities are abstract, like IfcRoot or IfcElement, meaning they're not used directly but provide a framework for specific, usable entities lower down.

3. Concrete, instantiable entities that represent tangible objects in the model (e.g., IfcWall, IfcDoor) are commonly but unofficially called IFC classes.

4. Entities carry data like geometry, properties, and relationships to build the model. They're organized in a hierarchy, where lower-stream entities inherit properties from upper-stream ones:



The inheritance tree of the IfcBuiltElement Class, extracted from buildingSMART official website

a. At the top is IfcRoot, an abstract entity giving every entity a unique ID, name, and description.

b. Below it, IfcObject (abstract) splits into IfcProduct (physical things like walls) and IfcProcess (tasks or schedules).

c. IfcProduct branches into IfcElement (abstract, for building components) and IfcSpatialElement (spaces like rooms).

d. IfcElement splits further into sub-classes used in building projects:

(i) IfcBuiltElement

Contains architectural and structural elements. Covers things like walls, doors, windows, beams, columns, slabs, and stairs. Inherits from IfcElement, adding traits for building structure.

(ii) IfcDistributionElement

Contains services entities. Includes pipes, ducts, fittings, terminals, and electrical appliances for MEP systems. Inherits from IfcElement, focusing on building services. This class is further specialised into IfcDistributionFlowElement and IfcDistributionControlElement.

(iii) IfcFeatureElement

Contains additions or subtractions. Handles openings (cuts) and projections (extras) that modify other elements. Inherits from IfcElement, tweaking the main structure.

e. IfcSpatialElement:

(i) An abstract entity for spatial organization—like giving every element in a building an address in a structured way: site, block, floor, room.

(ii) It defines areas or volumes in the project, such as the site (IfcSite), the building (IfcBuilding), a floor (IfcBuildingStorey), or a room (IfcSpace).

(iii) It inherits from IfcProduct, adding traits like spatial boundaries and containment. IfcSpatialElement acts like a map, linking physical parts—like an IfcWall—to their locations, ensuring everything has a clear spot in the model.

5. For easier understanding, the concrete IFC entities (IFC Classes) of a BIM element can be considered analogous to the CAT code (Category) of the Master Type List maintained by CIC (https://www.bim.cic.hk/en/resources/master_list) referred to in the DEVB BIM Harmonisation Guidelines for Works Departments.

6. The semantic definition of all IFC Type can be looked up in the IFC documentation on buildingSMART's official website.

6.1.3.41 IfcWall

NOTE This entity is a subtype of IfcProduct or IfcTypeProduct and hence part of every standardized schema subset and implementation level.

6.1.3.41.1 Semantic definition

The wall represents a vertical construction that may bound or subdivide spaces. Wall are usually vertical, or nearly vertical, planar elements, often designed to bear structural loads. A wall is however not required to be load bearing.

A wall may have openings, such as wall openings, openings used for windows or doors, or niches and recesses. They are defined by an *IfcOpeningElement* attached to the wall using the inverse relationship *HasOpenings* pointing to *IfcRelVoidsElement*. Walls with openings that have already been modeled within the enclosing geometry may use the relationship *IfcRelConnectsElements* to associate the wall with embedded elements such as doors and windows.

Semantic Definition of IfcWall, extracted from buildingSMART official website

3.2. IFC PredefinedType and Enumeration

1. IFC Type Enumeration is a way to provide fixed lists of options, called enumerations, for entity types or properties in IFC.

2. These options are stored in the "PredefinedType" attribute, which limits choices to keep data clear and consistent. For example,

   a. IfcWallType's PredefinedType might offer "SOLIDWALL," "PARTITIONING," or "SHEAR," so every wall of that type is one of those.

   b. IfcDoorType's PredefinedType could list "SINGLE," "DOUBLE," or "SLIDING."

   c. IfcFlowTerminalType (under IfcDistributionElement) might include "SINK," "SHOWER," or "AIRTERMINAL."

3. Enumerations stop vague inputs—like "SINGLE" always meaning one panel—making data reliable across software and teams.

4. IFC 4.3 has hundreds of these lists, tied to entities under IfcBuiltElement, IfcDistributionElement, and IfcFeatureElement, standardizing how building parts are defined.

5. This "PredefinedType" attribute is tied to IFC Type entities (like IfcWallType) and sometimes concrete entities (like IfcWall). For instance, an IfcWall not linked to a type can still have a PredefinedType of "STANDARD," but usually, it's the Type entity that carries the enumeration.

6. Lists of these PredefinedType should refer to the buildingSMART official IFC documentation. For example:

   a. *IfcWallTypeEnum - https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcWallTypeEnum.htm*

   b. IfcDoorTypeEnum - https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcDoorTypeEnum.htm

7. For easier understanding, the PredefinedType of a BIM element can be considered analogous to the sub CAT code outlined in the DEVB BIM Harmonisation Guidelines for Works Departments.

8. Example of PredefinedType Enumeration:

6.1.2.22 IfcWallTypeEnum

6.1.2.22.1 Semantic definition

6.1.2.22.2 Type values

Type	Description
ELEMENTEDWALL	A stud wall framed with studs and faced with sheetings, sidings, wallboard, or plasterwork. IFC4.3.2.0-CHANGE This enumeration value is no longer deprecated. NOTE An elemented wall can be modelled by means of an aggregation of various components (use *IfcRelAggregates* to relate to the *IfcWall*), but when this amount of detail is not necessary or available this enumeration value can be used instead.
MOVABLE	A movable wall that is either movable, such as folding wall or a sliding wall, or can be easily removed as a removable partitioning or mounting wall. Movable walls do normally not define space boundaries and often belong to the furnishing system.
PARAPET	A wall-like barrier to protect human or vehicle from falling, or to prevent the spread of fires. Often designed at the edge of balconies, terraces or roofs, or along edges of bridges.
PARTITIONING	A wall designed to partition spaces that often has a light-weight, sandwich-like construction (e.g. using gypsum board). Partitioning walls are normally non load bearing.
PLUMBINGWALL	A pier, or enclosure, or encasement, normally used to enclose plumbing in sanitary rooms. Such walls often do not extend to the ceiling.
POLYGONAL	A polygonal wall, extruded vertically, where the wall thickness varies along the wall path. IFC4-DEPRECATION The value is deprecated and shall no longer be used.
RETAININGWALL	A supporting wall used to protect against soil layers behind. Special types of a retaining wall may be e.g. Gabion wall and Grib wall. Examples of retaining walls are wing wall, headwall, stem wall, pierwall and protecting wall.

3.3. IFC Element Type

1. IFC Type entities are standalone definitions for shared characteristics of multiple instances of an element—not properties of other entities.

2. They start with "Ifc" and end with "Type" (e.g., IfcWallType) and pair with concrete entities (e.g., IfcWall) via the IfcRelDefinesByType relationship.

3. Types let you set details once—like a 200mm thick concrete wall in IfcWallType—and link every matching IfcWall to it, avoiding repetition.

4. They're abstract, inheriting from IfcTypeObject (under IfcObjectDefinition), which gives them a name and optional property sets (e.g., Pset_WallCommon).

5. For example, an IfcDoorType might define all doors as "SINGLE" with a 1-hour fire rating, and each IfcDoor in the high-rise lobby links to it.

6. Types exist for most IfcElement sub-classes:

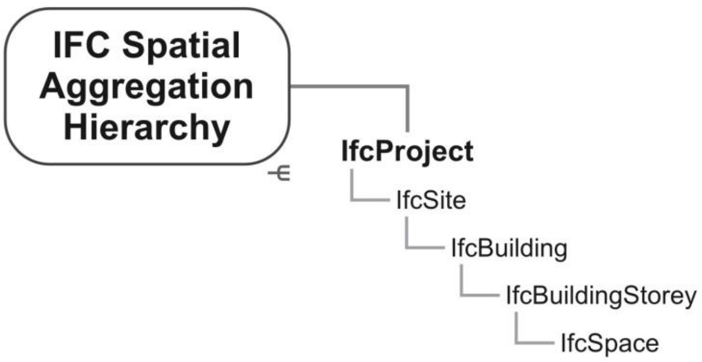
    a. IfcBuiltElement has IfcWallType or IfcBeamType,

    b. IfcDistributionElement has IfcFlowSegmentType for pipes, and

    c. IfcFeatureElement has IfcOpeningElementType for standard cuts.

7. For easier understanding, the IFC type of a BIM element can be considered analogous to the family type of an BIM object in Revit.

3.4. Spatial Structure Hierarchy



1. The spatial structure hierarchy in IFC organizes a building project into nested levels using IfcSpatialElement sub-classes.

2. It starts with:

    a. IfcProject (the whole project),

    b. then IfcSite (e.g., "66 Queensway, Admiralty, Hong Kong"),

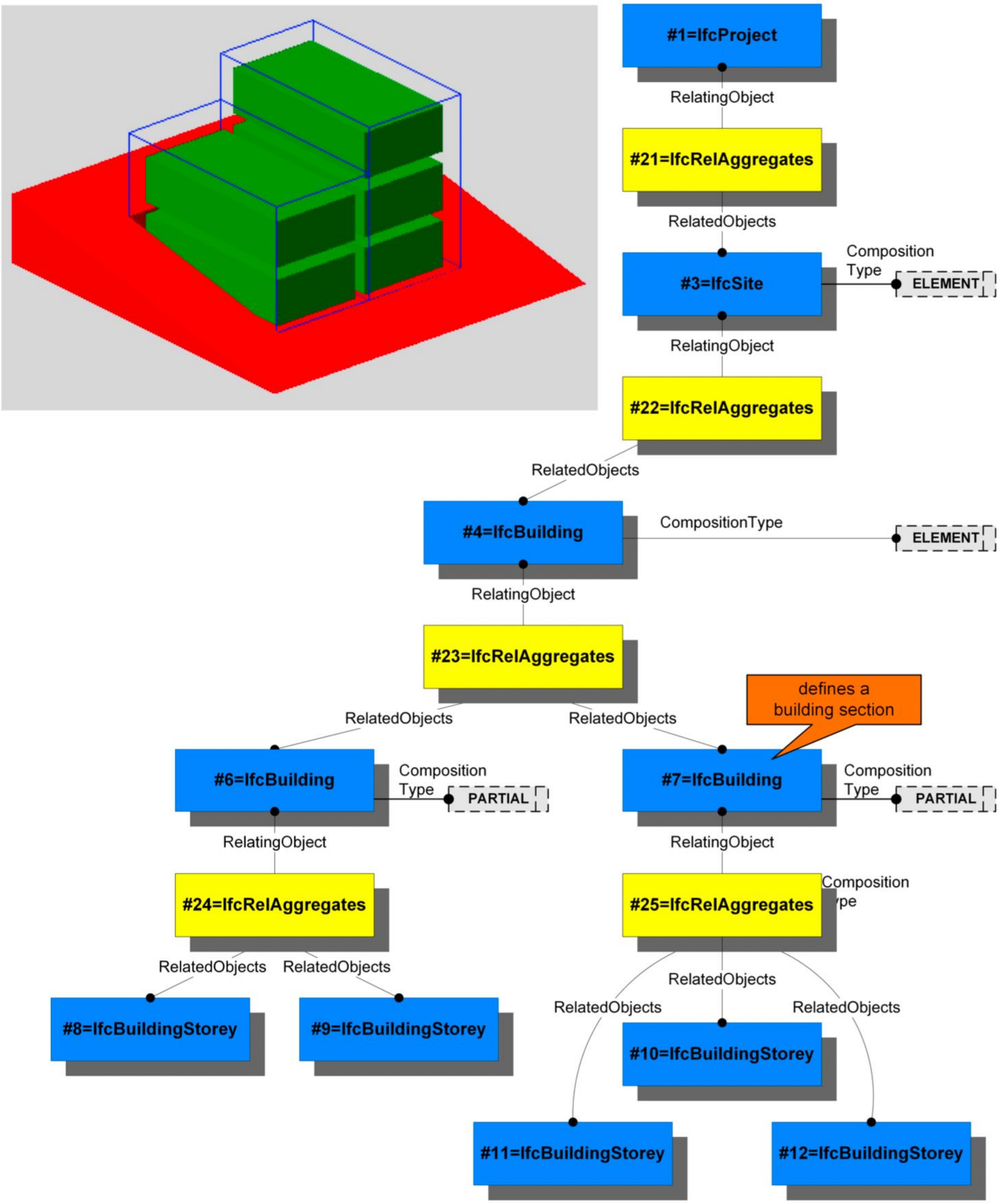
    c. IfcBuilding (e.g., "Queensway Government Office Building"),

d. IfcBuildingStorey (e.g., "36/F"), and

e. IfcSpace (e.g., "CAVE").

f. An IfcWall in the CAVE room links to its IfcSpace via IfcRelContainedInSpatialStructure, nesting up through 36/F, the building, and the site, all under the project.

3. IfcRelAggregates groups these levels, keeping the model structured so teams can pinpoint elements like a lobby door versus a 36th-floor wall.



Spatial structure element composition

https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcSpatialStructureElement.htm

3.5. Attributes, PropertySet (Pset) and QuantitySet (Qset)

1. Each element in IFC files contains information related to them. This information is divided into Attributes, PropertySet and QuantitySet.

   a. Attributes

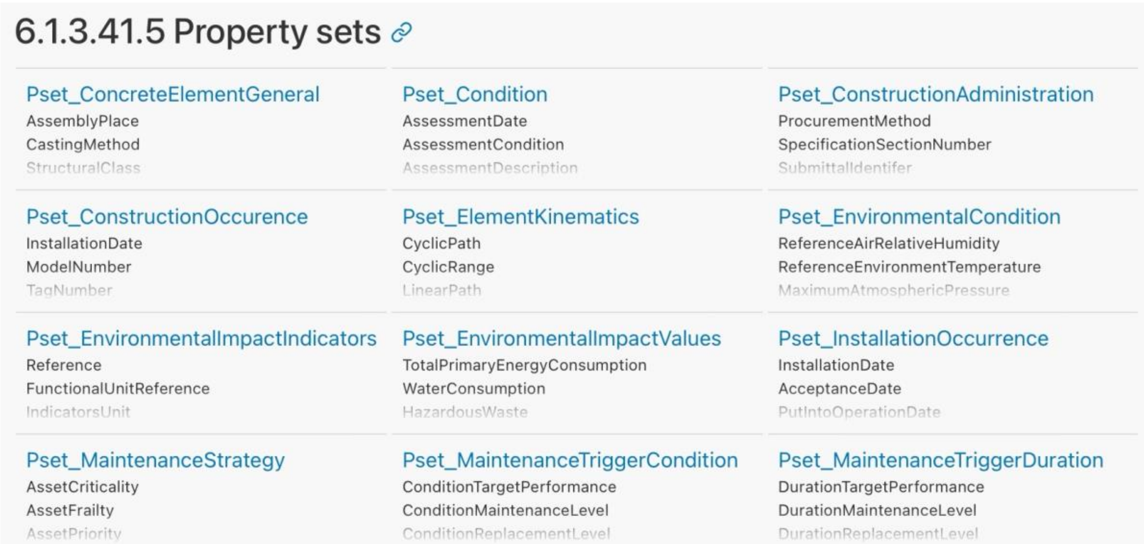
      (i) They are absolute minimum for defining an element. These attributes includes GlobalId, Name, Object Type, etc.

   b. PropertySet

      (i) These are properties additional to attributes.

      (ii) There are standardised property sets in IFC Data Schema. They have prefix "Pset_"

   c. Example, for IfcWall:



6.1.3.41.5 Property sets

Pset_ConcreteElementGeneral AssemblyPlace CastingMethod StructuralClass	Pset_Condition AssessmentDate AssessmentCondition AssessmentDescription	Pset_ConstructionAdministration ProcurementMethod SpecificationSectionNumber SubmittalIdentifer
Pset_ConstructionOccurence InstallationDate ModelNumber TagNumber	Pset_ElementKinematics CyclicPath CyclicRange LinearPath	Pset_EnvironmentalCondition ReferenceAirRelativeHumidity ReferenceEnvironmentTemperature MaximumAtmosphericPressure
Pset_EnvironmentalImpactIndicators Reference FunctionalUnitReference IndicatorsUnit	Pset_EnvironmentalImpactValues TotalPrimaryEnergyConsumption WaterConsumption HazardousWaste	Pset_InstallationOccurrence InstallationDate AcceptanceDate PutIntoOperationDate
Pset_MaintenanceStrategy AssetCriticality AssetFrailty AssetPriority	Pset_MaintenanceTriggerCondition ConditionTargetPerformance ConditionMaintenanceLevel ConditionReplacementLevel	Pset_MaintenanceTriggerDuration DurationTargetPerformance DurationMaintenanceLevel DurationReplacementLevel

https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcWall.htm

      (i) User may also add customised properties to meet the information requirements set out in EIR / BEP.

   d. QuantitySet

      (i) These are standardised property sets in IFC Data Schema. They have prefix "Qto_".

      (ii) They hold quantity related values that are automatically calculated by the native authoring softwares.

      (iii) They should not be customised.

3.6. Geometry Representations

1. The IFC4.3 schema (latest as IFC4.3.2.0 per ISO 16739-1:2024) defines various types of geometry representations via IfcShapeRepresentation for element shapes, categorized by RepresentationType strings (e.g., 'CSG', 'Brep', 'SweptSolid'). Below are key types:

a. CSG (Constructive Solid Geometry): Represents shapes by combining primitive solids (e.g., cubes, spheres, cylinders) through Boolean operations like union, intersection, and difference.

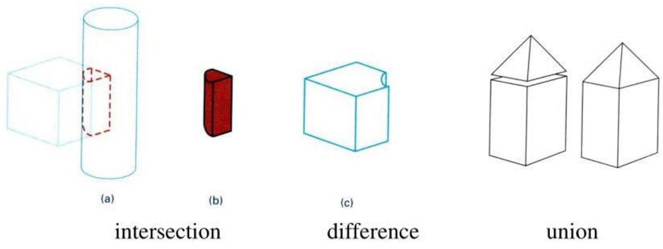


Figure 1: CSG illustration (source: https://www.slideserve.com/liam/modeling-and-the-viewing-pipeline)

b. BREP (Boundary Representation): Defines object geometry via boundaries, including surfaces, edges, and vertices.

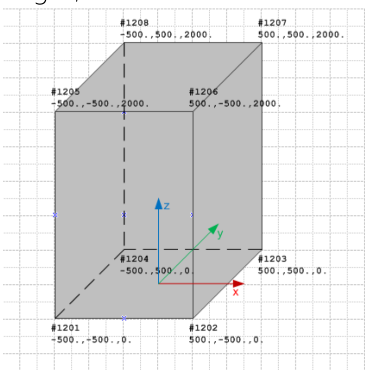


Figure 2: shows the block geometry represented by a faceted boundary representation (source: https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/annex_e/basic-geometric-shape/brep-model.html#Figure-E.B-shows-the-block-geometry-represented-by-a)

(i) Faceted Brep: Simplified BREP using planar facets (polygonal faces) for tessellated, approximate representations.

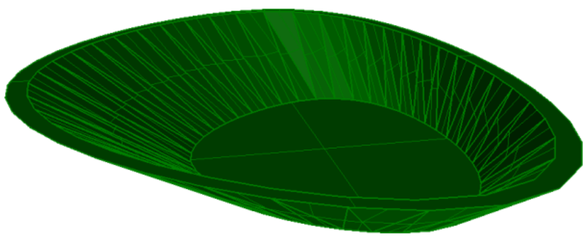


Figure 3: faceted Brep (source: https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/annex_e/advanced-geometric-shape/basin-faceted-brep.html)

(ii) Advanced Brep: Complex BREP using advanced faces with B-spline surfaces and curves (supporting true curves), including options with or without voids.

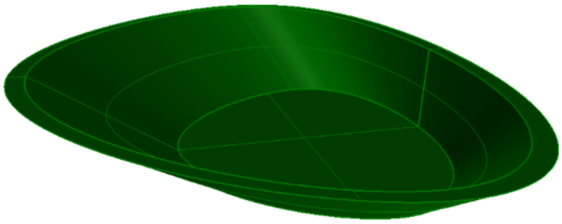


Figure 4: dvanced brep representation using NURBS (source: https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/annex_e/advanced-geometric-shape/basin-advanced-brep.html)

c. SweptSolid: Creates solids by sweeping a 2D profile (area) along a path via extrusion or revolution, excluding tapered sweeps.

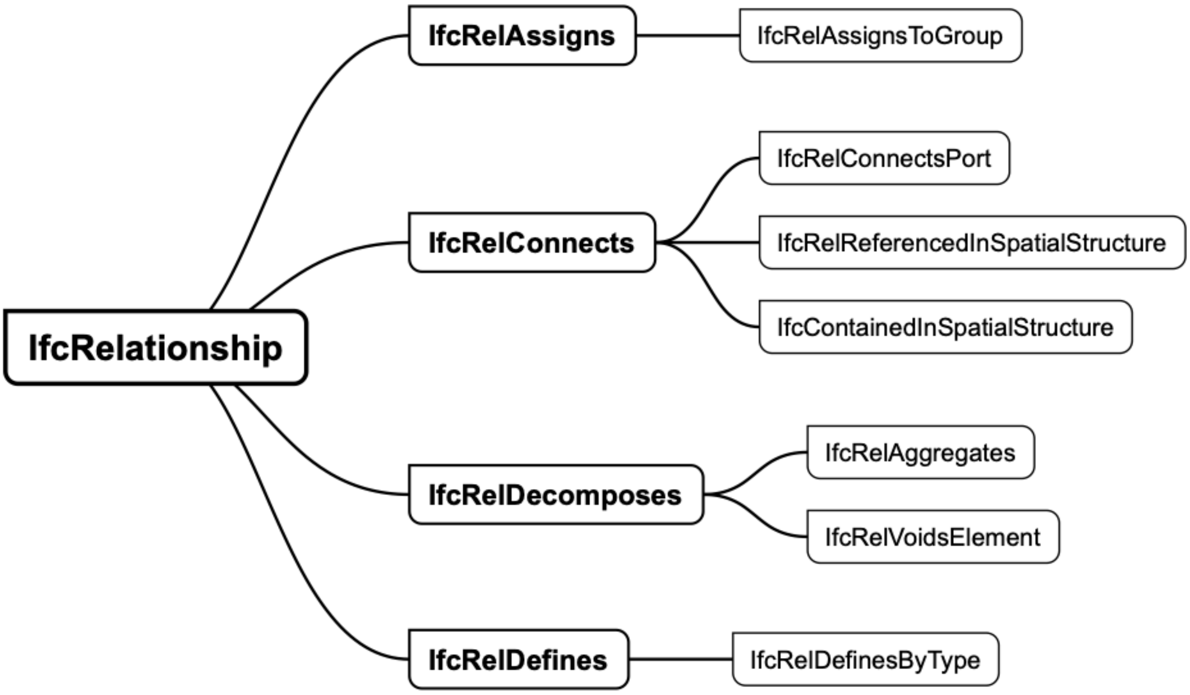
d. AdvancedSweptSolid: Extends swept solids to include tapered sweeps and more complex directrix-based profiles.

2. Although the IFC schema fully supports true curve free-form geometry (e.g., via Advanced Brep with B-splines), the IFC4 Reference View (RV) MVD—commonly used for design coordination—restricts geometry to tessellated shapes for broad compatibility and does not support advanced free-form features like true curves or B-splines.

3. Consequently, most BIM element geometries, when exported or converted to comply with IFC4 RV, are faceted as tessellated surface models or Faceted Brep representations.

3.7. Relationship

Various relationships can be defined among IFC entities. Relationship is an important concept and allows discrete BIM elements to relate to each other for different BIM uses. The following shows commonly encountered relationships



1. **Object location**

   a. IfcRelContainedInSpatialStructure links elements (IfcProduct) to a location (IfcSpatialElement), e.g. IfcSpace, IfcBuildingStorey, etc.

   b. This relationship is useful for generating lists of inventory based on location, e.g. in each room or storey.

   c. Depending on the project setup, some BIM authoring tool could automatically generate these relationships.

2. **Object Groups**
   By assigning BIM elements to certain group (IfcGroup) via IfcRelAssignsToGroup such as MEP equipment in a system. See Section Grouping.

3. **MEP connections**
   MEP routing and equipment are connected via IfcPort and IfcRelConnectsPort. An MEP element may have multiple ports for connection to other elements.

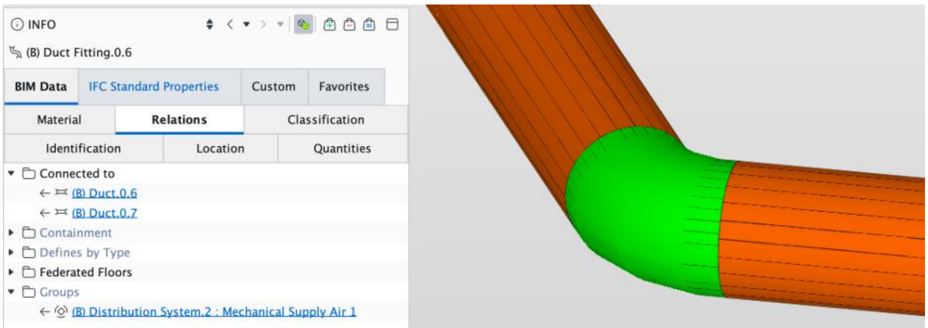


Figure 5: A Revit exported IFC example showing highlighted duct fitting is connected to "Duct 0.6" and "Duct 0.7", automatically generated during export

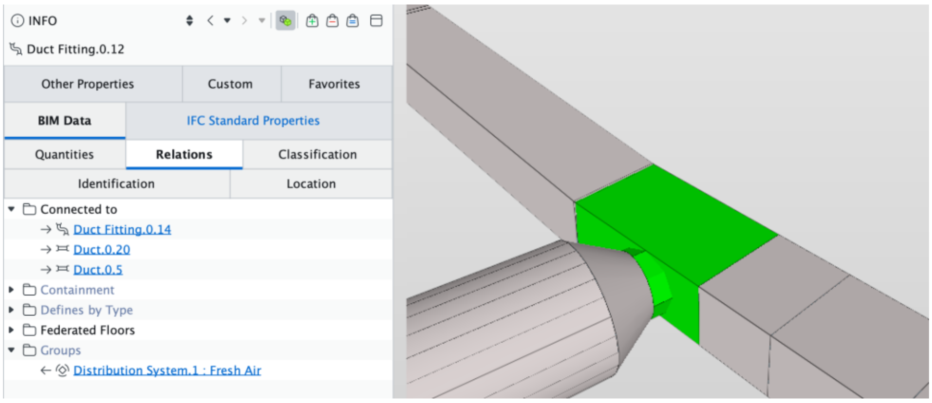


Figure 6:: An Archicad exported IFC example shoring highlighted duct fitting is connected to 3 other ducts accessories, automatically generated during export

4. **Reference Locations of Systems**

IfcRelReferencedSpatialStructure connects a system to the spatial elements it serves. Also in additional IfcRelContainedInSpatialStructure for spatial containment, A curtain wall might span through several stories, in this case it can be contained within the ground floor, but it would be referenced by all additional stories it spans.

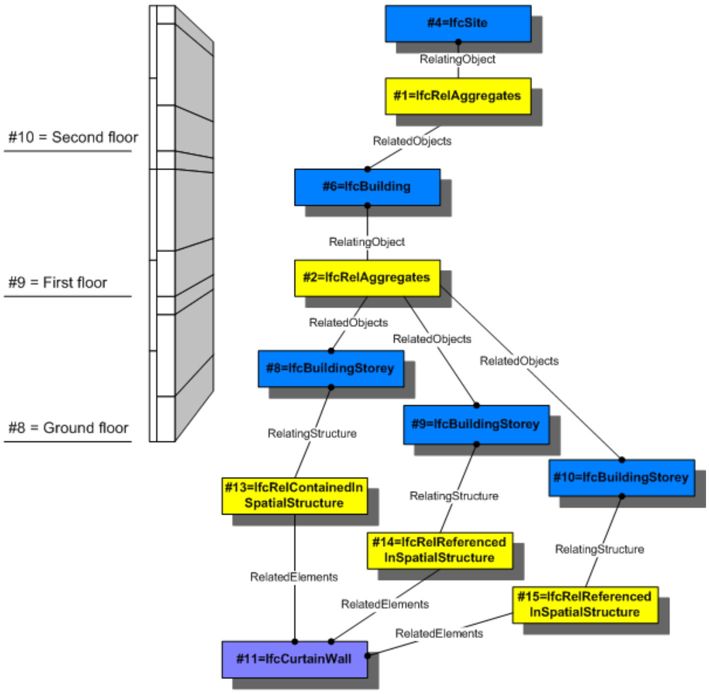
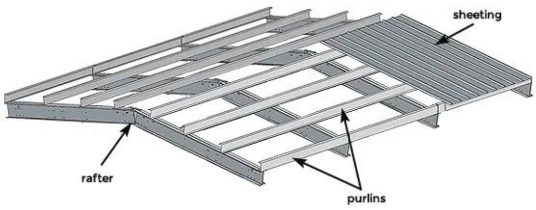


Figure 7: a curtain wall referenced to three storeys (source: buildingSMART)

5. **Whole and Part Relationship**

IfcRelAggtegates establishes whole / part relationship among elements. For example, a roof is the aggregation of roof elements such as roofing sheets, rafters, purlins[1].



6. **Void in an Element**

IfcRelVoidsElement is an objectified relationship between a building element and one opening element that creates a void in the element. It is a one-to-one relationship. This relationship implies a boolean operation of subtraction between the geometric bodies of the element and the opening.[2]

[1] https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcRelAggregates.htm

[2] https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcRelVoidsElement.htm

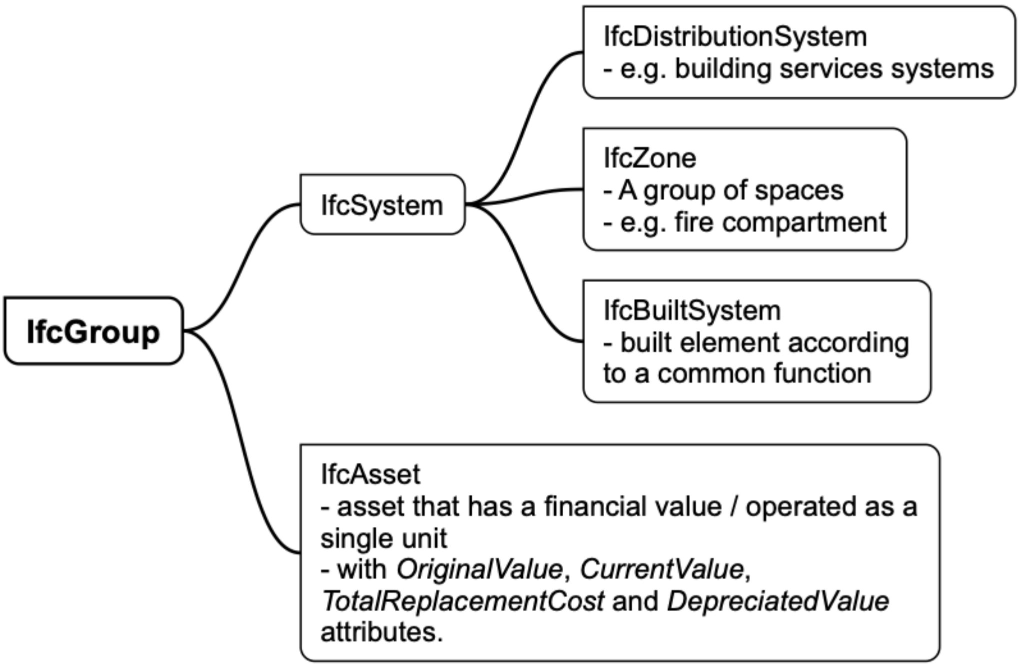
7. **Assigning Object to Object Types**

IfcRelDefinesByType defines the relationship between an object type (Similar to Family Types in Revit) and object occurrences. The IfcRelDefinesByType is a 1-to-N relationship, as it allows for the assignment of one type information to a single or to many objects.[3] For example, one door type can be installed in one location or many locations in a building.

[3] https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcRelDefinesByType.htm

3.8. Grouping

1. Grouping, provided as IfcGroup, allows entities to be organized into collections. The following diagram are some commonly encountered grouping and their hierarchy in the IfcGroup Tree:



a. Zone Grouping:

(i) IfcZone Type is used for grouping of spaces (IfcSpace), which may or may not be adjacent.

(ii) Common use cases are:

- Fire compartmentation
- Residential units (consist of living room, bedroom, kitchen, etc.)

b. Building Service System Grouping:

(i) The group IfcDistributionSystem defines the occurrence of a specialized system for use within the context of building services or utilities for built facilities.[4]

(ii) Commonly used for for grouping MEP elements into systems and routings under IfcDistructionSystem.

c. Built System

[4] (https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcDistributionSystem.htm)

(i) A built system (IfcBuiltSystem) is a group by which built elements are grouped according to a common function within the facility.

(ii) Possible Use Cases are:

- Facade / Roofing system
- MiMEP modules
- MiC modules

d. Assets

(i) An asset (IfcAsset) is a uniquely identifiable grouping of elements acting as a single entity that has a financial value or that can be operated on as a single unit. An asset is generally the level of granularity at which maintenance operations are undertaken. An asset is a group that can contain one or more elements. Whilst the financial value of a component or element can be defined, financial value is also defined for accounting purposes at the level of the asset.[5]

(ii) IfcAsset has standard attributes: OriginalValue, CurrentValue, TotalReplacementCost and DepreciatedValue

2. Group Assignment

a. E&M Elements can be assigned to these groups by a relation called IfcRelAssignsToGroup.

b. One BIM elements can be part of zero, one or many groups

3. Software Implementation

a. Some softwares (e.g. Revit) may not provide built-in functions for (automatically or manually) assigning these grouping relationships. Users of those softwares may use different strategies to achieve element group which may or may not make use of IfcGroup type. For example, some exchange information requirement may require user to create a custom parameter for assigning the system a piece of equipment they belong to, instead of directly assigning the equipment to a "system" in native bim software.

3.9. Space Boundaries

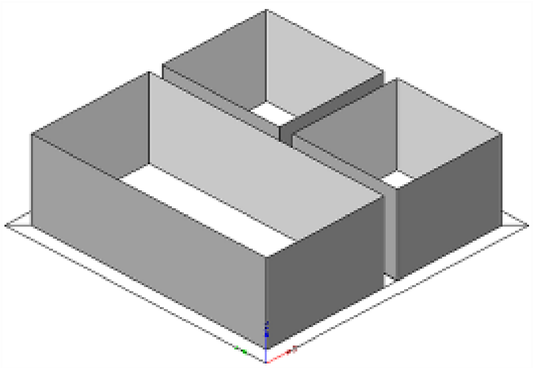
1. Space boundaries in IFC define the borders of spaces within a building model, such as rooms or zones, by establishing relationships between a space and the surrounding building elements

5 https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcAsset.htm

(e.g., walls or floors). These boundaries facilitate accurate calculations for area, volume, energy analysis, and spatial relationships, ensuring the model represents how spaces interact without overlaps or gaps.

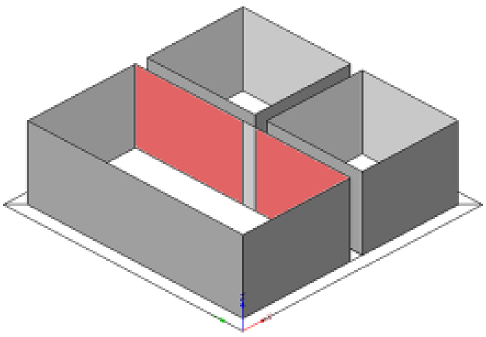
2. The IFC standard distinguishes two levels of detail for space boundaries:

   a. 1st Level Space Boundaries: These provide a basic representation focused solely on the space's internal perspective, outlining its shape without considering adjacent elements or materials. They are suitable for architectural planning or simple facility management.

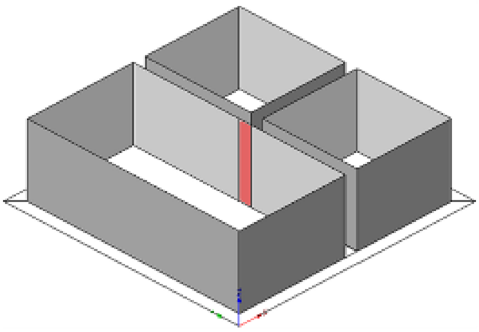
   

   b. 2nd Level Space Boundaries: These offer a more detailed representation that accounts for conditions on the other side of the boundary:

      (i) Type A: Applies when the boundary adjoins another space (e.g., a shared wall between rooms).

      (ii) Type B: Applies when the boundary meets a building element or the exterior (e.g., an external wall). This level supports advanced applications like energy simulations, where heat transfer or adjacency must be evaluated.

3. Boundaries are further **categorized** as:

a. Physical: Linked to tangible elements, such as a concrete wall.

b. Virtual: Representing non-physical dividers, such as zones in an open area.

c. Internal: Within the building (e.g., between interior spaces).

d. External: Facing the outside environment (e.g., a facade or balcony edge).

4. In the IFC schema, space boundaries are modeled using the IfcRelSpaceBoundary entity, which includes attributes like a unique identifier, optional geometric descriptions (e.g., curves or surfaces), and connections to the related space and bounding element. This structure enables precise, data-driven analysis in BIM software.

3.10. Classification

3.10.1. Why Classification (or Element Coding) Is Needed

1. **Identification**
   Assigning each BIM object a clear classification (e.g. IfcWall, Revit "Wall") ensures every stakeholder knows exactly what they're looking at.

2. **Downstream Process Integration**
   Classification links model elements to cost estimates (e.g. HKSMM4), schedules (4D), clash-detection rules, FM databases, and handover protocols.

3. **Structured Organisation**
   A consistent hierarchy—from broad categories (e.g. "Structural Elements") to detailed sub-types (e.g. "Concrete Beam—Cast In Situ")—makes models easier to browse, filter, and review for quality control.

4. **Local Standards and Future Proof**
   By layering Hong Kong–specific codes (e.g. DEVB Harmonization Guide's Object Naming or EMSD MEP codes) on top of IFC or software defaults, interoperability can be maintained while accommodating local regulations, trade practices, and future extensions.

3.10.2. Underlying Paradigms of Classifications

In Hong Kong projects, classifications generally follows three paradigms, each addresses a unique aspect (or dimension) of the construction process, organizing information in distinct ways

1. **Element-Hierarchy–Based:**
   Focuses on **physical components** (e.g., in BIM, a concrete column is classified under "Structural Elements > Columns > Concrete Columns"), arranging them by their properties and relationships.

2. **Trade/Work-Section–Based:**
   Focuses on **work activities** (e.g., HKSMM4 groups formwork, reinforcement, and concrete pouring under "Concrete Works"), grouping them by trade or contract sections.

3. **System/Product-Based:**
   Focuses on **functional systems** (e.g., MEP coordination classifies ducts, AHUs, and controls under "HVAC System"), combining components based on how they function together.

3.10.3. IFC's Native Classification

1. An IFC model natively use IFC entity types to define geometry, relationships, and behavior:

   a. IFC entities form an inheritance tree under IfcRoot (for example, IfcWall inherits from IfcElement, which inherits all the way up from IfcRoot).

   b. Many entities carry a PredefinedType attribute (e.g. IfcWallType.PredefinedType = PARTITIONING) for finer distinctions.

2. The IfcClassification and IfcClassificationReference mechanism lets any object link to external classification entries such as Omniclass.

3.10.4. Key Hong Kong Classification & Coding Systems

Local practices also apply local systems—some BIM-native, others pre-BIM—to address the needs and workflows for their trades such as costing, specification, and handover, etc.

Element-Hierarchy–Based			
System / Standard	Owner / Source	Main Focus	Levels
IFC Entity Type & PredefinedType	buildingSMART	OpenBIM schema and entity taxonomy	Inheritance tree + flat enumeration
Revit Category	Autodesk	Discipline categories → Families → Types	Category → Subcategory → Family → Type
Archicad Classification v2.0	Graphisoft	Deep, project-specific object classes	Up to five nested levels
Tekla Entity Type	Trimble	Structural/steel object types	Flat list + user-defined tags
CIC's BIM Object Guide	CIC	Object naming taxonomy	Discipline → System → Family → Type → Variant

Trade/Work-Section–Based			
System / Standard	Owner / Source	Main Focus	Levels
DEVB Object Naming	HK Gov – DEVB	Discipline → System → Component naming	Three concatenated code segments
ArchSD Elemental Code Relation Table	HK Gov – ArchSD	Trade sections → Elemental codes	Section → Clause
HKSMM4 Classification Table (HKIS)	HK Institute of Surveyors	Bills of quantities (measurement standard)	Section → Sub-section → Group → Item

System/Product/Performance–Based			
System / Standard	Owner / Source	Main Focus	Levels
EMSD BIM Standard	HK Gov – EMSD	MEP System, Routing, Equipment codes	Three parallel hierarchies
CIC BIM Standard LOIN Tables	CIC	Level of Information Need checklists	Stage-based requirements (no hierarchy)

3.10.5. Challenges of Direct Mapping Between Classification Systems

Mapping directly between different classification systems is generally not feasible for the following reasons:

1. Different Organisation

a. Different classification systems organise information based on distinct dimensions - components, activities and functions.

2. Granularity Mismatch

 a. **Trade-Based Aggregation**
 A single trade item (e.g., "Finishes") may cover entire assemblies (walls, finishes, skirtings), while BIM tools classify each component separately.

 b. **System/Product-Based Complexity**
 Classifications by functional systems (e.g., HVAC) often operate at an intermediate detail level, neither as granular as elements nor as broad as trades, complicating mappings.

3. Many-to-Many Relationships

 a. **Overlapping Classifications:**
 A single BIM object (e.g., insulated wall) may span multiple trade codes (structure, insulation, finish), while a trade code (e.g., "Plumbing") covers multiple object types (pipes, valves).

 b. **System/Product Overlaps:**
 System-based groupings (e.g., "Lighting Fixtures") can include components from various trades or element types, adding complexity.

4. Independent Evolution
 Classification standards and BIM software evolve on separate schedules. Changes in code values, hierarchy structures or entity definitions can quickly render any static cross-reference obsolete.

3.10.6. Tagging BIM Objects for Different Classification Paradigms

The following outlines that different tagging strategies should be used for different classification paradigms. While these strategies are straightforward way provided by the IFC schema, local practices may have other strategies to suit their software and system.

1. Element-Hierarchy-Based Classification
 Directly setting BIM objects to correct IFC types (e.g. IfcWall, IfcBeam), and if needed, the PredefineType.

2. Trade/Work-Section-Based Classification
 To align BIM objects with trade-based systems like HKSMM4, use one of the following approaches:

 a. Grouping:
 Combine multiple BIM objects into a single group and assign a trade code to the entire group. This is ideal for trade codes that apply to entire systems or assemblies, like a complete HVAC system.

 For example In IFC: Use IfcGroup to create a collection (e.g., a plumbing system) and assign the HKSMM4 code via IfcClassificationReference.

 b. **Decomposing**:
 Break down complex BIM objects into their constituent parts (e.g., split a wall into structural,

insulation, and cladding layers), and assign each part its specific trade code.

Example: A wall decomposed into structural core (tagged with a structural trade code), insulation (tagged with an insulation trade code), and cladding (tagged with a cladding trade code).

c. **Dual-Coding**:
Assign multiple classification codes to a single object, including both element-based and trade-based codes.

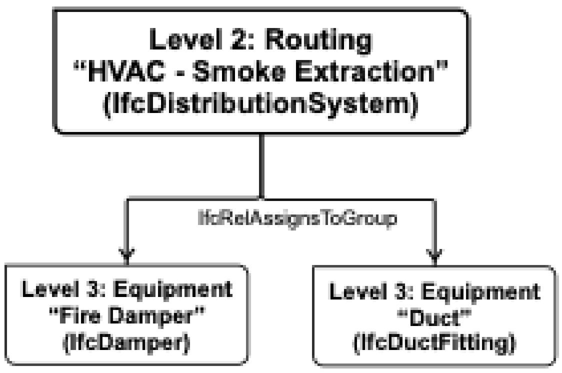
3. System/Product-Based Classification

a. IFC provides a straightforward 2-tier classification for system-equipment hierarchies in building services, aligning with basic MEP workflows:

(i) Define the overall system and routing using IfcDistributionSystem (e.g., for HVAC – Smoke Extraction or Electrical Distribution - Cable Containment for Low Voltage);

(ii) then define equipment elements in their appropriate IFC classes (e.g., IfcPump, IfcDuctSegment, etc);

(iii) and assign these elements directly to the system via IfcRelAssignsToGroup, creating a simple parent-child relationship without intermediate layers.



b. For more granular 3-tier classification (System -> Routing -> Equipment), IFC accommodates this through nested systems:

(i) Define the top-level system as an IfcDistributionSystem entity (e.g., overall HVAC);

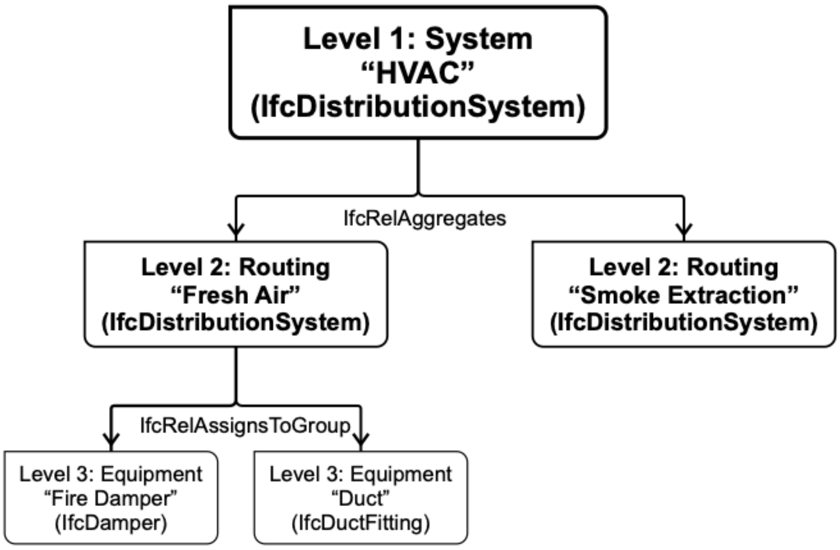
(ii) define routing as a subsystem using another IfcDistributionSystem (e.g., Smoke Extraction subsystem);

(iii) establish the parent-child relationship between the system and routing subsystem with IfcRelAggregates;

(iv) define equipment in appropriate IFC classes (e.g., IfcDamper, IfcDuctFitting, etc.);

(v) and assign equipment to the routing subsystem via IfcRelAssignsToGroup.

This structure supports hierarchical decomposition while maintaining semantic integrity and interoperability in IFC exports.



3.11. IFC schema

1. IFC has evolved through different versions to meet the building industry's needs, with IFC2x3, IFC4, and IFC4.3 being the officially used ones today.

Version	Name (HTML Documentation)	ISO publication	Published (yyyy-mm)	Current Status	Full package	EXPRESS	XSD	pSet XSD	OWL HTML	RDF	TTL
5.0 dev	IFC 5 dev			Next generation IFC. Refactoring of the IFC 4.x content. Currently under development.	GitHub	n.a.		n.a.			
4.4 - dev	IFC 4.4.x dev			Potential extension of 4.3. Adding additional functionality. Currently in planning phase.							
4.3.2.0	IFC 4.3 ADD2	ISO 16739-1:2024	2024-04	Official	ZIP	EXP	XSD	ZIP	ifcOWL		TTL (use correct MIME type)
4.2.0.0	IFC 4.2	-	2019-04	Withdrawn	ZIP	EXP	IFC4x2.xsd	-			
4.1.0.0	IFC 4.1	-	2018-06	Withdrawn	ZIP	EXP	IFC4x1.xsd	-			
4.0.2.1	IFC 4 ADD2 TC1	ISO 16739-1:2018	2017-10	Official	ZIP	EXP	IFC4.xsd	-	ifcOWL	RDF	TTL
4.0.2.0	IFC 4 ADD2	-	2016-07	Retired	ZIP	EXP	IFC4_ADD2.xsd	-			
4.0.1.0	IFC 4 ADD1	-	2015-06	Retired	ZIP	EXP	IFC4_ADD1.xsd	-			
4.0.0.0	IFC 4	ISO 16739:2013	2013-02	Retired	ZIP	EXP	ifcXML4.xsd	PSD_IFC4.xsd	ifcOWL	RDF	TTL
2.3.0.1	IFC 2x3 TC1	ISO/PAS 16739:2005	2007-07	Official	ZIP	EXP	IFC2X3.xsd	PSD_R2x3.xsl	ifcOWL	RDF	TTL
2.3.0.0	IFC 2x3	-	2005-12	Retired	ZIP	EXP	-	-	ifcOWL	RDF	TTL
2.2.1.0	IFC 2x2 ADD1	-	2004-07	Retired	ZIP	EXP	-	-	-	-	-
2.2.0.0	IFC 2x2	-	2003-05	Retired	ZIP	EXP	-	-	-	-	-
2.1.1.0	IFC 2x ADD1	-	2001-10	Retired	ZIP	EXP	-	-	-	-	-
2.1.0.0	IFC 2x	-	2000-10	Retired	ZIP	EXP	-	-	-	-	-
2.0.0.0	IFC 2.0	-	1999-10	Retired	-	-	-	-	-	-	-

The table IFC Release Database was last modified at 2024-11-13 07:42:16 by Léon van Berlo.

IFC Release Database showing the history of all current and previous IFC releases

2. IFC2x3 (2.3.0.1 - IFC 2x3 TC1)

   a. Baseline schema

   b. IFC2x3, released in 2005 and updated through 2007, latest being IFC2x3 TC1 - is widely adopted for basic building models but lacks newer features.

   c. It is widely adopted for basic building models but lacks newer features.

3. IFC4 (4.0.2.1 - IFC 4 ADD2 TC1)

   a. IFC4, finalized in 2013 with addenda up to 2018 (ISO 16739-1:2018), adds better geometry, properties, and infrastructure support, making it more robust.

b. Full ISO standard

c. Enhanced capability of the IFC specification in its main architectural, building service and structural elements with new geometric, parametric and other features

d. Enabled numerous new BIM workflows – including 4D and 5D model exchanges, manufacturer, product libraries, BIM to GIS interoperability, enhanced thermal simulations and sustainability assessments

e. Linking all IFC property definitions to the buildingSMART data dictionary

f. Multilingual translations of the schema

4. IFC4.3 (4.3.2.0 - IFC 4.3 ADD2)

   a. IFC4.3, the latest as of 2024 (ISO 16739-1:2024), expands further for complex projects like rail and roads, with over 1300 entities and 2500 properties

   b. The main purpose of IFC4.3 is to extend the IFC schema to cover the description of infrastructure constructions within the domains of Railways, Roads, Ports and Waterways including the elements that are common across those domains.

5. These versions are in active use, depending on project demands, while older ones like IFC2x2 fade out and IFC5 is in development.

3.12. IFC File Formats Overview

1. IFC files support multiple encoding formats—STEP, XML, and JSON—to accommodate diverse needs like manual editing, system integration, or web applications. Here's a comparison:

Format	File Extension	Description	Human-Readability	Editability (Manual)	Typical Uses	Notes/Verification
STEP (SPF)	.ifc	Text-based structure using ISO 10303-21; compact, human-readable lines like "#123=IFCWALL(…,'200mm',…)" for elements like wall thickness.	High (simple text)	High (editable in Notepad or text editors)	Daily model exchanges between BIM tools; hands-on import/export.	Verified as most compact and widely used; official standard; recommended for smallest size and widest compatibility.
IFC-XML	.ifcXML	XML markup based on ISO 10303-28; structured and verbose for machine parsing.	Moderate (structured but verbose)	Moderate (possible but cumbersome manually)	System integration, databases, or software parsing.	Verified; size ~113% of STEP; official with enhanced tool support, but less compact than STEP.
IFC-JSON	.json (.ifc.json also used)	JSON structure; web-friendly for data transfer.	High (compact syntax)	High (editable in text editors)	Modern web-based apps, real-time data, APIs.	Verified as provisional/candidate standard; size ~148% of STEP (not compact); community-supported but not fully official; web-friendly despite larger size.

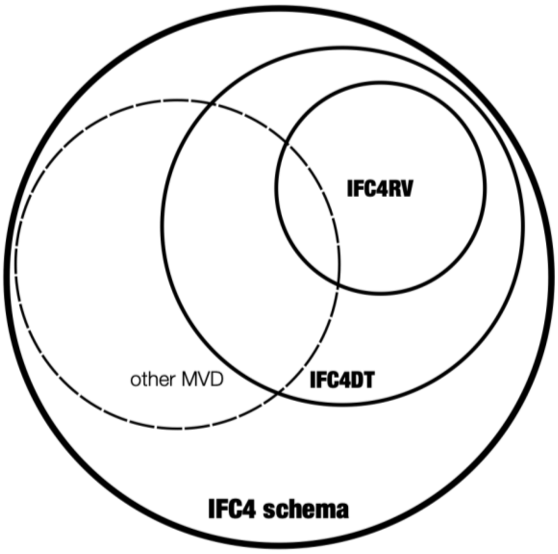
2. For project design coordination purpose, .ifc format is recommended.

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('ViewDefinition [ReferenceView_V1.2]','ExchangeRequirement [Architecture]','Option [Elements to export: Entire project]','Option
[Partial Structure Display: Entire Model]','Option [IFC Domain: All]','Option [Structural Function: All Elements]','Option [Convert Grid elements:
On]','Option [Convert IFC Annotations and Archicad 2D elements: Off]','Option [Convert 2D symbols of Doors and Windows: Off]','Option [Export
geometries that Participates in Collision Detection only: On]','Option [Split complex elements: Off]','Option [Material Preservation: Explode
all]','Option [IFC Model position: Survey Point and Project Origin]','Option [Curtain Wall export mode: Single Element]','Option [Railing export mode:
Single Element]','Option [Stair export mode: Single Element]','Option [Properties To Export: All properties]','Option [Space containment:
Off]','Option [Geometry to type objects: Off]','Option [Element Properties: Off]','Option [Building Material Properties: Off]','Option [Element
Parameters: Off]','Option [Component Parameters: Off]','Option [IFC Base Quantities: Off]','Option [Door Window Parameters: On]','Option [IFC Space
boundaries: Off]','Option [Archicad Zone Categories as IFC Space classification data: Off]','Option [Element Classifications: On]'),'2;1');
FILE_NAME('D:\\UsersProfile\\ngcy3\\Desktop\\8676-ADA-Z01-ZZ_AL-AR-M3_W.ifc','2024-05-09T09:50:30',('Architect'),('Building Designer Office'),'DDS_IFC
v3.0','IFC file generated by Graphisoft Archicad 26.0.0 INT FULL Windows version (IFC add-on version: 5002 INT FULL).','The authorising person');
FILE_SCHEMA(('IFC4'));
ENDSEC;
DATA;
#1=IFCPROJECT('36MmSVM9g9ACMByvffpBZx',#2,'ACTIVITY CENTRE FOR THE PROMOTION OF CHINESE HISTORY AND CULTURE',$,$,$,$,(#3),#4);
#2=IFCOWNERHISTORY(#6,#7,$,.NOCHANGE.,$,$,$,1715219394);
#3=IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,0.01,#8,#9);
#4=IFCUNITASSIGNMENT((#13,#14,#15,#16,#17,#18,#19,#20,#21,#22,#23,#24,#25,#26));
#5=IFCRELAGGREGATES('11UKMAmViwOgDAy4EyAQZE',#2,$,$,#1,(#27));
#6=IFCPERSONANDORGANIZATION(#28,#29,$);
#7=IFCAPPLICATION(#30,'26.0.0','Archicad','IFC add-on version: 5002 INT FULL');
#8=IFCAXIS2PLACEMENT3D(#31,#32,#33);
#9=IFCDIRECTION((0.,1.));
#12=IFCGEOMETRICREPRESENTATIONSUBCONTEXT('FootPrint','Model',*,*,*,*,#3,$,.MODEL_VIEW.,$);
```

Figure 8:Example of an IFC in STEP format, opened in common text editor

3.13. Model View Definition (MVD) and Software Certifications

1. In BIM workflows, users need to define and exchange specific IFC data subsets for use cases like coordination or handover, sharing only relevant information to reduce file complexity and maintain interoperability.



2. Model View Definition (MVD) provides a standardized subset of the IFC schema, specifying required data exchanges for scenarios while ensuring consistency across tools under buildingSMART governance to avoid fragmentation.

3. An MVD defines included data (e.g., entities, properties, relationships, geometry), exclusions, and applications of concepts like classification or material assignment, expressed via mvdXML (phasing out due to limitations).

4. Official MVDs include

   a. IFC4 Reference View V1.2 (RV1.2), released in 2018, supports robust geometry and properties for detailed reference models.

   b. Design Transfer View (still in progress), aim for editable handovers,

   c. IFC2x3 Coordination View V2.0 suits basic coordination, and

   d. IFC4.3 adds views for infrastructure (e.g., bridges).

   MVD Database at https://technical.buildingsmart.org/standards/ifc/mvd/mvd-database/ lists official and industry MVDs, though proliferation risks creating inconsistent mini-standards.

MVD Database

buildingSMART International official Model View Definitions (MVDs) releases are listed here. This includes exchange specifications for IFC2x3 and IFC4, as well as those that have been accepted as international standards or are currently under development via the buildingSMART Standards Process. Software applications may support sending or receiving data supporting any combination of exchanges.

All MVDs listed below are © buildingSMART International Ltd. Written requests should be addressed to: buildingSMART International Limited, Kings House, Station Road, Kings Langley, Hertfordshire, WD4 8LZ, UK. Email contact@buildingsmart.org.

IFC Schema	MVD Name	Status	Documentation	Summary
IFC4 ADD2 TC1	IFC4Precast	Final	Full documentation (zip)	Exchange of geometric information between CAD and MES systems for automated production of precast building components.
IFC4.2	Bridge Construction View	Draft	BRie 2017.10.24	Build and maintain bridges.
IFC4 ADD2 TC1	Reference View	Final	RV 1.2 HTML RV_1-2.mvdxml	Simplified geometric and relational representation of spatial and physical components to reference model information for design coordination between architectural, structural, and building services (MEP) domains
IFC4 ADD2 TC1	Design Transfer View	Draft	DTV 1.1	Advanced geometric and relational representation of spatial and physical components to enable the transfer of model information from one tool to another. Not a "round-trip" transfer, but a higher fidelity one-way transfer of data and responsibility.
IFC4 ADD2 TC1	Quantity Takeoff View	Draft	mvdXML	Estimate and track construction materials and costs.
IFC4 ADD2 TC1	Energy Analysis View	Draft	EV	Estimate and track energy usage and costs.
IFC4 ADD2 TC1	Product Library View	Draft	LV 0.1	Manufacturer product information and configurations.
IFC4	LandXML view		LandXML MVD 1.2	Basic buildingSMART MVD for LandXML v1.2.
IFC2x3 TC1	Coordination View	Final	CV 2.0	Spatial and physical components for design coordination between architectural, structural, and building services (MEP) domains
IFC2x3 TC1	Space Boundary Addon View	Final	SB 1.1	Identification and export of additional Space Boundaries (polygons which define the extents of a space's contact with directly adjacent surfaces [e.g. walls, floors, ceilings] and openings). Can be used for building energy analysis and quantity take-off.
IFC2x3 TC1	Basic FM Handover View	Final	Diagrams – Documentation	Handover of model information from planning and design applications to CAFM and CMMS applications, as well as the handover of model information from construction and commissioning software to CAFM and CMMS applications
IFC2x3 TC1	Structural Analysis View	Final	SA	The structural analysis model, created in a structural design application by a structural engineer to one or many structural analysis applications.
IFC 2×3	Architectural Design to Building Energy Analysis	Candidate	link	This is not a formal bSI MVD

A full list of official MVD releases by the buildingSMART

5. Developers may develop or users may select MVDs from the database, integrate them into software for IFC export/import (e.g., via certification), and validate models using tools like the Validation Service, combining with IDS for custom requirements in workflows like facility handover.

6. For example, the IFC4 Reference View V1.2 MVD supports detailed reference models, exporting precise info (e.g., an IfcWall's shape and material) for architect-engineer review, excluding extras like scheduling data.

7. It should be noted that there is no MVD that contains every data the IFC4 schema can hold.

8. The BIM Harmonisation Guidelines for Works Department stated in paragraph 5.3.4 Mandatory Requirements for BIM modelling, that:

   h) When preparing IFC for submission to the GBDR, Reference View v1.2 in ArchiCAD, OpenBuildings Designer, and Revit; or Coordination View v2.0 in Civil 3D should be set as Model View Definitions.

9. When MVD is not specified in EIR or BEP, IFC4RV should be used since:

   a. It is the latest IFC version

   b. Certified for export and import in most commonly used BIM authoring tools (explained in following section)

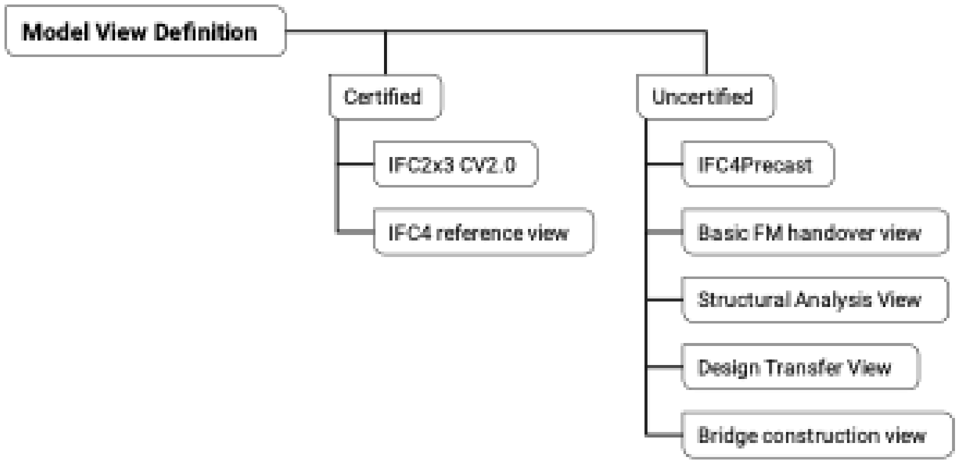
   c. Incorporates geometries defined by BREP (Boundary Representation), which is non-parametric 3D shapes, that can be consistently reproduced in most BIM tools without mis-interpretation.

10. The major difference between the IFC4 Reference View and IFC4 Design Transfer View Model View Definitions (MVDs) lies in their purpose, geometric complexity, and intended use in building information modeling (BIM) workflows. Below is a concise summary of the key distinctions:

Aspect	Reference View	Design Transfer View
Purpose	Designed for coordination and interoperability. Used for sharing a stable, simplified model for reference, analysis, or visualization (e.g., clash detection, quantity takeoff).	Designed for design and editing. Enables detailed, editable models for modification and further development in BIM software.
Geometric Complexity	Supports simplified geometry (e.g., faceted B-Rep, basic swept solids, limited CSG). Avoids complex or software-specific representations to ensure broad compatibility.	Supports complex geometry (e.g., advanced B-Rep with topology, curve-based geometry, complex CSG, custom profiles). Allows richer, detailed modeling.
Editability	Read-only or minimally editable. Geometry is static, meant for viewing or reference, not modification.	Fully editable. Geometry and properties can be modified, supporting design iterations and detailed workflows.
Use Case	Used in downstream processes like coordination, simulation, or facility management. Common for cross-disciplinary sharing (e.g., architects to contractors).	Used in upstream design phases where models are actively created or modified (e.g., architectural or structural design).
Data Scope	Limited to essential data for coordination. Excludes advanced parametric relationships or detailed topological data.	Includes comprehensive data, including parametric relationships, complex constraints, and detailed topology for advanced modeling.
Interoperability	High interoperability due to simplified geometry and standardized representations. Compatible with most BIM tools for viewing.	Lower interoperability due to complex, software-specific geometry. May require specific tools to fully utilize editable features.

File Size	Typically smaller due to simplified geometry and reduced data complexity.	Larger due to detailed geometry, parametric data, and advanced representations.

3.13.2. Certifications



1. buildingSMART's certification program aims to ensure consistent and reliable implementations of IFC standards across software vendors—like Revit, Tekla, or ArchiCAD—in the global market.

2. It focuses on validating both the export and import capabilities of software against specific MVDs, historically supporting IFC2x3 Coordination View V2.0 (CV2.0) and IFC4 Reference View V1.2 (RV1.2).

3. The process involves software vendors registering and submitting their tools for testing through the validate.buildingsmart.org validation service (or GitHub test files for imports), where IFC files are checked for accuracy against schemas and MVDs, with results leading to scorecards or official certification upon review by buildingSMART.

4. As of 2025, over 50 vendors have successfully certified/validated their software, including popular tools like ArchiCAD (by GRAPHISOFT), Revit (by Autodesk), and Tekla (by Trimble), with a full list available at technical.buildingsmart.org/services/certification/ifc-certification-participants/.

IFC Certification Participants

For the market to benefit the most from IFC, there must be a robust implementation in software available to users in their respective regions and marketplaces. buildingSMART International provides an ongoing platform and process to certify applications for IFC2x3 Coordination View 2.0 and IFC4 Reference View 1.2.

50 **entries per page** **Search:** ifc4

Vendor	Product	Schema	Exchange Requirement	Import / Export	Status	Started	Completed	Report (link)
Allplan GmbH	Allplan	IFC4	Architectural Reference Exchange	Import	In Progress	13/03/2020		
Allplan GmbH	Allplan	IFC4	Architectural Reference Exchange	Export	Finished	13/03/2020	06/05/2022	
GRAPHISOFT SE	ARCHICAD	IFC4	Architectural Reference Exchange	Import	Finished	21/06/2018	31/05/2023	
GRAPHISOFT SE	ARCHICAD	IFC4	Architectural Reference Exchange	Export	Finished	21/06/2018	05/06/2019	
Autodesk	Autodesk Revit	IFC4	Architectural Reference Exchange	Import	Finished	29/08/2017	14/09/2022	
Autodesk	Autodesk Revit	IFC4	Architectural Reference Exchange	Export	Finished	29/08/2017	19/10/2020	

Extract from the full list of certified BIM tools on official website

"Certified" means the software has undergone buildingSMART's validation process and achieved passing results (e.g., no red error indicators in scorecards), confirming it can reliably produce or consume IFC files that conform to specified MVDs without structural, semantic, or geometric inaccuracies. For example, Autodesk Revit is certified for IFC4 Reference View V1.2 export, meaning it can export BIM models (e.g., architectural elements like walls with properties and geometry) as IFC files that accurately adhere to the RV1.2 subset, enabling seamless import into tools like Solibri for clash detection without data loss, as verified through buildingSMART's test cases and anonymous user metrics.

4. Common Data Environment

This section outlines the selection of a Common Data Environment (CDE) for an OpenBIM project.

While many CDEs aim to provide comprehensive functionality, some features may be poorly implemented. Teams should be open to adopting complementary software or platforms to address specific functional deficiencies

4.1. Functional Requirement

1. According to "*CIC Beginner's Guide on Construction Digitalisation -Adoption of Common Data Environment (CDE) for Information Management using BIM*", the 3 essential functions of a CDE are:

   a. EDMS (Electronic Documentation Management System)

   b. Workflow Management

   c. 2D & 3D Coordination

2. The functional requirements for a CDE should remain consistent for both OpenBIM and software-specific BIM projects, with the only difference being that the CDE and its model-related functions should support IFC models (IFC2x3, IFC4, and the latest versions, in different Model View Definitions (MVD)).

4.2. Robust IFC Model Viewer

Robust and responsive IFC model viewer in the CDE is crucial for effective model coordination in OpenBIM projects. It should meet the following criteria:

1. Responsiveness

   a. Process IFC models efficiently without relying on real-time calculations of stored parametric definitions, prioritizing pre-computed geometry for responsiveness.

2. Performance Optimization

   a. Able to handle large number of models and huge amount of polygons without costing heavily performance and responsiveness. Optimize performance through polygon count management and intelligent culling, dynamically reducing model detail to ensure fast loading and navigation of federated IFC models while maintaining visual fidelity.

3. Object Organization and Visualization

   a. Organize objects by IFC spatial structure (e.g., IfcSite, IfcBuilding, IfcBuildingStorey, IfcSpace), provide customizable object coloring for enhanced visual clarity, and support object filtering to isolate specific elements or IFC Classes (e.g., IfcWall, IfcDoor).

4. Part Model Display

   a. Allow view sections and model element filtering by selected criteria, to allow users to view their selected part of the model.

5. Flexible Model Viewing

   a. Allow users to freely toggle BIM models (e.g., by state like WIP or Shared, or by version) without restrictions for flexible federation and viewing.

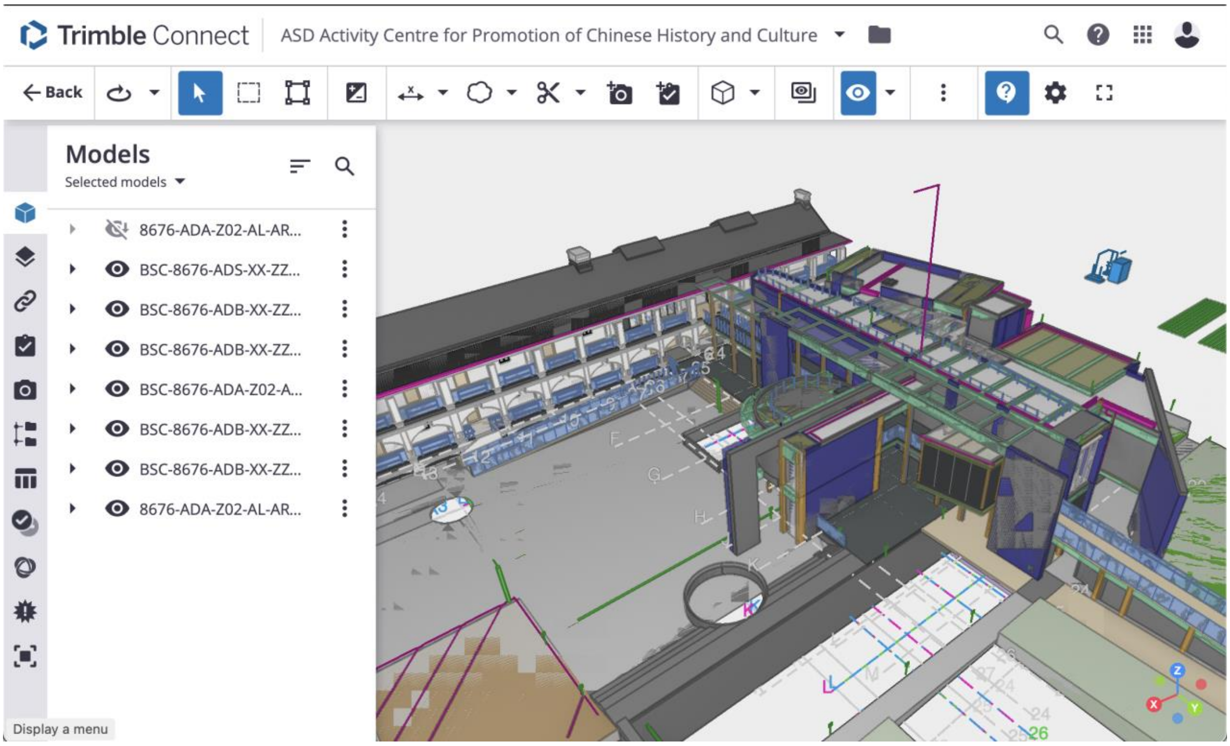


Figure 9: Example from Pilot Project - individual model can be toggle on or off from view in Trimble Connect for different viewing purposes

6. View Saving

   a. Enable users to save private or public views with annotations for specific purposes, such as checking discrepancies between architectural and structural models or comparing approved drawings with the latest design model. The following frequently used views may be pre-set and maintained by BIM team leaders or relevant stakeholders to be readily used during coordination:

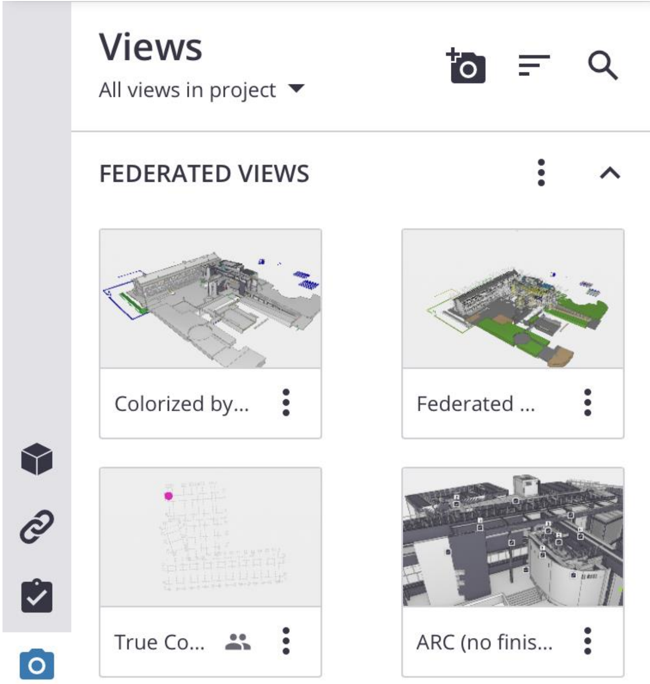


Figure 10: Example from Pilot Project - Pre-set public views (federated views, architectural only view, grid and coordinates setting view, etc.) were saved in Trimble Connect for easy retrieval by whole team

Example: Trimble Connect ?

(i) For architectural core comparison against structural models. To verify discrepancy such as levels, vertical structural member sizes and positions, structural openings for doors and windows, etc.

(ii) Combined building services with architectural smoke vents, against others.

(iii) General federated models views fully textured and colored, or colored for identification purpose, as may be required by EIR or BEP.

(iv) For headroom checking, IfcSpaces against structures and combined building services.

(v) General federated models using latest shared models, or models at various milestones like major presentation and tender, etc.

IfcSite origin point and grids

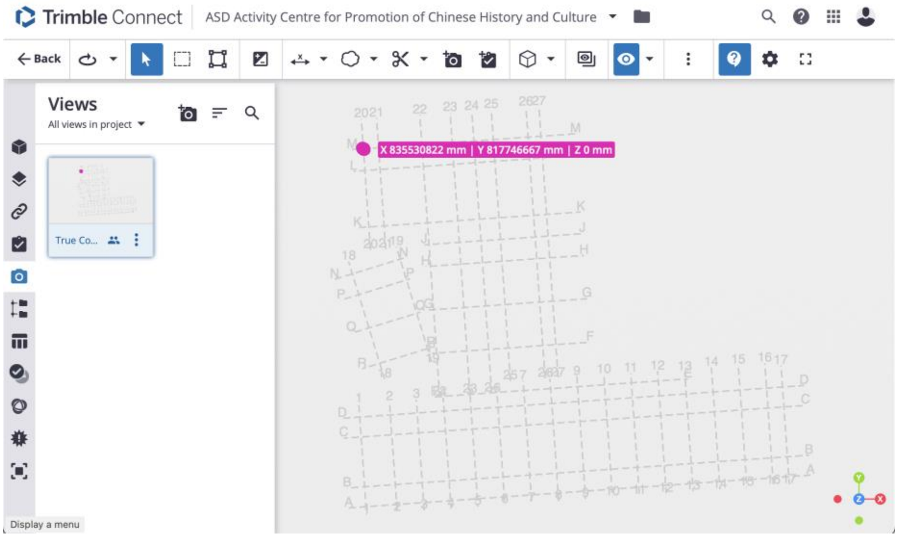


Figure 11: Example from Pilot Project - view saved showing the positions of IfcSite origin and building grid, to facilitate model position alignment checking

Example: Trimble Connect

7. Property Handling

   a. Access most (if not all) IFC properties (e.g., Psets, Qsets), non-IFC properties, and relationship properties (e.g., IfcRelContainedInSpatialStructure) to deliver comprehensive model data visibility.

8. Active Development

   a. Undergo active development to support evolving IFC standards (e.g., IFC4.3 and future versions) and incorporate user feedback for continuous functional improvements.

4.3. 2D & 3D Coordination Functions

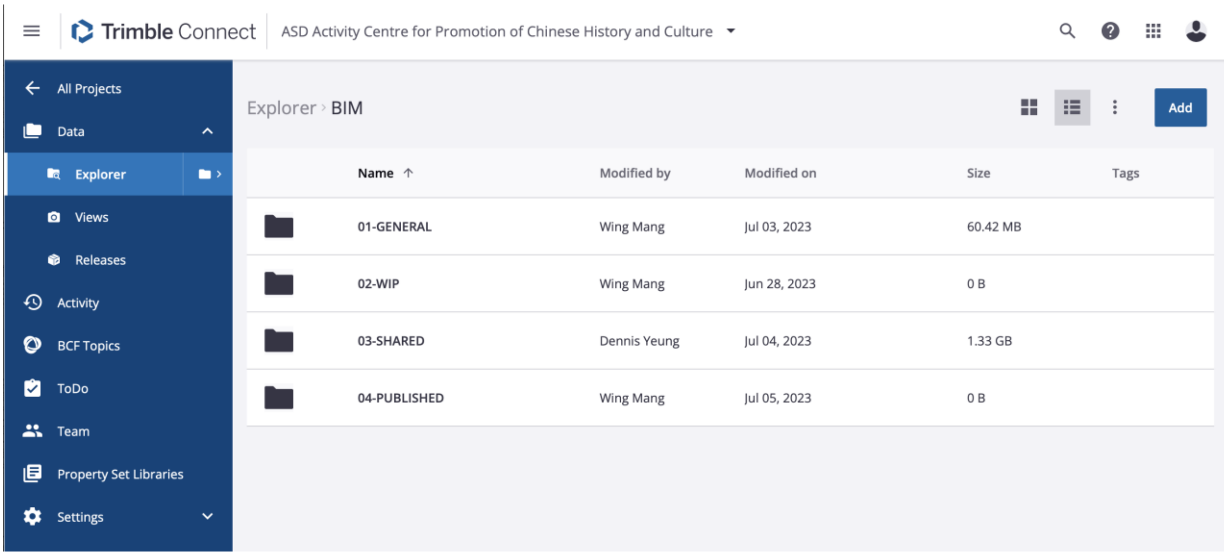
The CDE should have the following functions for effective 2D & 3D Coordination:

1. Data Repository with versioning

   a. Serve as a centralized repository for BIM files, including IFC models and related documents. All uploaded versions of BIM models should be automatically saved and readily retrievable.

2. Information States

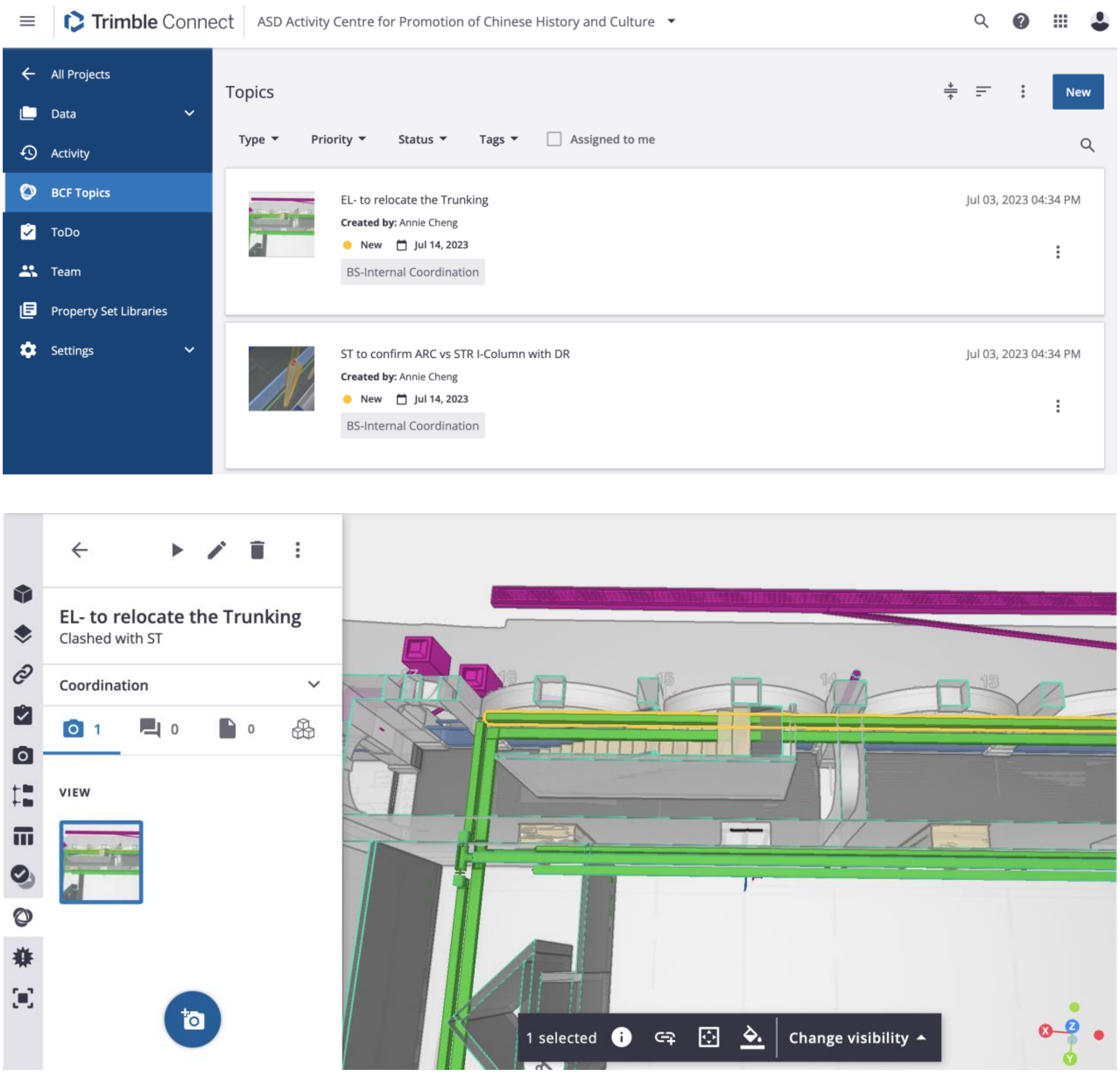
   a. Implement ISO 19650-1 information states (e.g., Work-in-Progress, Shared, Published, Archived), usually by using dedicated folders to organize BIM files by state.



Example: Trimble Connect

3. BCF Issue Tracking

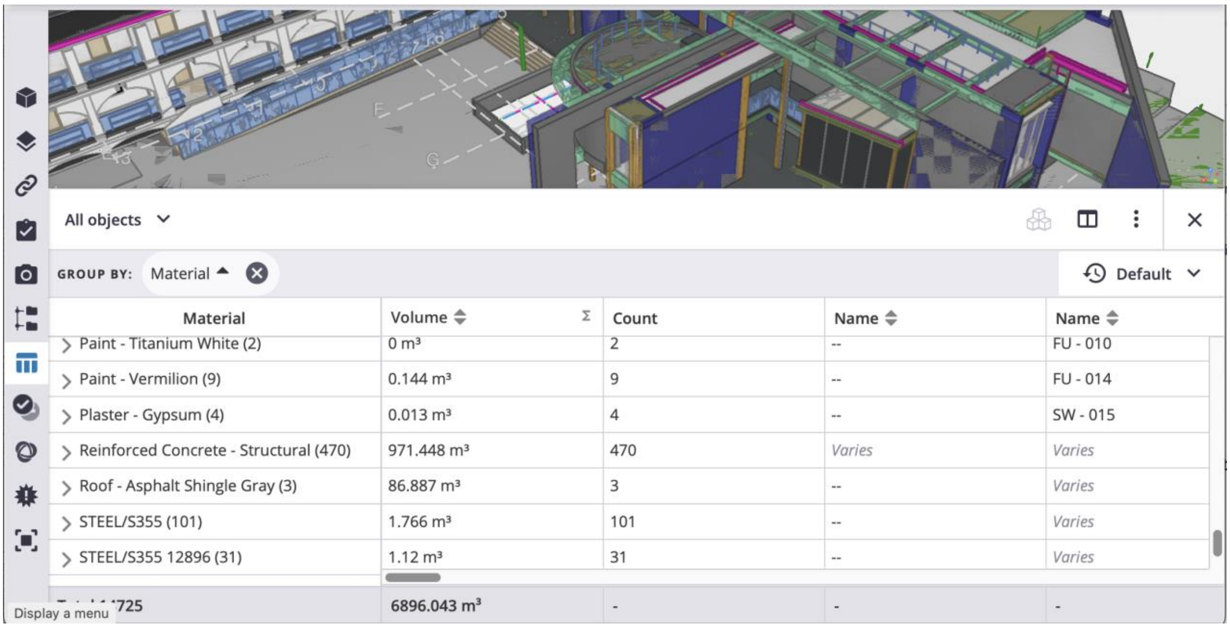
a. Provide a centralized BCF server to create, view, modify, and comment on issues, automatically locate issues in the model viewer, and support BCF import/export for interoperability with other BCF-compatible software



Example: BCF in Trimble Connect

4. Quantity Takeoff

   a. Support basic quantity takeoff from IFC models, such as element counting, element scheduling with properties complemented with model filter by schedule.



Material	Volume	Count	Name	Name
Paint - Titanium White (2)	0 m³	2	--	FU - 010
Paint - Vermilion (9)	0.144 m³	9	--	FU - 014
Plaster - Gypsum (4)	0.013 m³	4	--	SW - 015
Reinforced Concrete - Structural (470)	971.448 m³	470	*Varies*	*Varies*
Roof - Asphalt Shingle Gray (3)	86.887 m³	3	--	*Varies*
STEEL/S355 (101)	1.766 m³	101	--	*Varies*
STEEL/S355 12896 (31)	1.12 m³	31	--	*Varies*
...725	6896.043 m³	-	-	-

Example: Basic function of Quantity Takeoff in Trimble Connect

4.4. Scalability

1. While functional capabilities are essential, they represent only part of a CDE's effectiveness. Scalability—reflecting the CDE's maturity, reliability, and performance—is critical for managing increasing volumes of IFC files.

2. Scalability refers to the platform's ability to maintain efficient performance, including fast model viewing, data extraction, and system stability, as the number of files, users, and data complexity grows, ensuring reliable collaboration across large or complex projects.

3. Poor scalability can render claimed functional capabilities (e.g., model viewing, 3D coordination, issue tracking, etc.) unusable, as performance degradation may prevent effective access to or use of these functions. In such cases, these functions should be deemed not provided, and the project team member responsible for proposing or procuring the CDE should be required and allowed to propose complementary software or platforms to address these deficiencies.

4. The BIM Team Leader should anticipate the approximate number of models the project will handle throughout its lifecycle to guide scalability testing.

5. A CDE should address these scalability criteria:

   a. Viewing Performance

(i) Enable fast loading and navigation of IFC models (e.g., combined building services models across multiple floors) as file numbers increase, showcasing a mature platform.

b. Data Extraction

(i) Support efficient extraction of IFC data (e.g., Psets, Qsets) for reports, even with large datasets, demonstrating reliability.

c. Performance Testing

(i) Test scalability by uploading increasing IFC files, based on the BIM Team Leader's anticipated model count, to verify viewing, data extraction, other claimed functions (e.g., coordination, EDMS, issue tracking), and stability under multi-user access, assessing overall performance.

d. System Stability

(i) Maintain reliability under load from multiple users or large datasets, indicating a robust platform.

EXPORTING IFC

5.1. One Size Fit all or Tailored for Purpose?

1. There is no one-size-fit-all export setting or exported IFC model which can meet the required information for all purposes. Therefore, before determining what information a IFC model should contain and how they should be translated from native model, it is important to identify the purpose of the IFC on the receiving end. For example:

    a. Only facade, external wall elements and IfcSpace with space boundary would be necessary for engineering analysis for environmental energy analysis (model filter)

    b. For early design coordination purpose, detail geometry export down to centimeter precision might not be necessary, but would be necessary for structural detailing purpose, and down to millimeter precision with true curve for fabrication purpose. (LOD-G and geometry precision)

    c. Although IFC4DT is designed to store parametric geometry definition, IFC4DT is still not covered under buildingSMART's software certification scheme. It would be safe to assume the parametric definition in IFC4DT models would not be readily extracted by model receiving parties. Also, the native BIM might not contain the parametric definitions of complex objects in the first place, e.g. complex curtain wall geometry. Therefore, for transferring parametric definition of such complex geometries, origin definition files should be transferred such as rhinoceros grasshopper files.

    d. Properties related to future building maintenance would usually only be needed in as-built and asset information model. (LOD-I)

2. The following sections focus on IFC export for design coordination purpose.

5.2. Exchange Format and Model View Definition

1. It should be noted that each MVD of IFC contains different set of BIM data, and no single MVD is design to contain all data that can be exported from native BIM model. Therefore, it is crucial to always have the end goal in mind before deciding which IFC MVD for export.

2. As a minimum, IFC4RV (IFC 4 Reference View) should be used for model exchange among different disciplines and softwares, except for special occasions and uses.

3. EIR / BEP should state the exchange format requirements and purpose. The following table summarises a sample of which IFC schema should be imported and exported for different softwares and purposes:

		AR / LA (Archicad)	ST (Tekla)	BS (Revit)
		exports:		
AR / LA (Archicad)	imports:	-	- IFC4DT (Archicad might convert ST elements to native Archicad elements)	- IFC4RV
ST (Tekla)		- IFC4RV	-	- IFC4RV
BS (Revit)		- IFC4RV - IFC4RV containing IfcSpace only, for Revit conversion to native Room element - DWG (as background for drawings production in BS drawings)	- IFC4RV	-

4. Upon completion of any milestone, to maximise compatibility with future receiving software applications, the following formats of the BIM models should be shared:

   a. Native models

   b. Three IFC file schema:

      (i) IFC2x3 Coordination View 2.0 (IFC2x3CV2),

      (ii) IFC4 Reference View (IFC4RV), and

      (iii) IFC4 Design Transfer (IFC4DT).

   c. PDF from drawing production[6]

   d. DWG from drawing production[7]

   e. Any other supporting or linked files, such as xlsx, doc, etc.

5. Architectural discipline should share 2D architectural layout plan (in dwg preferably, or PDF in vector) for building services & others (where necessary) as underlay for drawing production, since:

   a. Building Services drawings usually have architectural layout underlay in their drawings.

   b. However, most BIM authoring software cannot generate architectural layout plan that meets the usually required drawings standard from the shared architectural IFC model.

   c. The architectural IFC model aligning with the 2D layout plan should be shared at the same time.

6. Tool for converting IfcSpace into native room may be needed for some MEP authoring tool:

   a. In MEP BIM authoring softwares, room usually is an important location information for onward building services analysis function.

   b. For example, in Revit MEP, add-on would be necessary to convert imported IfcSpace into native Room, which would be subsequently natively convert to Space and combined as Zone for MEP analysis.

[6] For MEP drawing production, architectural layout is required. As imported IFC may no be able to produce architectural background in required drawing standards, architectural layout in pdf or dwg might be necessary for drawing underlay.

[7] DWG files should be exported for use by non-BIM softwares

5.3. Coordinates

5.3.1. Round Tripping Principle

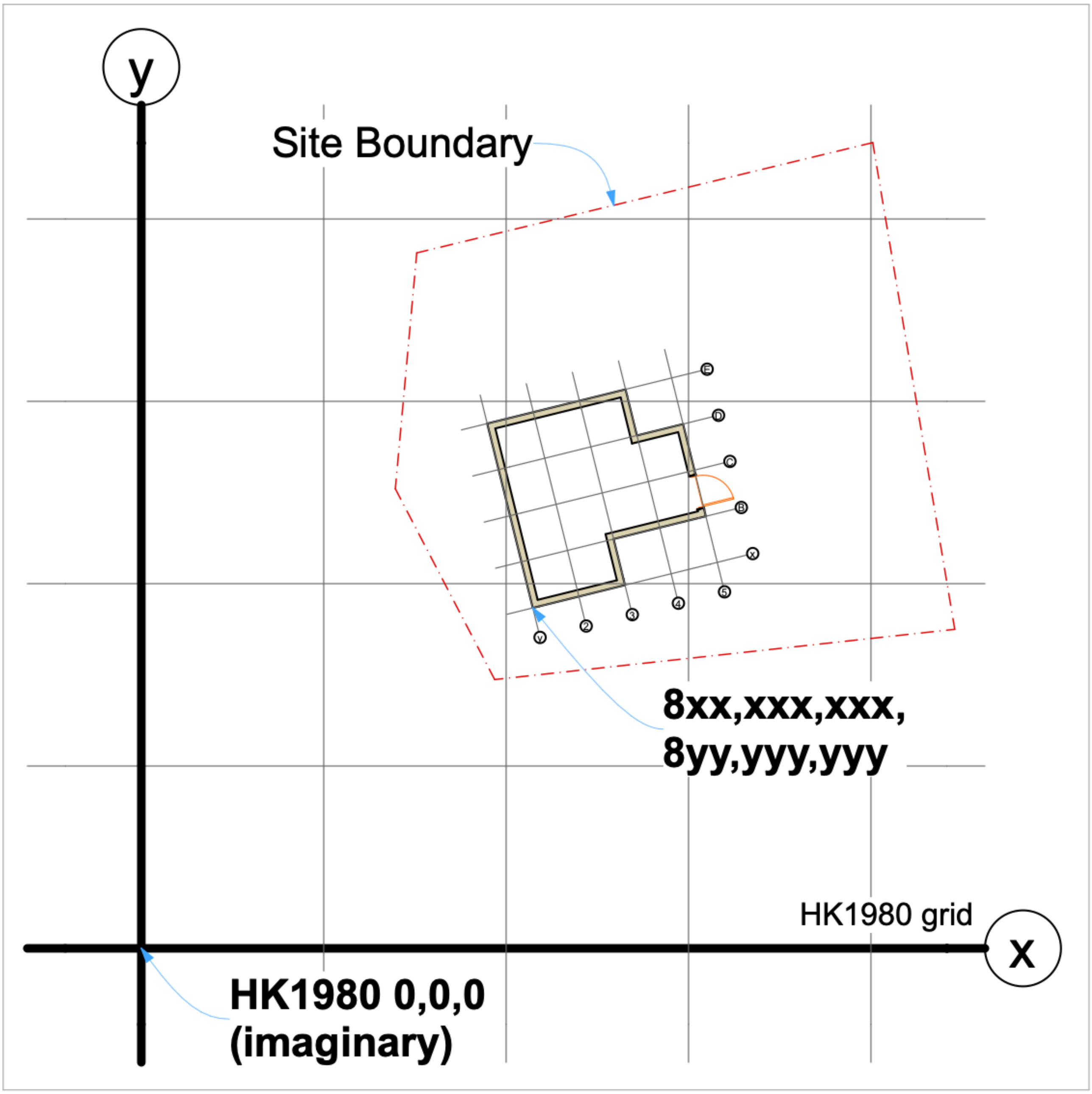
1. Since IFC is used for major model exchanges, all coordinates setup should be capable to do “round trip” with this IFC setup. That is, all native model files in their respective BIM authoring softwares should be so set up that:

    a. importing IFC files with these coordinates setup should match the project locations in the native files, and

    b. exporting IFC files should yield same coordinates setup.

2. As a general rule, in BIM authoring softwares. IFC models should be imported and exported based on the IfcSite, instead of IfcProject.

5.3.2. Typical Site Coordinates of Building Projects in Hong Kong



HK1980 Grid

The majority of building projects in Hong Kong adhere to a standardized coordinate system, as depicted above. This system is characterized by the following:

1. Located at 8xx,xxx,xxx, 8yy,yyy,yyy on the HK1980 Grid system,

2. Building grid at an angle to true north.

5.3.3. Coordinates Mapping between HK1980 and IFC

1. The coordinates systems of IFC models and the HK1980 Grid System are two separate and independent grid systems.

2. The following mapping should be adopted:

IfcProject in IFC model	HK1980 Grid System
IfcProject at 0,0,0	0,0,0 (imaginary location outside Hong Kong)
X Axis	Easting
Y Axis	Northing
Z Axis at 0	0 mPD (Hong Kong Principle datum)

3. However, due to software limitations, the building should always be drawn:

    a. near the software's origin of its internal coordinates,

    b. Orthogonally, as much as practicable, to the building grid.

4. Therefore, another secondary coordinate system is used in both IFC and BIM authoring software. The mapping between them are as follow:

IFC	Revit	Archicad	Tekla
IfcSite 0,0,0	Internal Origin	Project Origin	Global Point Of Origin

5. IfcSite is located relative to IfcProject grid system, at a distance and angle.

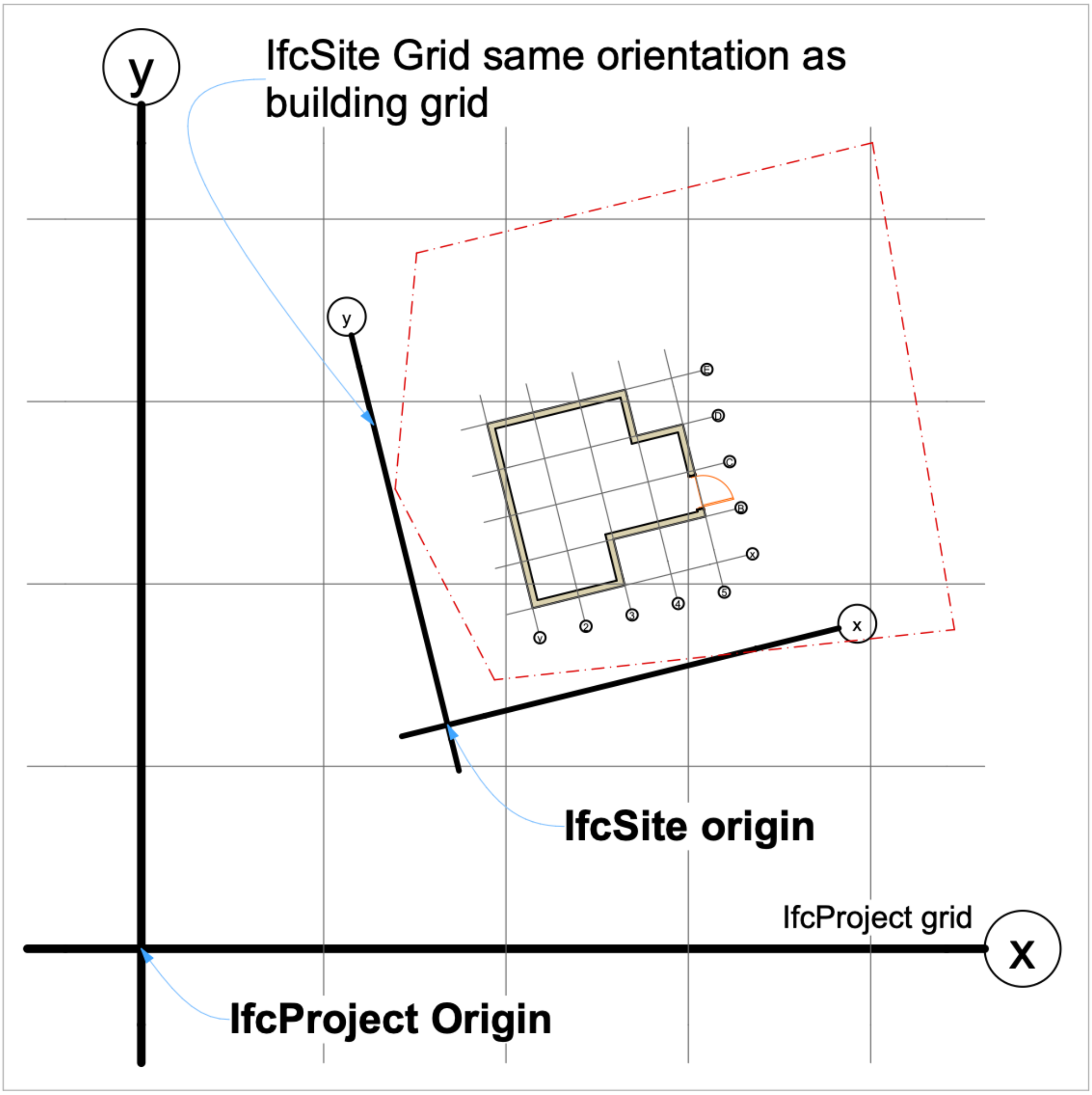
6. Example of IfcSite's placement relative to IfcProject coordinates is as below:

```
ObjectPlacement=IFCLOCALPLACEMENT($,RelativePlacement);
RelativePlacement =IFCAXIS2PLACEMENT3D(Location,Axis,RefDirection);
Location =IFCCARTESIANPOINT((835530822.,817746667.,0.));
Axis=IFCDIRECTION((0.,0.,1.));
RefDirection=IFCDIRECTION((0.9974964096816783,0.0707171313909262632,0.));
```

    a. The example shows IfcSite is rotated at 4.0552 degree (angle a) counter-clockwise relative to IfcProject (calculated by tan(a) = 0.0707/0.9975, derived from the RefDirection value representing the x-axis vector of IfcSite in IfcProject coordinate), and

    b. is placed (835530822,817746667,0) away from IfcProject (0,0,0).

5.3.4. IFC



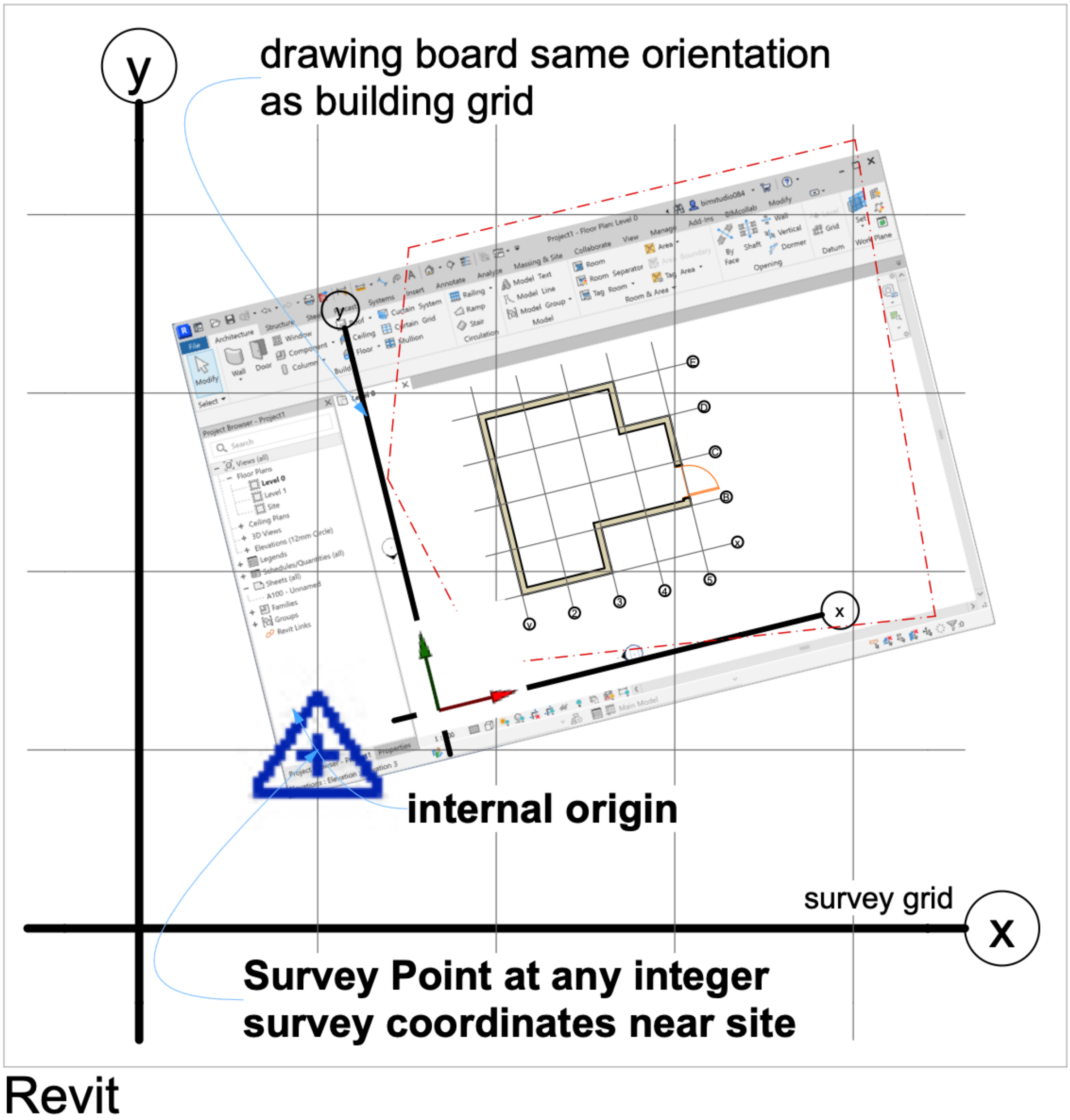
IFC

1. In IFC models, IfcProject coordinates should represent the HK1980 grid coordinates.

2. IfcSite represents a coordinates system placed close to and at the same orientation to the project building. Its origin point might be arbitrary located, but it should be at an integer coordinate point of IfcProject coordinate.

3. The site boundary and building grid do not need to be exactly placed on the IfcSite (0,0,0), since it is not uncommon that they would be changed over the course of the design phase of the project.

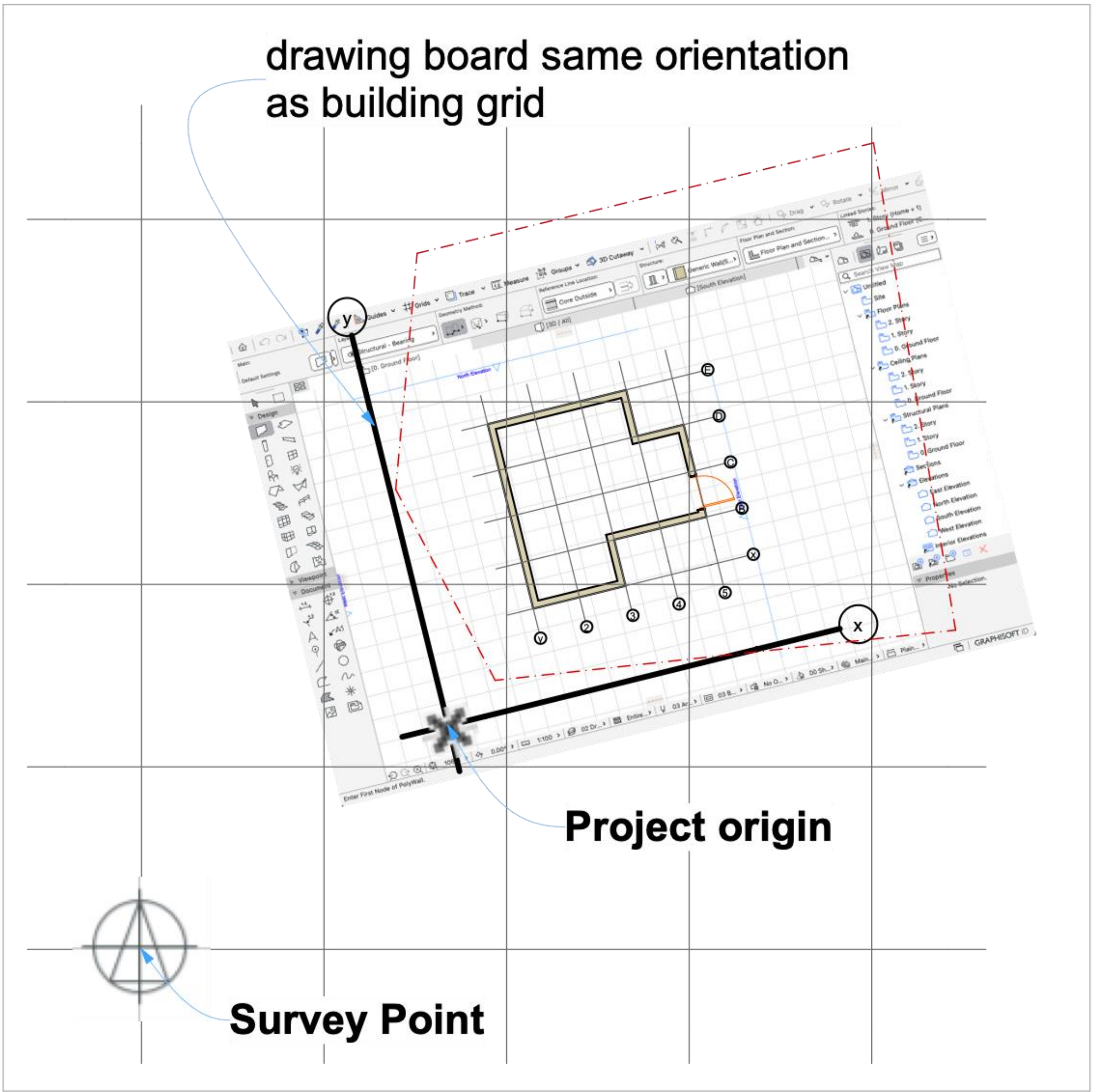
5.3.5. Revit



1. In Revit, the grid system of the drawing board, (i.e. internal coordinate system), with its origin called Internal Origin, should represent the IfcSite coordinate system.

2. The Survey Point object should be moved and rotated such that it represent an integral point of HK1980 Grid, such that any point on the drawing board would be given a HK1980 coordinates inferring from the placement of this Survey point.

5.3.6. Archicad

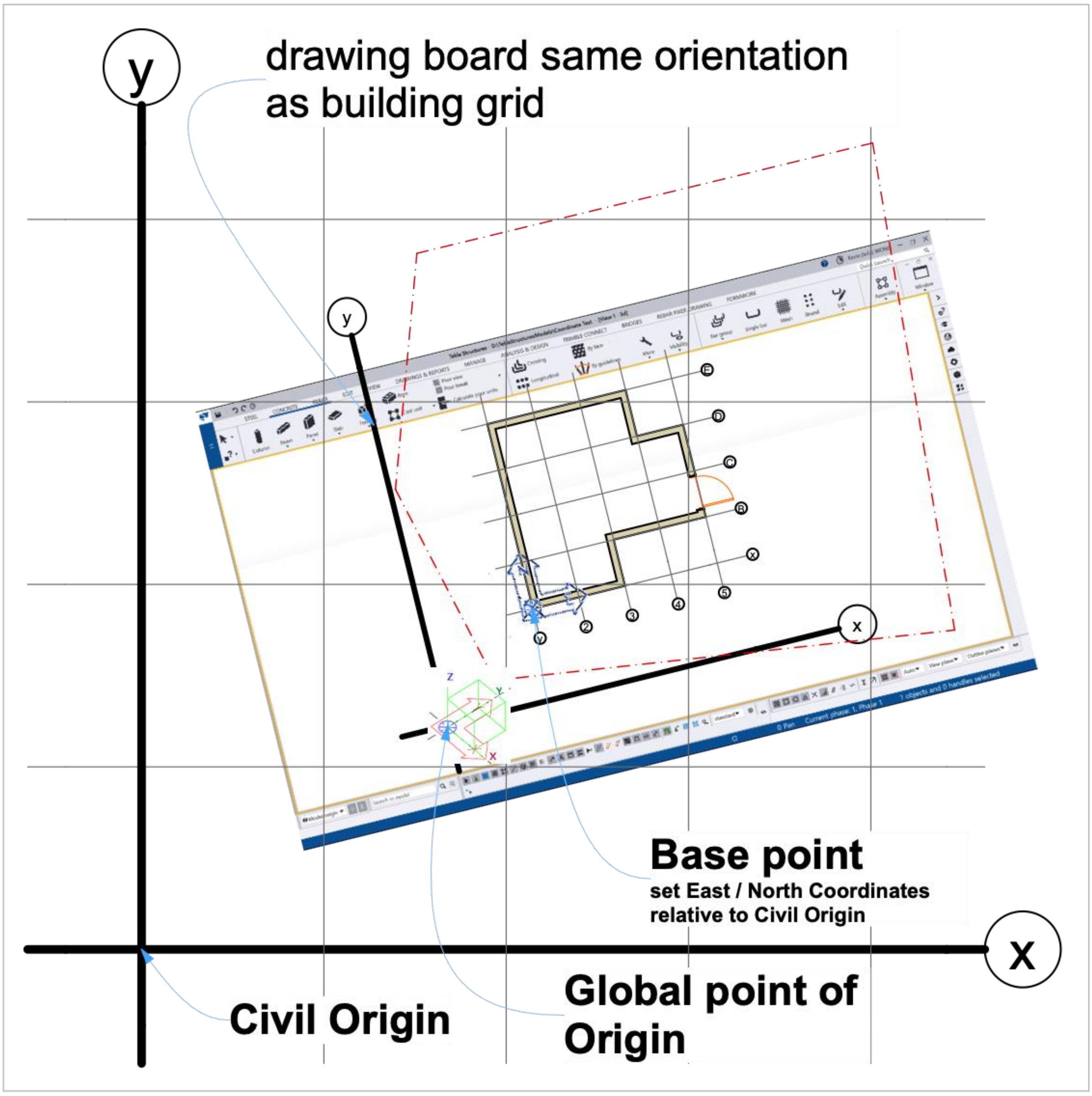


Archicad

1. In Archicad, the grid system of the drawing board, (i.e. internal coordinate system), with its origin called Project Origin, should represent the IfcSite coordinate system.

2. The Survey Point object should be moved and rotated such that it represent the (0,0,0) of HK1980 Grid, such that any point on the drawing board would be given a HK1980 coordinates inferring from the placement of this Survey point.

5.3.7. Tekla Structures



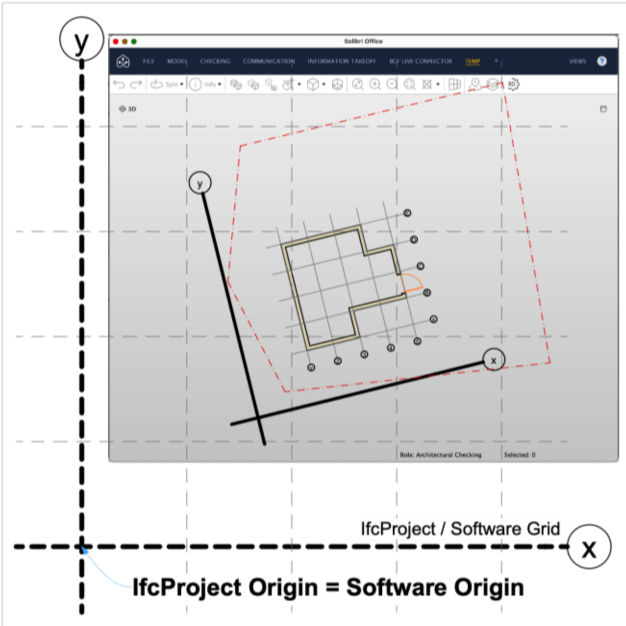
Tekla Structures

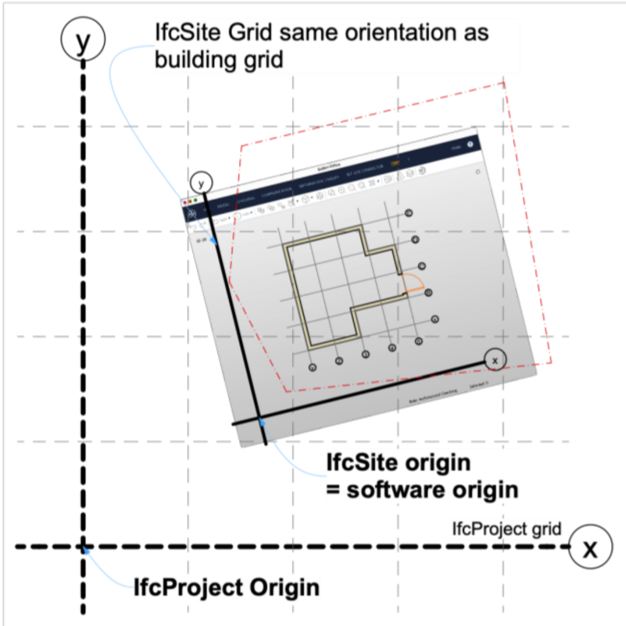
1. In Tekla Structures, the grid system of the drawing board, (i.e. internal coordinate system), with its origin called Global Point of Origin, should represent the IfcSite coordinate system.

2. A Base Point should be set up such that it represent the an integral coordinate point of HK1980 Grid, such that any point on the drawing board would be given a HK1980 coordinates inferring from the placement of this Base Point.

3. The (0,0,0) of the HK1980 Grid is called "Civil Origin" inferred from the placement of this Base Point.

5.3.8. Coordinates in Federation software





1. While most BIM authoring software can rotate the view in the authoring windows to suit the orientation of the project gird (or orientation of main project elements on plan), many model federation softwares do not provide such function. For example, Autodesk Navisworks or Solibri.

2. Many federation softwares directly open and place the IfcProject coordinates of IFC models to their software grid. As a result, many of the view functions of the softwares may not be orientated to the project grid orientation and that makes navigation of the model difficult. There functions usually includes:

   a. Section cuts

   b. Automatic annotations, etc.

   c. Model elements sometimes cannot be found since they are very far away from software origin.

3. Although it is a common local practice and software default that the project model would be opened at an angle to the software grid due to the above reasons, BIM team leader may explore whether it is worthwhile to have project model (IfcSite) aligned to the software grid for certain stage of the project like design coordination during design stage.

4. The table below shows some of the pros and cons for which the software gird aligns to:

	IfcProject	IfcSite
Coordinates	Coordinates per HK1980 grid basis, consistent for projects adopting HK1980 Grid system	Coordinates per project basis. Require coordinates transforming for federation across different projects.

	IfcProject	IfcSite
Compatibility	Many federation software can support very large coordinates	Model elements close to softwares' system coordinates origin. Highest compatibility across softwares
Navigation and readability	Models at an angle to software system coordinates. More difficult navigation Annotation may be at an angle the building layout.	Models mostly at same orientation as software system coordinates. Easier navigation and readability for annotations.
BCF Compatibility	Many BCF manager import and export camera viewpoints base relative to IfcProject coordinates.	Some BCF manager provides options for users to align the imported camera viewpoints to IfcSite instead of IfcProjec coordinates.

5.4. IFC Spatial Aggregation Hierarchy

The IfcProject, IfcSite, IfcBuilding and IfcBuildingStorey should be properly setup and align among all project models.

5.4.1. Aligning Common data fields

Although it is not well documented in the buildingSMART official documentation that whether the below data in every model of a project should be consistent, it is still a good practice that align them during model setup to prepare for future use.

The relevant fields in all exported IFC models should align to the tables in the following sub-sections. (using the Pilot Project as example)

1. IfcProject

Sample line of IfcProject from Architectural IFC4RV model:

```
#1=IFCPROJECT('36MmSVM9g9ACMByvffpBZx',#2,'ACTIVITY CENTRE FOR THE PROMOTION OF CHINESE HISTORY AND CULTURE',$,$,$,$,(#3),#4);
```

Field no.	Field	Content
1	GlobalId	36MmSVM9g9ACMByvffpBZx
3	Name	ACTIVITY CENTRE FOR THE PROMOTION OF CHINESE HISTORY AND CULTURE

For detail explanation of each field, refer https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcProject.htm

2. IfcSite

Sample line of IfcSite from Architectural IFC4RV model:

```
#27=IFCSITE('20FpTZCqJy2vhVJYtjuIce',#2,'Site',$,$,#4369,#4370,$,.ELEMENT.,(47,33,34,948800),(19,3,17,204400),0.,$,$);
```

Field no.	Field	Content
1	GlobalId	20FpTZCqJy2vhVJYtjuIce
3	Name	Site

For detail explanation of each field, refer https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcSite.htm

3. IfcBuilding

Sample line of IfcBuilding:

```
#51112=IFCBUILDING('00tMo7QcxqWdIGvc4sMN2A',#2,'Building',$,$,#51110,$,$,.ELEMENT.,$,$,$);
```

Field no.	Field	Content
1	GlobalId	00tMo7QcxqWdIGvc4sMN2A
3	Name	Building

For detail explanation of each field, refer https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcSite.htm

4. IfcBuildingStorey

The values of the following fields of building storey should be the same across all BIM models:

Below are the example lines of IfcBuildingStorey extracted from the architectural IFC4RV model:

```
IFCBUILDINGSTOREY(‘1oZ0wPs_PE8ANCPg3bIs4j',#2,'BF',$,$,#78513,$,$,.ELEMENT.,8970.);
IFCBUILDINGSTOREY('1p_MMA2In40RshTYdqmALY',#2,'GF',$,$,#78514,$,$,.ELEMENT.,12970.);
IFCBUILDINGSTOREY('01hjfK_6v9ohgHcIU5e9gU',#2,'1F',$,$,#78515,$,$,.ELEMENT.,16370.);
IFCBUILDINGSTOREY(‘3y21AUC9X4yAqzLGUny16E’,#2,'RF',$,$,#78516,$,$,.ELEMENT.,20750.);
```

Name	Elevation (in mm above principle datum)
BF	8970
GF	12970
1F	16370
RF	20750

5.4.2. Spatial Relationship Assignment

1. All BIM elements should be placed relative to correct spatial elements in relation to its actual placement, for example:

    a. A furniture inside a room should be placed in the IfcSpace, while a play equipment in open area should be placed in the IfcSite.

    b. Beams for a floor slab should be placed in the IfcBuildingStorey.

2. The IFC Project Manager in Archicad allows the users to adjust the hierarchical locations.

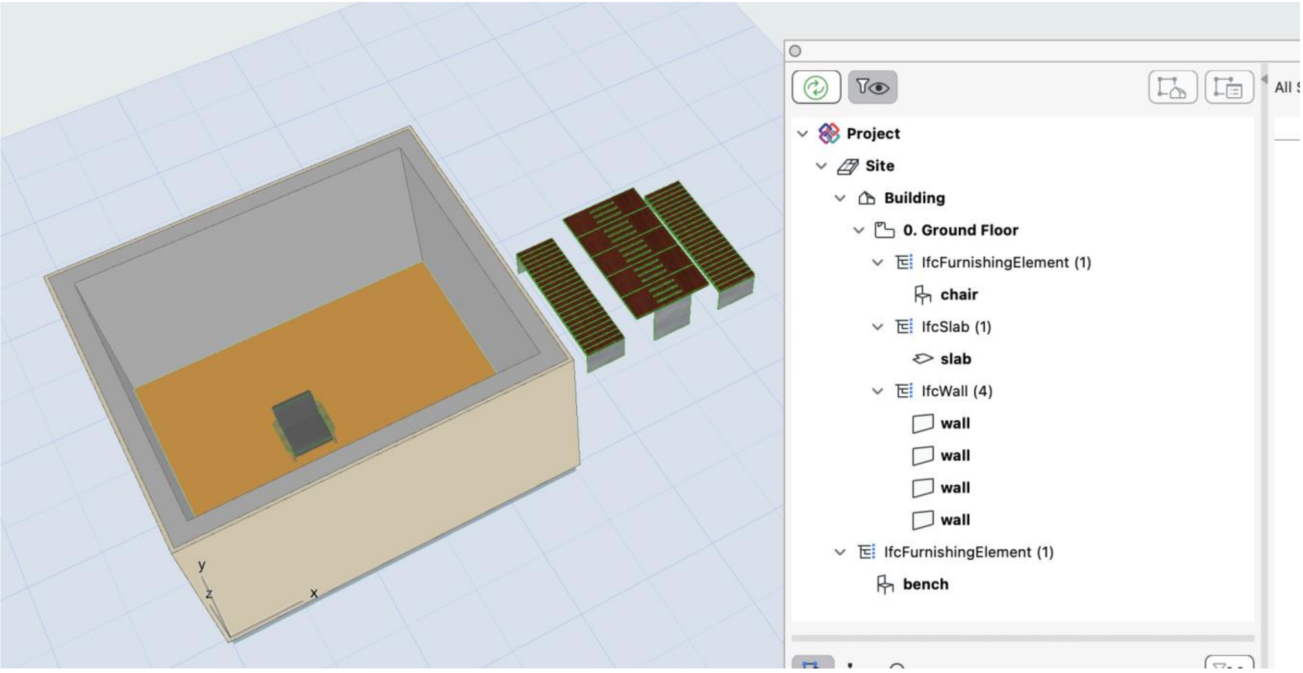
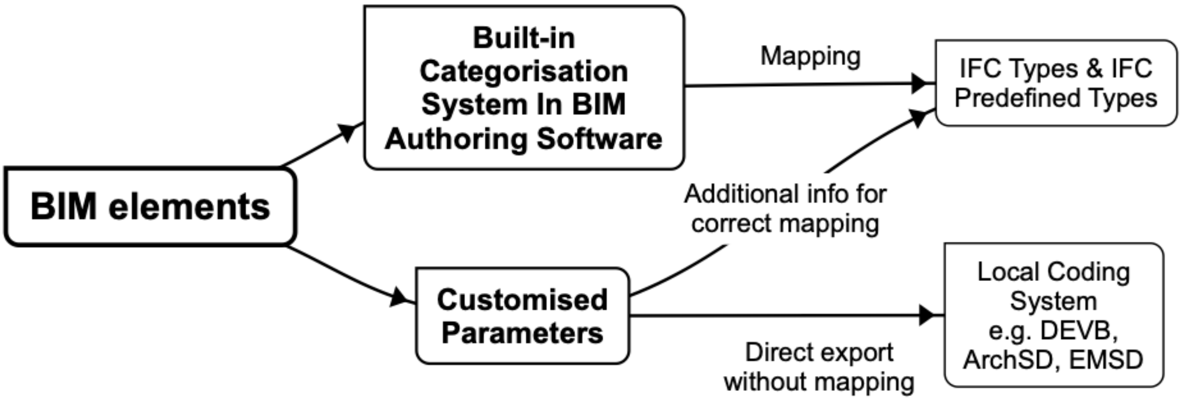


Figure 12: Archicad's IFC Project Manager allows user to change the spatial location of any elements in spatial hierarchy tree. In this example, the bench has been moved the under IfcSite, instead of under the IfcBuildingStorey (Ground Floor)

3. There are specific IFC export settings in Tekla Structures to establish the hierarchical locations in relation to IfcProject, IfcSite, IfcBuilding and IfcBuildingStorey. (Detail operation refers to the Appendix "Comprehensive Guide to OpenBIM Workflow in Tekla Structures")

5.5. Classification



5.5.1. IFC Types and IFC Predefined Types

1. All elements in IFC models should be exported to correct IFC Types

2. All BIM elements in the IFC may also further classified into IFC Predefined Type, that:

IFC Types	IFC Predefined Types
IfcWall	IfcWallTypeEnum.MOVABLE
IfcWall	IfcWallTypeEnum.PARAPET
…	…

   a. These requirement should be set out in the EIR or BEP, which should specify a full list of IFC classes and their Predefined Types. This would be a list analogous to the Master Type List of the DEVB BIM Harmonisation Guidelines for Works Departments, e.g.:

   b. If this list is not provided and thus no requirement is specified, all elements should be mapped to correct IFC Types as a minimum.

3. While native BIM authoring softwares usually provide default mapping settings from their native categorisation system to IFC classes and predefined types, additional manual setting for the following situation should be required:

   a. Nested objects (e.g. object inside a Revit family, curtain wall member in Archicad, etc.)

   b. Native category system mapping to IFC type is one-to-many, for example:

      (i) Cable trunking (should be IfcCableCarrierSegment.CABLETRUNKINGSEGMENT) modelled as cable tray in Revit (exported as IfcCableCarrierSegment. CABLETRAYSEGMENT)

      (ii) Wall finishes (should be IfcCovering) modelled as wall (exported as IfcWall)

c. All elements that are exported as IfcBuildingElementProxy.

4. All IfcBuildingElementProxy type elements should be avoided in the exported IFC file as far as practicable, since this is a generic IFC class essentially meaning the objects are of undefined IFC type.All BIM elements should be classified.

5.5.2. Local Classification or Coding System

1. There is no official classification mandate for BIM elements in Hong Kong construction industry.

2. However, additional classification or coding system for BIM objects should be assigned and exported according to EIR / BEP requirements. Usually encountered coding system are required by DevB, ArchSD & EMSD.

3. Unless otherwise stated, all classification system should be assigned through IFC standard IfcClassisification and IfcClassificationReference mechanism.

4. It should be noted that some local coding requirements may adopt other methodology for applying coding to BIM elements, such as assigning through customised attributes. Users should refer to respective documentation.

5. Unless the names of these additional customised attributes are not compatible with the native authoring software, these attributes should be exported directly to the IFC model without mapping.

5.6. Properties

5.6.1. Principle

1. The EIR/BEP should clearly specify all required properties (e.g., LOD-I requirements) that must be included in exported IFC. They could also be documented as an IDS file (see IDM / MVD / IDS (Information Delivery Specification) – Open standard for defining and validating BIM data requirements in a machine-readable format), the preferred openBIM standard for consistency and machine interpretability.

2. These properties must be created within the BIM models and configured in the IFC translator to ensure they are directly exported or mapped to the IFC models.

3. Some BIM authoring softwares can automate this procedure by importing the organization IDS file or linking to bSDD for better consistency, while some would require add-on or manual operations.

5.6.2. Departmental Requirements

1. There might be LOD-I requirements of multiple department stakeholders to be complied with. EIR / BEP should list out these stakeholders and their relevant BIM standards.

2. Common stakeholders for government projects are:

    a. Development Bureau,

    b. Architectural Services Department, for builder's work maintenance,

    c. Electrical and Mechanical Services Department (EMSD), for building services maintenance.

3. Properties required by different standards may overlap, meaning they might be same property with the same value but with different property names. However, they should not be combined into one property.

    For example, in DevB Harmonisation Guide Appendix VI Section 4 identifies, DSD.Com.Asset Code and EMSD.Common.Asset Code are identified to be both refering to "Asset Code" for solutions. However, it would be prudent to keep both fields independent, as they might refer to different sets of codings.

4. Some properties are derived from combining several element properties, meaning they are calculated values based on multiple attributes of the model. Automatic mapping can be explored and set up in the BIM authoring software to improve efficiency and reduce discrepancy.

5.7. Geometry Conversion

5.7.1. Purpose and Geometry Method

1. Federated model should be for "reference view" only:

    a. The imported / referenced IFC model should not be modified by receiving party.

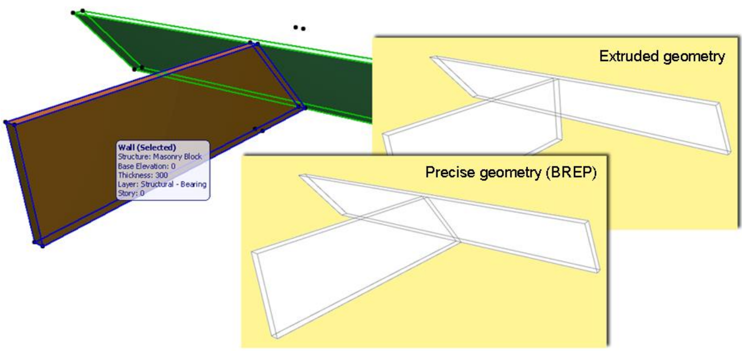
    b. Exported geometries should be exact reproduction of the shape, and their shapes should not be changed by the viewing softwares even if it contains parametric definition.

2. BREP (Boundary Representation) for geometry conversion in export setting should be used.

3. If the BIM authoring software has functionalities to cut out overlapping geometry, the result of such cutting should be exported instead of their original geometries before cutting. (e.g. priority junctions in Archicad, Join function in Revit, etc.)

4. IFC4 Reference View adopts BREP conversions, by default.

5. There is no need to specify the BREP conversions in IFC export settings when IFC4 RV is used.



Parametric (extruded) geometry (IFC4 DT) vs Boundary Representation (IFC4 RV)

6. Other geometry transformation recommended:

    a. Elements should be split by floors

    b. Curtain walls should be split by panels, mullions and transoms.

    c. Finishes should be separated from composite walls and floors

5.7.2. Geometry Fidelity and Polygon Count

1. As far as clash detection accuracy is concerned, Users should adjust tessellation density during export from authoring tools (e.g., Revit, ArchiCAD, or Tekla) to retain higher resolution by better approximating curved or complex shapes, reducing false positives or misses in tools like Navisworks or Solibri.

2. However, at the same time, high polygon count BIM elements in IFC should be avoided. Main caveat for unreasonably high polygon count elements are curve surfaces. Solutions:

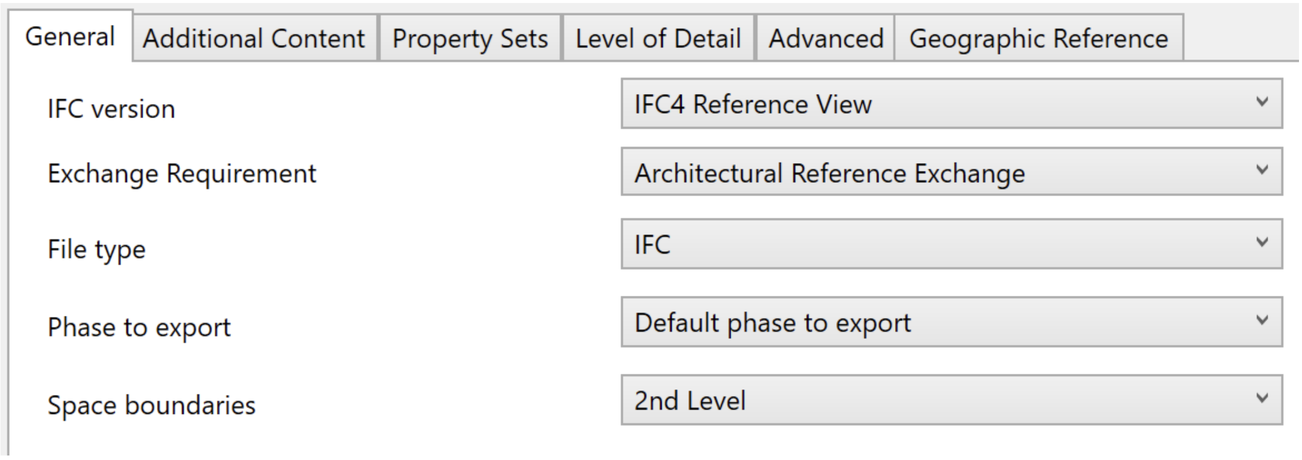
    a. Use the latest IFC exporter available to the BIM tools. For example, IFC exporter in latest Revit has significant polygon count reduction in export over previous version.

    b. Adjust number of facets in export, such that fidelity of curve surfaces are adequate for BIM uses on receiving end.

    c. Simplify native model, e.g. avoid unnecessary double curvature surface in elements.

5.7.3. 2nd Level Space Boundaries

1. 2nd level space boundaries are used by many analysis packages that require a surface view of the building that can be transformed into the various simple topological models. Examples of such analysis packages include: (1) energy analysis, (2) lighting analysis, (3) fluid dynamics.



2. 2nd level space boundaries should be exported.

3. While different BIM authoring softwares' implementation vary, in order to export correct 2nd level space boundaries, the IfcSpaces and their bounding elements (e.g. walls, separation lines, etc.) should be modelled in the same file and exported at the same time into IFC..

5.8. Model Element Filtering

5.8.1. Export according to Modelling Responsibilities

1. Duplicate model elements may occur across design disciplines due to various reasons, with the commonest being the need for the same element in their respective drawings.

2. To prevent duplicated elements in the federated model, the BEP should define which sets of BIM elements should be exported by which discipline, for example:

   a. Structural columns should be included in structural models;

   b. For staircases, the exported architectural model should only contain the finishes but not the structure.

3. DevB's Harmonisation Guide - Appendix III - Data Filtering Rule Table (v2.0)should also be referenced to for demarcation of modelling among departments.

4. As the modelling responsibilities varies from projects to projects, the following are common items which should be defined in the BEP:

   a. Modelling structures in architectural models

   b. Wall, floor and beam opening for building services penetration

   c. Elements on false ceilings

   d. Sanitary fitments

   e. Concrete plinths for heavy equipment

   f. Concrete water tank

   g. Manholes

   h. Fire rated enclosure for MEP pipes and ducts.

5.8.2. Exporting 2D elements

1. Exported IFC models should also contain the following 2D elements:

   a. Grids

   b. Element footprints, e.g. Walls

   c. Element symbols, e.g. door swing

5.8.3. IfcSpace in Architectural Model

1. The architectural IFC model should contain IfcSpace that corresponds to spaces in the building.

2. In other words, all spaces or rooms should have IfcSpace assigned to them. They include:

a. Rooms not mentioned in the Schedule of Accommodation

b. All circulation spaces,

c. All MEP pipes ducts, water tanks, etc.

d. Terraces and roofs

e. Inaccessible spaces, voids, etc

f. Creep space such as smoke vents

g. Space around buildings, landscape areas, open spaces, etc.

3. Upon federation of all architectural models, each space should be represented by one non-overlapping IfcSpace element.

4. IfcSpace is the single most important set of entities in a BIM model. IfcSpace essentially gives an address to each space in a building.

5. Principles of creating IfcSpace are:

   a. Counterparts of IfcSpace in Archicad is "Zone" and in Revit "Room".

   b. Physical room space is a space bounded by walls, columns, floors and doorways, without overlapping with any solid elements. The concept is similar to Revit "Room".

   c. All accessible and inaccessible spaces should be assigned with IfcSpaces including: pipe ducts, shafts, voids, pits, terraces, flat roofs, etc.

   d. They should be separated from UFA, UFS, GFA, Compartment demarcation, which do not necessarily and consistently have same definition as the physical room space.

   e. To better understand the concept, in archicad, use "Zone" in combination of layer and custom property to mimic:

      (i) Revit's "Room" for physical room space,

      (ii) Revit's "area plan" for GFA, UFA, UFS, compartment, etc

   f. Revit's areas in area plan, by default, are exported as IfcSpace. These areas should be filtered out and not included in the IFC exported.

   g. According to current common practice, the heights of the IfcSpaces are usually set to their required headroom. Some might not agree to this as IfcSpace should represent the whole physical space but not an information for headroom requirement. Therefore, the EIR / BEP should state what the heights represent.

5.9. Relationship and Grouping

5.9.1. Object Location

1. All BIM objects should have information on which room (IfcSpace) it is located.

2. If any objects cannot be set relating to an IfcSpace, it should be at least set relating to an IfcBuildingStorey, i.e. located on which level.

3. Some BIM authoring tool e.g. Revit or Archicad, natively has this room containment information. They can export this containment information to IFC automatically if and only if the object and room are placed (or hotlinked in archicad) in the same model and exported to the same IFC file.

4. In some BIM authoring tool, due to model size and software performance concern, it is common to place BIM objects, e.g. furnitures and equipment in models separated from model containing rooms (IfcSpace). It should be noted that objects exported from such model setup would lose their room containment information.

5. In some IFC viewer (e.g. Solibri), information similar to room containment (e.g. related room) can be calculated even though the exported IFC objects do not carry containment information.

5.9.2. MEP System Grouping

1. All MEP systems should be created as IfcSystem or its child type IfcDistributionSystem,etc.

2. All MEP elements should be grouped via IfcRelAssignsToGroup to one of more of these systems.

3. Some BIM Authoring tools e.g. Revit or Archicad, can automatically export mep grouping relationship if the grouping is done using their native function.

5.9.3. MEP connections

1. All MEP elements should be connected via IfcRelConnectsPort relationship

2. Some BIM Authoring tools e.g. Revit or Archicad, can automatically export mep grouping relationship if the connections are done using their native function.

3. A system should not be split across multiple models. It should be noted splitting a MEP system into multiple models (for example, split by floors) breaks their connection relationship (for example all vertical risers are disconnected).

4. Where model splitting can not be avoided, efforts should be made to minimise the number of split models of same MEP system.

5.9.4. Other Grouping

1. Where the BIM authoring software provides manual grouping functions, any groups of built system or zones should be identified and group as IfcBuiltSystem or IfcZone respectively. For example, MiC modules or Fire compartments, etc.

2. Some local standards may requires grouping of elements be assigned using customised properties. Relevant document should be referred to.

5.9.5. Room Adjacency relationship

1. Room adjacency refers to whether 2 rooms are located adjacent to each through openings or non-physical boundaries (eg. Room separating lines in Revit)

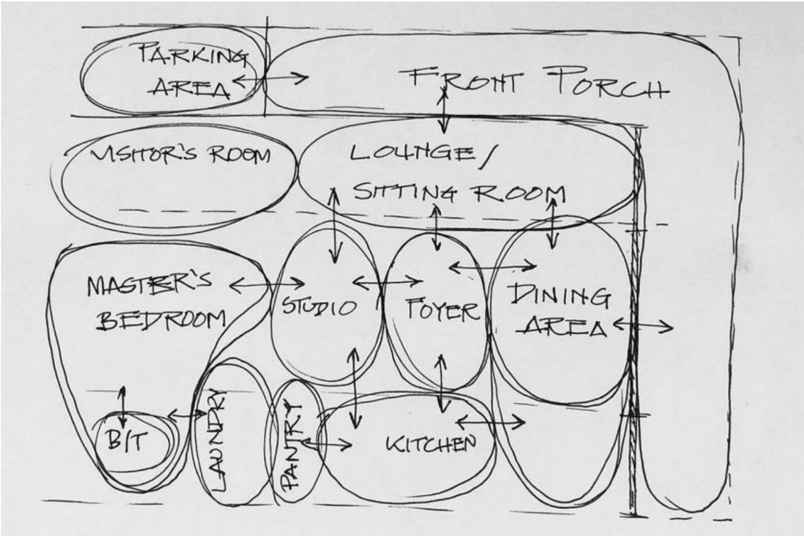


Figure 13: Example adjacency diagram source: https://www.archisoup.com/adjacency-diagrams

2. Room adjacency is important for further smart calculations, such as escape routes, properties for elements among adjacent rooms.

3. Most BIM softwares do not provide room adjacency relationship directly.

4. Some BIM softwares can calculate and derive room adjacency from the exported IfcSpace with 2nd level space boundaries. Therefore, efforts should be made to export 2nd level space boundaries into the IFC model.

REFERENCING IFC

6.1. Shared IFC as Reference Only

1. Shared IFC models from other parties should be used for reference only. No editing should be allowed.

2. The shared IFC models should be referenced (by linking or importing, subject to software implementation) into the native BIM authoring software as underlay model during model authoring.

3. The IFC should be referenced as is without modification, that:

    a. No conversion of geometry into native software geometry

    b. No mapping of any IFC types into native software object types

    c. No mapping of any properties into native software properties

4. The referenced IFC models should not be used for drawing production, since most BIM authoring software could not meet local drawing standards for drawings generated from IFC models.

6.2. Coordinates and Positioning

1. The imported IFC should be positioned back to its correct location according to the diagrams in section “Exporting IFC > Coordinates”, that:

    a. The IfcSite coordinates in referenced IFC model should become the internal coordinates

    b. The IfcProject coordinates in reference IFC model should become the survey coordinates in the native model.

2. However, some softwares (for example, Revit) may not provide options for placing the referenced IFC model. In such case, users would need to manually move the imported IFC to their correct positions.

6.3. Updating IFC Reference

1. The referenced IFC model should be easily replaced by updated version. The positioning of the reference should remain unchanged.

2. Some softwares (navisworks, archicad, etc.) can identify the changes in the updated version against the previous version.

QUALITY ASSURANCE

Quality assurance in this section generally refer to the quality of BIM deliverables while not the quality of the design coordination.

Quality assurance should be carried out in form of BIM audit that consists the following parts:

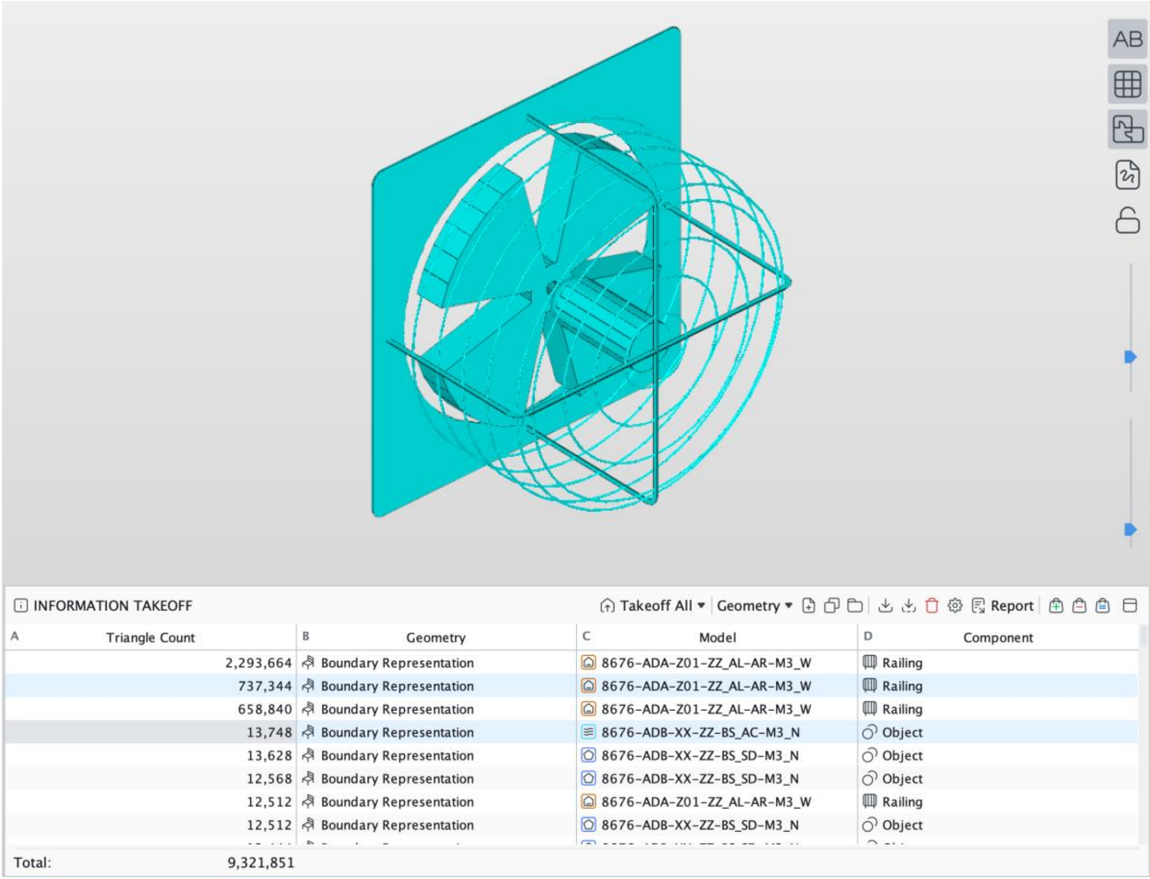
7.1. Auditing Individual Native Models

1. Individual disciplines should carry out audit on their own models before shared to CDE for coordination.
2. Usually carried out in native BIM authoring softwares visually and by listing in schedules.
3. Audit items should refer to BEP and ArchSD's BIM Guide.

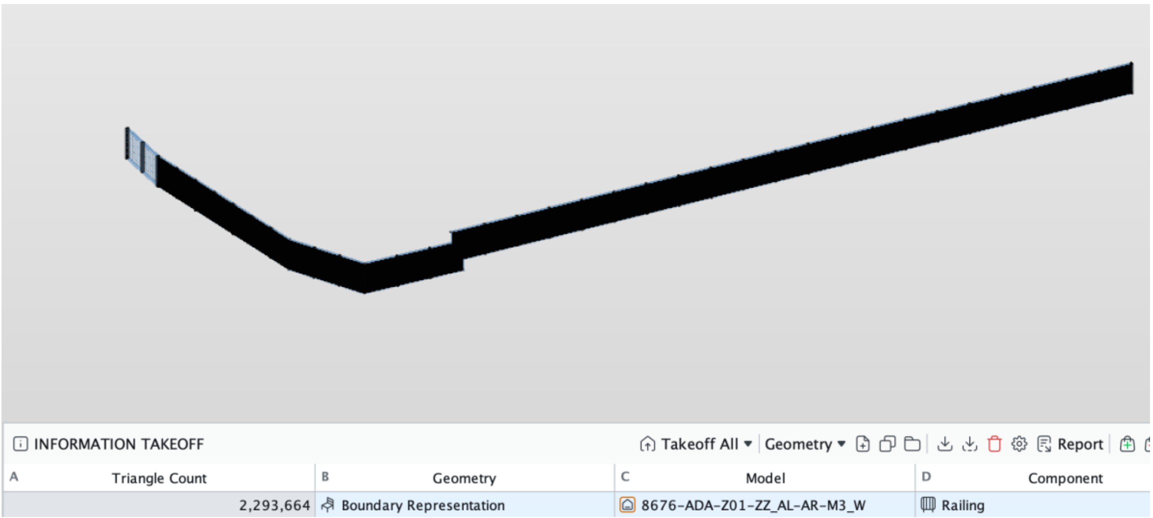
7.2. Auditing Individual IFC Models

The following checks are required on the individual exported IFC models, to be carried out by the both the BIM team leader and discipline BIM coordinator:

1. IFC types and predefined types
2. unassigned spaces
3. overlapping elements from different disciplines
4. overlapping elements from different files
5. unpopulated data fields against information requirements / IDS
6. naming - object / file
7. spatial structure hierarchy
8. Coordinates
9. Polygon Count
    a. High total polygon count in an IFC BIM model directly affect the navigation and calculation performance when viewed in IFC viewers. (The same is true in most BIM softwares)
    b. Elements with unreasonable high polygon count should be identified and flagged as an issue (e.g. via BCF) for the respective BIM author to adjust the export settings for re-export.
    .



High polygon count caused be overly detailed native model



There may be errors in the native model elements. IFC Translator may also need to be adjusted

7.3. Auditing Federated IFC Models

Audit to be carried out by BIM team leader at interval agreed in BEP.

Model Quality Audit:

1. Model structure, data validation and completeness

BIM Audit Item	Content / Methodology
LOIN compliance	visual checking on required detail level on geometries check whether required properties have value
Naming convention	include model file names, view names, sheet names, object names, etc.
Unassigned spaces	whether all spaces have ifcSpace assigned
IFC class	whether all elements are exported to correct IfcClass by manual checking (visual checking)
Model performance	Whether the model can be navigated on mid-end computer; Whether the model has excessively high polygon count objects.

2. Spatial checking

BIM Audit Item	Content / Methodology
Unintended model components	Ensure there are no unintended model components
Interference Checking	Detect problems in the model where two building components are clashing

7.3.2. Model Walkthrough for Completeness and Consistency

1. Model Walkthrough should be carried out regularly, led by BIM Team Leader, with all discipline BIM Coordinators.

2. The following table shows a list of items model completeness and discrepancies generally should be checked during walkthrough.

Model Walkthrough Items		D - design initiator I - to incorporate / coordinate			
		AR	ST	BS	LA
Setup					
1	Site coordinates - AR to setup master IFC site model. - All models must be able to do round trip import and export from and to this coordinates setup in this master IFC site model.	D	I	I	I
2	Scope of works - make sure all works areas are covered in models. All areas must have at least indications of design intent. - Define existing or new elements	D	D	D	D
3	Grids & Boundary	D	I	I	I
4	Building Storeys and structural levels - AR to clearly define storeys and their structural levels & finished floor levels - All models to tally AR model's storeys and levels	D	I	I	I
Building					
5	Slab levels and slab edges	D	I		
6	STR elements generally tally with AR layout - ST columns, ST walls	D	D		
7	Headrooms - per regulations (eg. 2000m) - per design requirement	D	I	I	
8	AR openings (doors, windows, etc.)	D	I		
9	BS louvres - internal / external walls, doors, etc	I	I	D	

Model Walkthrough Items		D - design initiator I - to incorporate / coordinate			
		AR	ST	BS	LA
10	BS Floor openings - is Pipe duct a shaft or with floor slab?	I	I	D	
11	BS penetration through structures (also known as clash)		I	D	
12	Exposed BS in external area - Gutters - External downpipes - Vent pipes outlets	I		D	
13	Exposed BS in FOH area - Valves and overflow in toilets - Floor drains / BITG	I		D	
14	BS FRR enclosure	I		D	
15	Ceiling access panels	I		D	
16	Pipe ducts and plant rooms door access and maintenance space	I		D	
17	Manholes & pits		I	D	
18	Hanger walls	D	I		
19	Drainage and sewage channels - any sunken slab / ST detail?	I	I	D	
20	Existing underground utilities (UU) - any critical UU not divert-able? - Any diversion?			D	I
21	Existing underground structures - Tunnels - Footings	I	D		I
External Works					
22	Hard landscape layout structures - pavilions, shelters, seatings, fences - Pavements, kerbs - Water features & filtration - Signages	D	I		D

Model Walkthrough Items		D - design initiator I - to incorporate / coordinate			
		AR	ST	BS	LA
23	Hard landscape BS installation structures - BS pillar boxes - BS valves & switches boxes - Manholes & pits - Light / camera poles	I	I	D	
24	Hard landscape BS features - Lightings	D	I		
	Hard landscape level fall - vs gullies and channels	D	I		
25	Soft Landscape vs underground obstructions - Required soil depth vs footings / BS		I	I	D
All disciplines					
26	Ground level - across internal to external to existing level or newly formed streets, with falls - Below ground services & concrete surrounds - Underground trenches & pits - Excavation levels	D	D	D	D

7.3.3. Clash Analysis

Clash analysis is a critical BIM coordination process for detecting and resolving conflicts between building elements. Traditionally based on Revit categories, the shift to openBIM introduces standardized systems like IFC types or the DEVB Harmonisation Guide's CAT code to improve interoperability and efficiency.

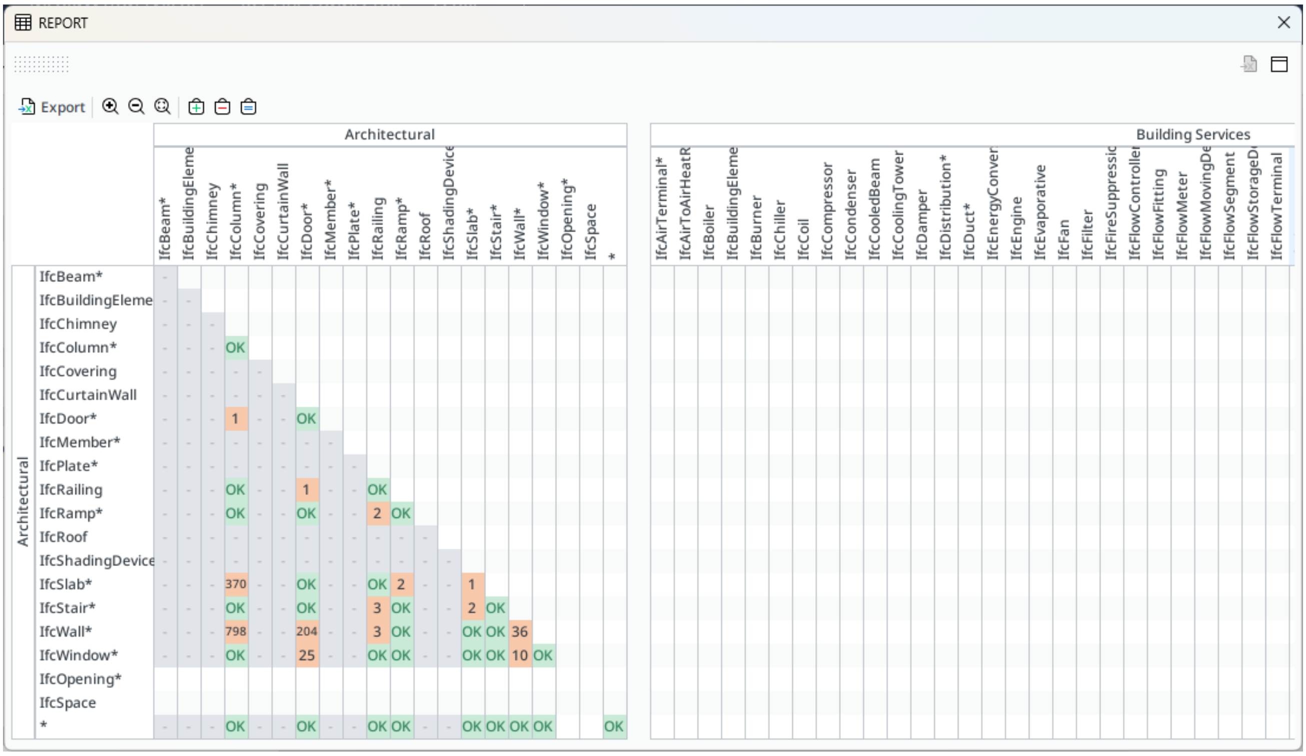
1. Clashes are identified by comparing two element sets: Set A (e.g., structural elements like columns) and Set B (e.g., MEP systems like ducts), allowing detection of conflicts.

2. In single proprietary software ecosystem, platform specific object categorization (e.g. Categories in Revit) is usually used to define Sets A and B for clash detection.

3. With openBIM, IFC types (e.g., IfcColumn, IfcPipe) should be used for defining Sets A and B, ensuring standardisation across different BIM software.

4. Alternatively, the DEVB Harmonisation Guide's CAT code provides a local standardized coding system for defining Sets A and B.

5. A clash matrix based on IFC types, or CAT code should be established.



Clash matrix example from Solibri (https://help.solibri.com/hc/en-us/articles/23765423193111-245-Clash-Detection-Matrix)

7.3.4. Advanced Automatic Design Checking

1. There are some advance automatic design checking softwares using IFC models in the market, which can check design items beyond clash detection. Project Team may use this kind of BIM softwares for design checking and general BIM auditing.

2. In the Pilot Project, Solibri Office was used to check the design models against the following issues:

Item	Detail
Unassigned Space	Whether any physical spaces having no IfcSpace assigned
Headroom	minimum 2000m for all spaces and staircases
Clearance for doors and windows	check sufficient clearance for door and window opening
MEP equipment clearance	check whether sufficient clearance is allowed around MEP equipment
Coverage	Coverage for sprinkler, hose reel, etc.
Architectural vs Structural models	Check whether the structural elements align and fit in the architectural layout

3. It should be noted that in order for such automatic design checker really works, the input BIM model should be relatively tidied up. For example:

   a. All model elements should be correctly classified or properly coded.

   b. The model building setup, such as location, levels should be aligned across all models

8. ISSUE TRACKING

8.1. BIM Collaboration Format (BCF)

1. BCF is a standard protocol for communication. It may be use for:

   a. Discussion of design issues

   b. BIM Issue discovery, discussion, responses and tracking. It provides audit trail of all issues encountered and solved.

   c. To replace email communication regarding issues that can be better represented with their location in models and mark-ups.

8.2. Ideal BCF workflow

1. The following describes an ideal workflow of using BCF within a project:

   a. User marks up comments (create BCF issues) in their native BIM authoring / viewing softwares.

   b. The BCF issues automatically delivers to all other users.

   c. The user on the receiving end can read the BCF issues in their own native BIM authoring / viewing software. Clicking on the issue retrieves the exact view point where the issue was created. The receiving user replies to the issues and the reply automatically delivered to all other users.

   d. All issues and corresponding comments and replies are stored, and can be easier retrieved, summarised and analysed.

2. There are commercial BCF platforms, for example, BIMCollab and BIMtrack etc., that make effort to implement this workflow by providing add-ons to as many as BIM softwares available.

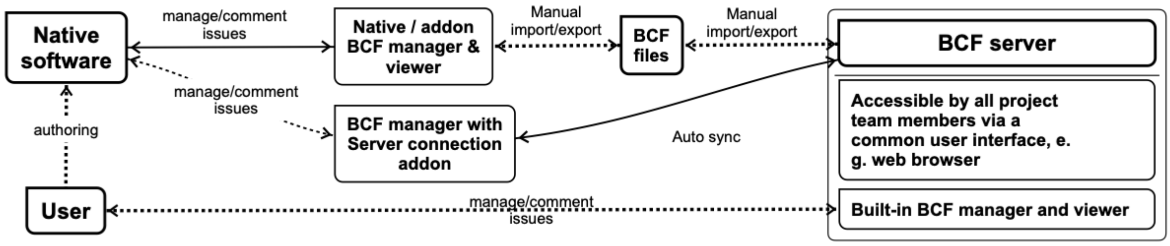
8.3. BCF in Practice

1. Despite various BCF platforms' effort to implement the perfect BCF workflow, It is not uncommon that the BCF workflow would be a hybrid workflow combining online platform and offline manager since:

   a. It is not realistic to expect the BCF platforms to provide add-ons to every BIM softwares on the market, especially for those without API access.

   b. Those BCF manager addons from BCF platforms may not be as integrated and user friendly as those built-in in native software.

2. The following diagram shows how users would manage issue (view, create, comment, etc.) with a BCF platform:



a. Users may manage the BCF issues directly in the viewer in the BCF platform.

b. User may manage in the BCF add-on in native BIM softwares, and the add-on would automatically sync with the BCF platform in real time.

c. User may use native BCF manager in BIM softwares, then "sync" manually and asynchronously (i.e. not real time) the issues with the BCF platform by manually importing and exporting BCF files, on demand or at regular intervals.

8.4. Choosing BCF servers

1. Below are some examples of centralised place for managing BCF issues. They may be integrated CDE platforms, BCF only platforms, online file based storage, or offline software solution.

BCF Server	Remarks
Trimble Connect	Online CDE
Dalux	Online CDE
BIMCollab	Online
BIMTrack	Online
Local File Server	Intranet, file based. Require viewers for analysis
Solibri with local file	Offline, save in Solibri file Can sync with Trimble Connect, interoperate with Autodesk ACC, etc.
Autodesk Collaboration Cloud	Not BCF compatible

2. Below are BCF functions provided in 3 common BIM authoring softwares:

	Revit	Archicad	Tekla Structures
Built-in BCF function	No	Yes, issue manager	Yes
BCF import and export	No	Yes, issue manager	Yes
BCF function via plugin	Yes	Yes	Yes
Sync with BIMCollab	Yes, via their BCF add-on	Yes, via their BCF add-on	Yes, via their BCF add-on
Sync with BIMTrack	Yes, via their BCF add-on	Yes, via their BCF add-on	Yes, via their BCF add-on

8.5. Compatibility across Softwares and Platforms

1. It is crucial that the selected BCF server can support round trip import and export of BCF files to and from other softwares.

2. BCF 3.0 (currently latest version) should normally be used. However, the BIM team leader should test whether all the softwares adopted in the project would be compatible with the adopted BCF version, and state in BEP.

3. To ensure optimal compatibility when visualizing an issue in various BCF-compatible software applications, an issue should include the following components: :

a. Involved elements selected

b. Section planes to show the area

c. Screenshot

d. Camera and target location

e. Orthogonal view more preferable than perspective view (since some of BIM softwares may not support perspective view)

8.6. Pre-defined Metadata Value

1. Each issue should be created with the following metadata. The value should be agreed in the BEP. Below are examples from the Pilot Project:

Assignee

Assignee	When to use
(Name, or group)	Who should respond to the issue

Topic Type

Topic Type	When to use
Discrepancy	Discrepancy found among different disciplines' models
Model Error	Modelling mistakes or Error not related to design
Request for Info	To request further Information from others
Clash	Errors found in Model Audit / Clash Detection
Other	Issues cannot be categorised by other topic types

Topic Statuses

Topic statuses	When to use
New	All new found or unresolved issues
Done	Issues resolved with model updated, by party who respond
re-open	Done issues found to be still unresolved, by member who raised them
Closed	Issues verified and closed by member who raised them

Topic Priorities

Topic Priorities	When to use
Critical	Issues to be resolved in 1 week
High	Issues to be resolved in 2 week
Normal	Issues to be resolved in 4 week
Low	Issues to be resolved upon project completion

Tag

Tag	When to use
AR	Architectural related
LA	Landscape related
ST	Structural related
AC	Air Conditioning related
DR	Drainage related
EL	Electrical related
FS	Fire Services related
PL	Plumbing related

MODEL FEDERATION

9.1. Federation Platform / Software

1. In the past, model federation are often carried out by BIM manager of the project using offline standalone softwares such as Autodesk Navisworks.

2. As BIM online platforms and CDE become more and more mature with functionalities, the model federation has been moved to these online platforms, for example Autodesk Collaboration Cloud and Trimble Connect.

3. The benefits of online model federation platforms over previous offline model federation softwares are:

    a. Allow real time model federation. No need to wait for BIM manager.

    b. Allow users to federate different sets of models.

    c. Real time statistics of models available.

    d. More accessible, both user friendliness and available on multiple devices (e.g. web, smart phones, tablets), not limited to single computer.

4. Permissions should be set for all users to allow them to federate the models of their choice without the help of BIM manager.

5. As far as practicable in the project procurement, online platform accessible to all project team members should be provided for model federation, and all users should be permitted to federate the models by themselves.

6. The model federation platform may be integrated into the CDE. The assessment criteria for this platform should refer the Chapter "Common Data Environment".

7. It should be re-iterated that integrated model federation function on some CDEs do not meet the recommendation set out in the Chapter "Common Data Environment". Procurer should assess whether complementary solutions should be included.

8. On the other hand, BIM team leaders may have their own sets of federated models on their offline software for advance functions not provided in the online platform such as clash detection and audit checking.

9.2. Model Versions for Design Branching

1. As online federation platforms allow users to select their own sets of models for federation for their purpose, it becomes even more important for project team members to be able to identify different versions of models and their design content.

2. A typical project team design process is illustrated in the following diagram. The diagram explains that the design process usually contains several design branches, that:

    a. The master branch is a design-in-process model aiming at becoming the design intent model to be constructed. The design head of each discipline should decide which sets of their design models should be uploaded for this purpose.

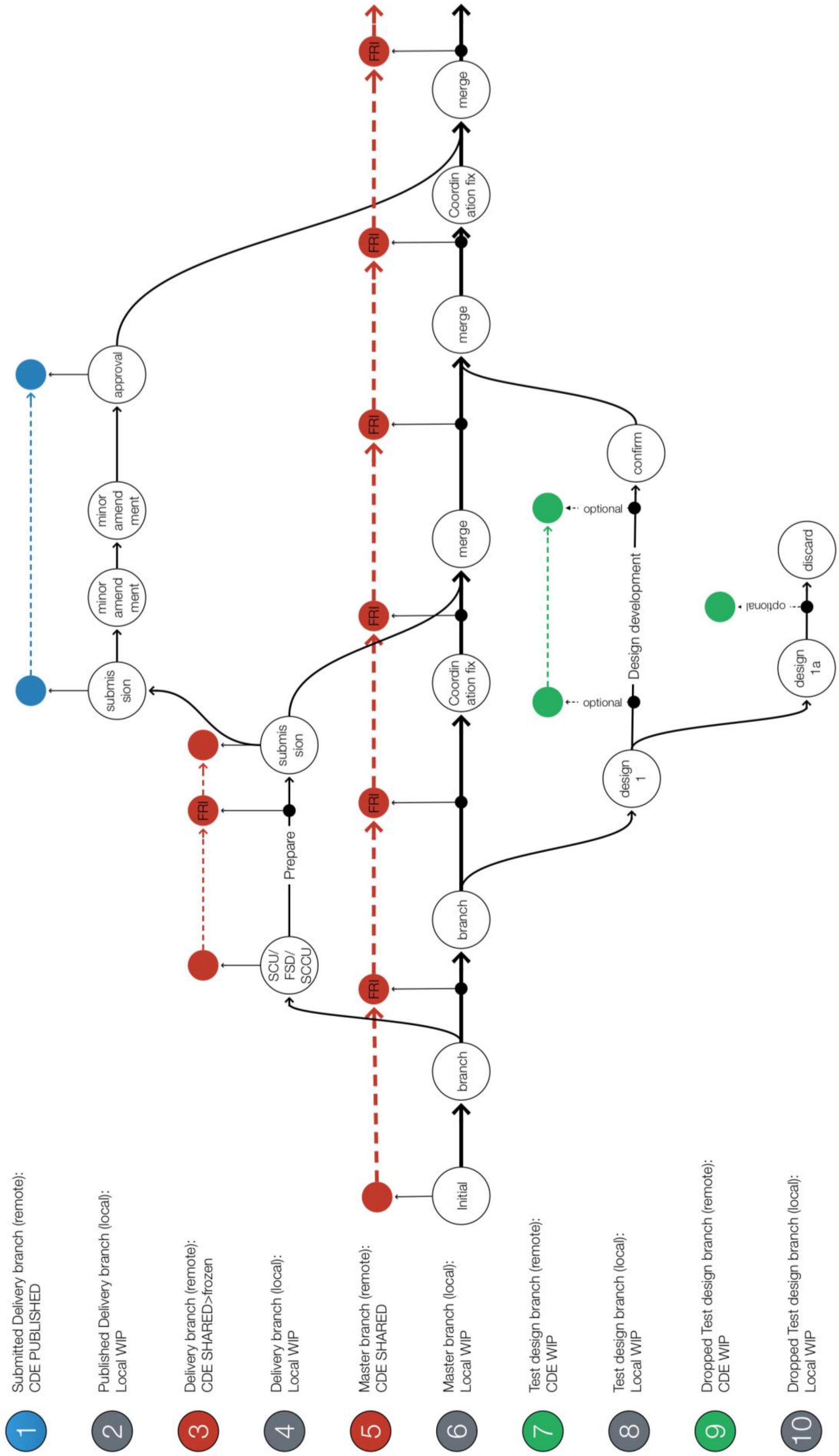
    b. From time to time, there are design models branching out from the master branch for different purposes, such as for design testing, statutory or milestone submission.

    c. Once any branched design is confirmed, it is merged back into the master branch.

3. The diagram also illustrates where and when the files for the design model should be stored as example:

    a. In local file repository, i.e. local folders, or the local collaborating BIM authoring platforms.

    b. In remote file repository, i.e. the Common Data Environment. (Refer to CDE Folder Structure for explanation)



Design Process Diagram

Item	Design / Collaboration process	Upload to CDE
2	Submitted deliverable Branch for submitted deliverable, and any post amendments, e.g. SCCU submission, minor amendments and approval	- upon submission and final version upload
4	Deliverable Branch for submissions / milestone deliverable such as statutory, tender.	- Regularly every Friday - Upon branch out - Upon submission
6	Master branch of design intent model aiming towards model for construction	- Regularly every Friday
8	Test Design Branch Merge to master branch upon confirmation	- Optional for design evaluation in model federation
10	Dropped Test Design Branch dropped during the process. No effect to master branch	- Optional for design evaluation in model federation

Legend for the "Design Process Diagram"

4. In order to easily keep track of design branches, a robust model version convention and a register would be necessary, for example:

date	branch code	branch description	status
2024-06-10		initial design	published
2025-12-31	20240610-1	modify column location	active
2025-01-23	20240610-1b	omit hall columns	active
2024-12-15	20240610-1a	relocate hall overhead transfer beam	retired
2025-01-10	20240610-2	relocate AHU room to roof	retired

a. Give the branch a simple code.

b. For example, date with alternating number and alphabet, 20231122-1c3b = first branch's 3rd branch's 3rd branch's 2nd branch from 2023-11-22 model version

c. Create a folder with this branch name to hold the BIM model and related files. Avoid changing the name of the BIM models to carry the version name.

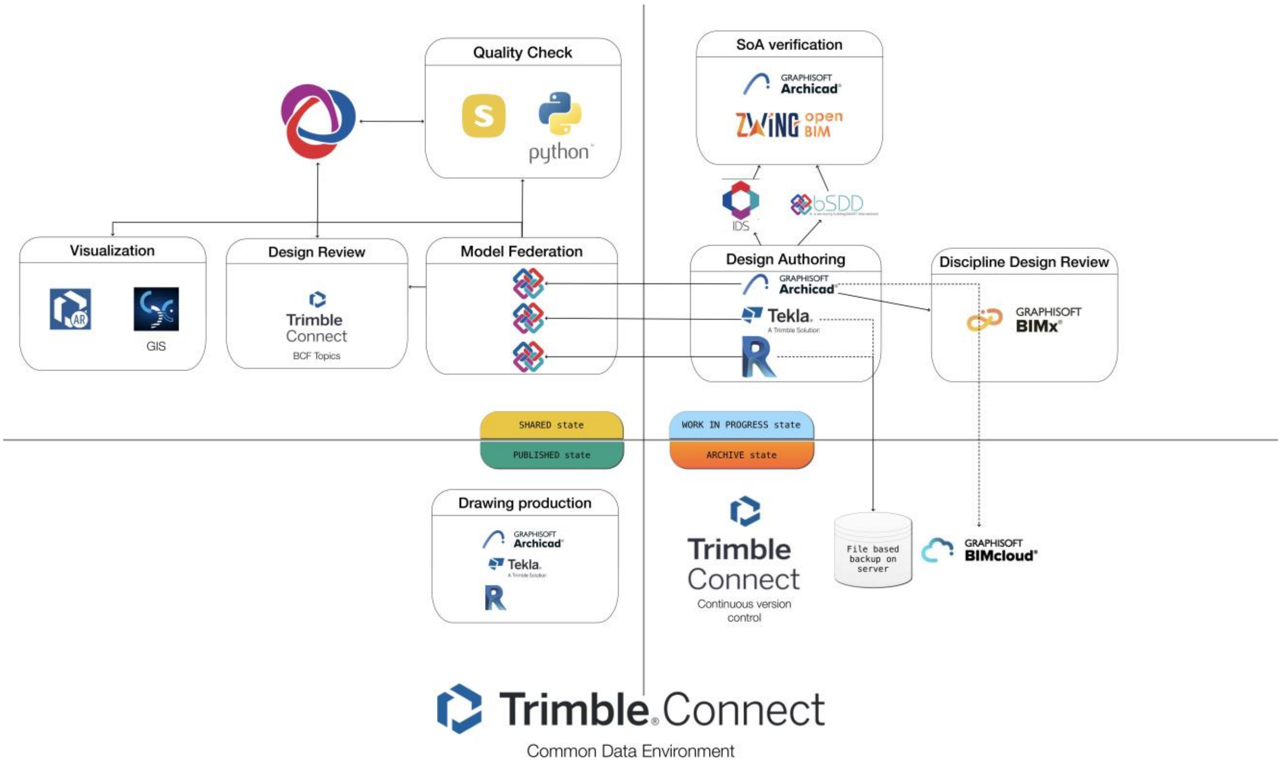
9.3. Software Ecosystem around Federated Model

1. Model Federation is the central part of many BIM uses. Many of the functions described in previous chapters revolve around federated models

2. In the Pilot Project, these functions were implemented via the following platforms and software.

BIM use	Software / platform
BIM Authoring	Graphisoft Archicad, Autodesk Revit, Tekla Structures
Discipline Design Review	Within native authoring tool, underlaid with referenced IFC models from other disciplines when necessary. Trimble Connect was also used for model review since it was easier to navigate than in native BIM authoring tools.
CDE with integrated viewer	Trimble Connect as file repository and BIM viewer
Inter-discipline 3D coordination	Trimble Connect All disciplines exported IFC models and uploaded onto CDE. Any team members could select which models to federation for design review and coordination.
Model Audit, Advanced Audit	Solibri Office Customed Python script was used for checking consistency for model coordinates and particulars of IfcProject and IfcSite.
Clash Detection	Solibri Office
Issue tracking & response	Issues are raised in several platforms: - Solibri, for results from automatic checking and clash detection - Trimble Connect, for visual checking from any team members. All issues are synced to Trimble Connect's BCF Topics, which is a BCF server.

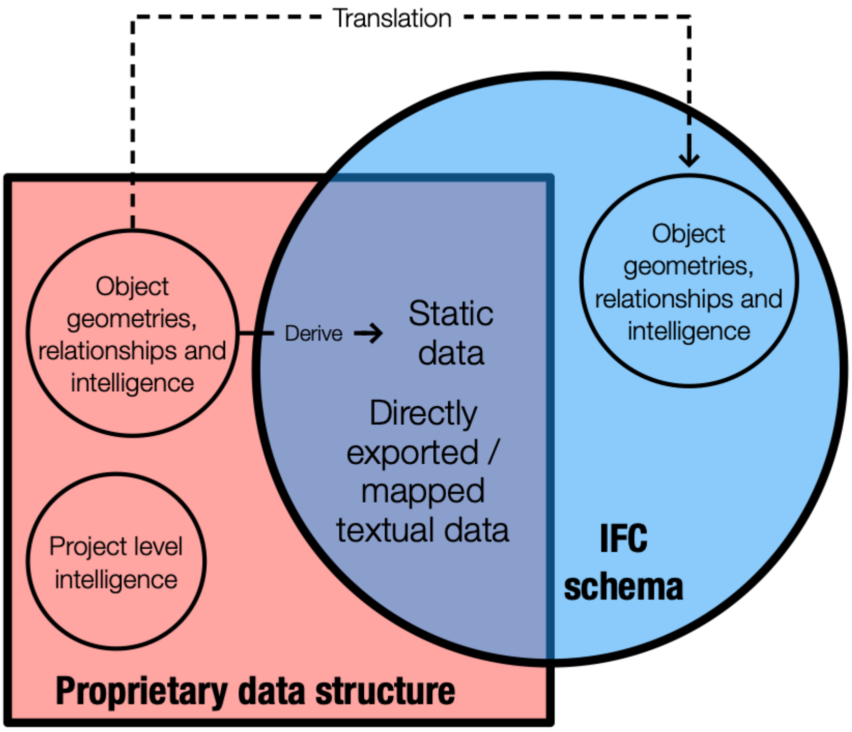
MODEL FEDERATION



Common (Mis-)Understanding

The local construction industry has been hesitant to adopt OpenBIM workflows, primarily due to concerns about data translation when exporting native BIM models to IFC models. Understanding this process and its benefits is essential for encouraging wider adoption of OpenBIM practices.

10.1. Fear of Data Loss in Translation



1. Exporting BIM models from native BIM authoring software (e.g., Archicad, Revit, Tekla Structures) to IFC models is a process of data extraction and restructuring to align with the standardized IFC schema.

2. Like translating languages such as Chinese to English, this process strives for comprehensive representation but inherently cannot convey every embedded nuance, even with detailed mappings, requiring selective inclusion of essential data.

3. Despite these concerns, exporting to IFC is critical for communication and collaboration across diverse BIM authoring tools.

4. On the other hand, the export process gives opportunity for project teams to:

   a. standardize properties

   b. omitting redundant or sensitive details.

   c. protect trade secrets by selectively sharing data

5. The following are several key area of concerns of data loss:

a. Static Textual Data:

(i) These are static, user-defined values independent of geometric or relational changes, such as material specifications, equipment tags, or room names. These values are typically either:

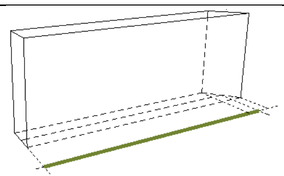
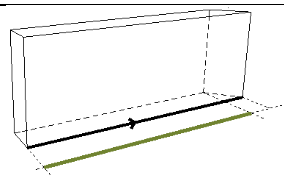
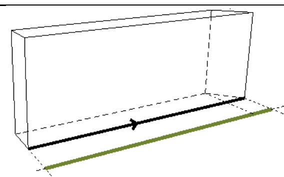
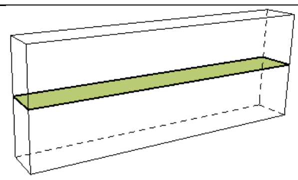
- mapped to IfcPropertySet or IfcQuantitySet attributes, or
- directly exported as user defined property.

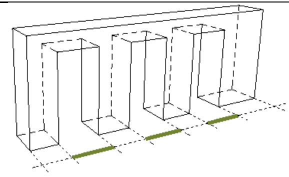
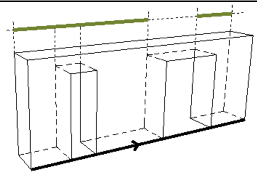
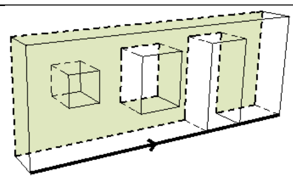
(ii) For example: In a Revit model adhering to the EMSD BIM Standard, custom properties like "Asset Code" or "Maintenance Interval" are defined. Although they are not standard IfcPropertySet such that they may not map directly to predefined IFC Psets, they would be directly exported under a customized property name.

(iii) Observation: This should be straightforward and no loss of data would be expected.

b. Quantity Listing from Object Geometry

(i) Native BIM software often provides rich parametric quantity listing capabilities directly derived from parametricgeometries. Below are some examples from GRAPHISOFT Archicad (see https://helpcenter.graphisoft.com/user-guide/76908/):

Parameter	Illustration
Length of Reference Line	
Length of the Wall on Inside Face	
Length of the Wall on the Outside Face	
Area of the Wall	

Combined Width of doors in Wall	
Conditional Length of the Wall on Inside Face	
Conditional Surface Area on the Inside Face	

(ii) However, as many of these quantities are software specific,when exported to IFC, these quantities must be converted to static values, as IFC does not retain native parametric quantity.

(iii) For example: In Archicad, a complex roof with parametric slopes and overhangs can generate detailed quantity takeoffs for material volumes or surface areas. When exported to IFC, the roof's geometry is represented as a static IfcRoof with fixed dimensions, requiring quantities to be pre-calculated and exported as static values in IfcQuantitySet, losing the ability to dynamically recalculate based on design changes.

(iv) Observation: All schedule-able quantities in native model should be exported into IFC model as static quantities stored in custom properties for future use. Therefore loss should be minimum.

c. Object Parametric Properties:

(i) These are dynamic values that drive or result from changes in a model's geometry or relationships, embodying the model object's algorithmic logic or "intelligence." During IFC translation, these dynamic properties are often simplified into static values within IfcPropertySet attributes, capturing essential data but omitting nuanced logic.

(ii) For example, A change in height and width properties dynamically update an object's geometry in native model, while the IFC export will only record fixed numerical values without the parametric logic.

(iii) Observation: There should be total loss of parametric properties when exporting using IFC4 Reference View. IFC4 Design Transfer has been developed to translate object's parametric properties, but softwares implementation are still uncertain and not certified by buildingSMART.

d. Geometry Fidelity:

(i) Native BIM models often include complex or proprietary geometric representations that may not be fully supported by the IFC schema's geometry definitions (e.g., IfcSweptSolid, IfcBrep). During export, intricate details or custom shapes may be simplified to fit IFC standards, affecting visual or analytical accuracy.

(ii) Example: In an Archicad model, a parametrically designed curved facade with intricate patterns is exported to IFC as a simplified IfcFacetedBrep, retaining essential form but losing fine geometric details.

(iii) Faceted Geometries: Faceted representations (e.g., IfcFacetedBrep) are generally acceptable for coordination purposes, as they provide sufficient accuracy for clash detection or spatial analysis. However, they may not suffice for fabrication, where precise curvature or smoothness is critical—e.g., a curved steel beam's fabrication may require true curve definition rather than IFC's faceted geometry, potentially necessitating native model use or alternative exchange methods (such as sharing Rhinoceros grasshopper's graph).

(iv) Observation: While IFC4 Design Transfer view is still immature, recommended IFC4 Reference View mostly used faceted geometries, that costs polygon counts and suffers losing true curve definition in most cases. However, for common BIM uses like spatial coordination or asset management, faceted geometries usually would be of little concern. Level of geometric fidelity still depends on the BIM use on the receiving end.

e. Object Semantic Category:

(i) Native software uses proprietary classifications or object types that may not directly map to IFC Classes (e.g., IfcWall, IfcDoor). Misalignment during export can lead to incorrect categorization, affecting downstream processes like quantity takeoff or clash detection.

(ii) Example: In a Revit model, a wall skirting modelled using wall tool would be mapped to IfcWall if mapping was not properly configured in the IFC export.

(iii) Observation: The IFC schema provides methods to accommodate different classification paradigms, such as using IfcRelAssignsToClassification or custom Psets, which are discussed in the Section "Classifications". Therefore, there should not be any concern to accommodate and retrieve common classification system in the IFC models.

f. Static Relationships among Objects:

(i) Native BIM authoring software supports various kinds of relationships among BIM objects. For example:

- A wall hosting a door
- Two pipes connected to each other
- MEP routing and equipment grouped together as a system

(ii) The IFC schema also supports a variety of relationships, as discussed in the Chapter 2.3.8 Relationship. Although these relationships may not perfectly match those available in native software, the relationships provided by the IFC schema are comprehensive and more than sufficient for common usage.

(iii) Some users may complain that exporting to IFC causes many of the object relationships to be lost in the software. However, most of these complaints likely refer to the constraints associated with the relationships. For example, when moving a wall, the hosted door moves along with it. This is a nested relationship with constraints, and such nesting relationships are preserved during IFC export.

g. Parametric Constraints among Objects:

(i) These relationships represent embedded rules, constraints, or behaviors between objects, such as parametric dimensional or positional relationships, most for design purpose. The IFC schema prioritizes static data, often reducing these relationships to fixed attributes, dropping dynamic functionality.

(ii) Example: In a Tekla Structures model, a steel connection's parametric rule adjusts bolt spacing based on beam size. When exported to IFC, this is represented as static IfcMechanicalFastener properties, omitting the adaptive logic.

(iii) Geometry Generation (Grasshopper/Dynamo): Parametric intelligence, such as geometry generated via Grasshopper or Dynamo, is typically not preserved in IFC exports. While IFC may not accommodate such intelligence, native but open formats (e.g., Grasshopper scripts, Python code) can be shared separately to convey design logic.

(iv) Observation: There are varying views on sharing parametric design intelligence. It may be considered a trade secret or irrelevant to other project parties, serving primarily as an internal workflow tool. Thus, its exclusion from IFC models can be intentional.

10.2. Models Cannot be Round Tripped

1. Some BIM users would complain exported IFC models cannot be imported back to the native softwares and retain all the original native model "intelligence" and data, i.e. round tripping between native BIM authoring software and IFC.

2. As discussed in previous section, IFC schema does not carry all data and "intelligence" that a native BIM model contains. Expecting full model round tripping is futile.

3. Although IFC4DT has been developed for this common "wish" of round tripping, it is still under development and its capability and software implementation is still unknown. (See Chapter Model View Definition (MVD) and Software Certifications

10.3. Uncertainty incurring Time and Cost

1. Many local BIM standard and guidelines, while maintaining software neutral in the main content, usually use market dominant BIM authoring software for illustration. This effectively implies that by using such software:

   a. All technical issues of that software have been considered, and worked out;

b. And all technical limitation of that software would be accepted.

2. These implications naturally become the technical threshold and incur additional time and cost of using alternative BIM softwares. I.e. If a company / user proposes to use an alternative software, it would need to make sure the software technically can follow with same standard and guidelines, at its own cost. For example:

   a. All BIM objects should be named following DevB's Harmonization Guide. In Revit, this can be done by renaming the object Family name in project or the Family file name (if exported as standalone file).

   b. However, in Archicad, neither renaming object name in project or object file name is advisable. (Reader may refer to Archicad manual for further recommendation)

Appendix A

A. COMPREHENSIVE GUIDE TO OPENBIM WORKFLOW IN ARCHICAD

A.1. Coordinates

Setting up coordinates correctly in Archicad is essential for ensuring accurate model placement and alignment when collaborating with other disciplines using OpenBIM workflows. The following steps outline how to configure the project location, survey point, and IFC export settings to meet OpenBIM requirements.

A.1.1. Define the Project Location

The project location specifies the geographic coordinates and orientation of your project. The latitude, longitude, and sea level information stored in the Project Location settings establish the global coordinates for the project's origin and site, facilitating geolocation purposes. These settings are mapped to specific IFC properties as detailed below:

Archicad Parameter	IFC Data
Latitude	Attribute>RefLatitude
Longitute	Attribute>RefLongitude
Altitude	Attribute>Name

1. Accessing Location Settings:

   To access the location settings, navigate to Options > Project Preferences > Location Settings.

2. Setting Geographic Coordinates:

   Enter the latitude and longitude of the project site in the designated fields. If available, also enter the altitude or elevation data (e.g., mPD).

Defining the Project Location in Archicad ensures your model is geographically accurate for sun studies and energy analysis.

A.1.2. Configure the Survey Point

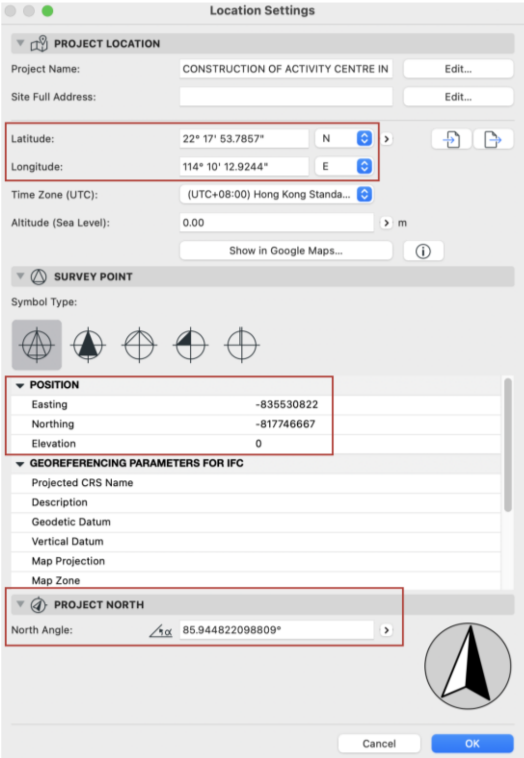
The Survey Point is an essential reference for real-world coordinates, ensuring that models from different disciplines are accurately aligned. When an Archicad Survey Point is established, it serves as the foundation for positioning the IfcSite entity during the export process. The location and rotation of the Survey Point establish the global coordinate system for the exported IFC file.

1. Show the Survey Point

   To make the Survey Point visible on your Floor Plan or 3D view, navigate to Use Options > Project Preferences > Survey Point and select Show Survey Point.

2. Positioning the Survey Point

Open Options > Project Preferences > Location Settings… under the POSITION tab, type the Easting (X) coordinate with the minus sign (-) in front of the value (e.g., -835530822) and then type in the Northing (Y) and Elevation (mPD Level) values again with the minus sign (-) in front of the value.



3. Defining True North

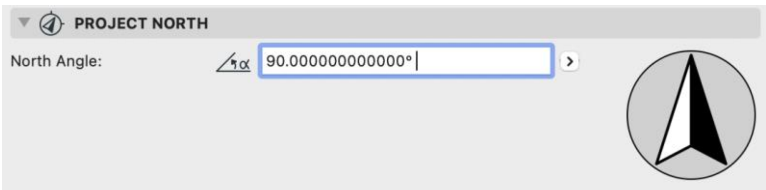
a. By Numeric Input:

(i) Navigate to Options > Project Preferences > Location Settings.

(ii) In the North Angle field, input the azimuth angle to establish the Project North direction.

The accuracy of these units is determined by the settings for Angle Unit, found under Options > Project Preferences > Working Units. When using degrees, minutes, and seconds, enter 45′ 30′ 15′.

In Archicad, the default Project North aligns with 90 degrees (the Y-axis). If adjustments are necessary to align with True North, calculate the difference between 90 degrees and the True North angle.



b. By Graphical Input on Plan:

(i) Go to Options > Project Preferences > Set Project North.

(ii) In the floor plan, click and drag the compass icon to align with your desired North direction.

(iii) Click once to confirm the new orientation.

4. Locking the Survey Point

To prevent accidental movement of the Survey Point during modelling, navigate to Menu and select Options > Project Preferences > Survey Point > Lock Survey Point.

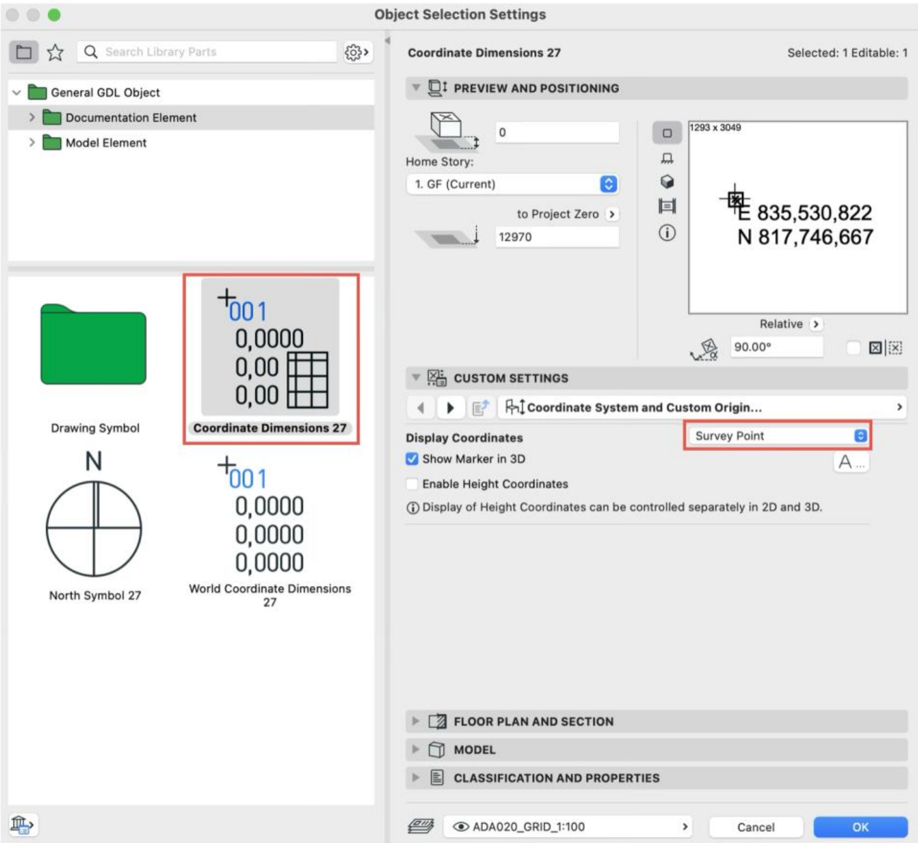
A.1.3. Verify Coordinate Setup

Before exporting your project, it is essential to ensure that all coordinate settings are correct.

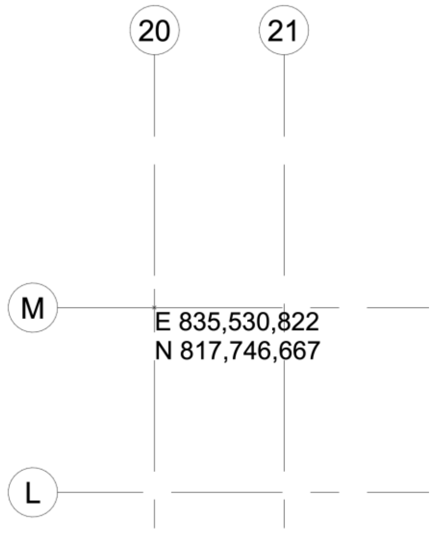
1. Checking Coordinates

To confirm the proper placement of the Survey Point, you can use the Coordinate Dimensions Object, which is located in the Documentation Element folder

The Coordinate Dimensions Object displays the X/Y/Z values based on the position of the Survey Point. If you redefine the Survey Point's location, the Survey Coordinate Objects will automatically update to reflect this change. Open Object Settings, under Custom Settings, set set Display Coordinates to "Survey Point" to display the world coordinates. Place object at the project origin location (0,0,0).



The screenshot below illustrates the Survey Coordinate Object positioned at Grid Intersection M/20 (X: 835,530,822 Y: 817,746,667) in relation to the Survey Point values specified in the previous section.



A.1.4. Setting Up IFC Export Translators

To ensure that your exported IFC files are correctly aligned with your coordinate setup, it's important to configure Archicad's IFC Translators.

1. Accessing IFC Translator Settings

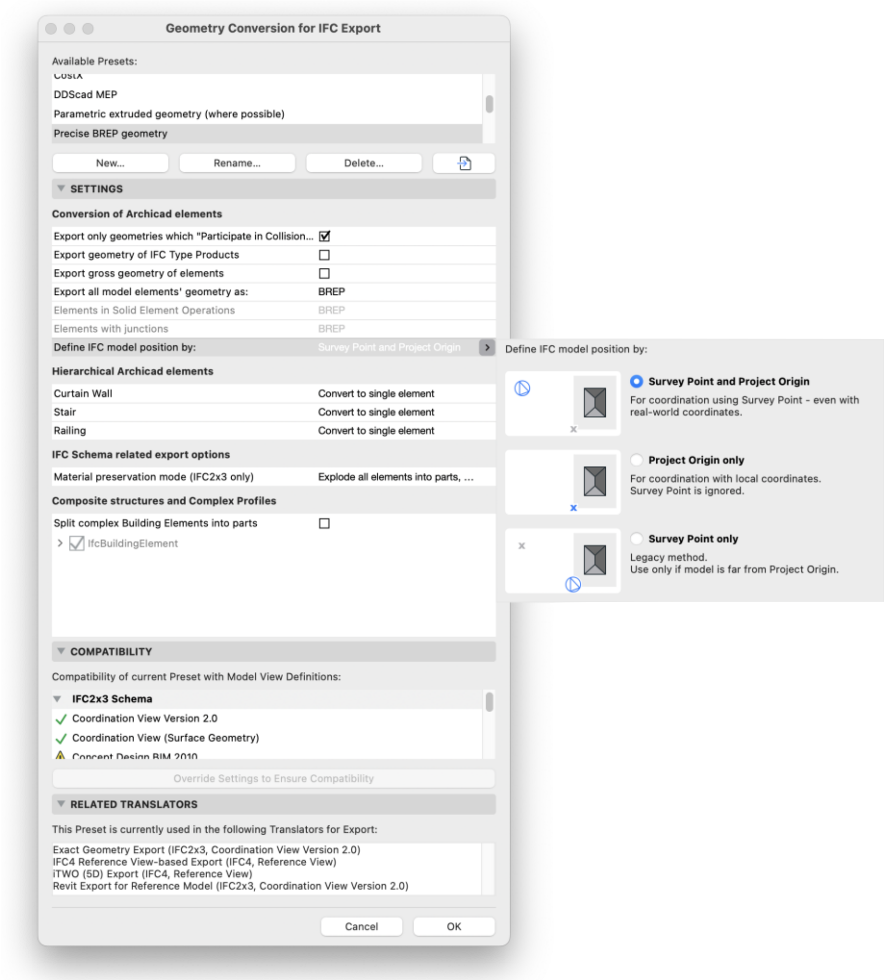
   Navigate to File > Interoperability > IFC > IFC Translators to open the translator settings.

2. Selecting or Creating a Translator for Export

   You can either select an existing translator, such as the "IFC4 Reference View-based Export," or create a new one. To do this, simply duplicate an existing translator and modify its settings as needed.

3. Defining IFC Model Position

   In the Translator settings, go to Geometry Conversion and select Define IFC Model Position By. You have the option to include the project origin, the survey point, or both (recommended). The option "Define IFC model location by:" is set to "Survey Point and Project Origin" as shown below:



The Survey Point acts as a reference for real-world coordinates and geolocation data. It defines a specific geographic location and aligns the model with True North. Additionally, the Survey Point is automatically utilized as the global origin in exported IFC models.

The Project Origin in Archicad is a fixed reference point that cannot be moved. It serves as the origin (0,0,0) for the local coordinate system but does not adapt to geographical factors.

Below is a detailed technical comparison relevant to BIM management and Archicad workflows.

Feature/Aspect	Project Origin	Survey Point
Definition	Internal, fixed zero point in Archicad	User-defined, real-world reference point
Purpose	Local modeling reference	Georeferencing and real-world alignment
IFC Mapping	Becomes the IFCSite Origin (local origin)	Becomes the IFCproject Origin (world coordinates)

Model Coordination	Suitable for internal coordination, not geolocated	Essential for federated models and GIS integration
Precision	High, if model is kept near origin	High, with correct survey data
Typical Use Case	Early design, internal team workflows	Multi-disciplinary, site-based coordination
Export/Import Alignment	Models align by their internal origins	Models align by shared real-world coordinates
IFC Export Setting	Geometry exported relative to project origin	Geometry exported relative to survey point

A.2. IFC Spatial Aggregation Hierarchy

Defining Spatial Structure in Archicad for IFC Export

Defining the spatial structure in Archicad for IFC export involves preparing project/site/building information, modeling stories and zones accurately, assigning elements to the correct spatial containers, and using the IFC Project Manager to review and adjust the hierarchy before export. Proper configuration ensures downstream interoperability and compliance with openBIM requirements.

Understanding IFC Spatial Structure

The IFC spatial structure follows a hierarchical organization that includes:

1. IfcProject: Represents the entire building project.
2. IfcSite: Defines the geographic position of the project. It can include site geometry represented by Mesh elements or Site-type Objects.
3. IfcBuilding: Contains one or more IfcBuildingStorey entities.
4. IfcBuildingStorey: Represents individual story levels within a building.
5. IfcSpace: Represents individual spaces or rooms within a building story.

In Archicad, elements are automatically linked to their home story, and by default, they are listed under the corresponding IfcBuildingStorey in the IFC hierarchy. However, this automatic assignment can be modified to create a more meaningful spatial organization.

A.2.1. Project Information Settings

To establish a solid IFC spatial structure, it's essential to begin with the correct configuration of the Project Info settings. Archicad offers a systematic method for linking project information to IFC entities and attributes, which guarantees that vital metadata is accurately conveyed during IFC exports. This mapping is crucial for ensuring interoperability and adherence to openBIM standards.

1. Core Project Information Mapping: The key project information fields in Archicad are mapped to the primary IFC entities and their attributes as follows:

Archicad Field	IFC Entity	IFC Attribute
Project Name	IfcProject	Name
Project Description	IfcProject	Description
Project ID	IfcProject	GlobalId
Site Name	IfcSite	Name
Site Description	IfcSite	Description
Site ID	IfcSite	GlobalId

Building Name	IfcBuilding	Name
Building Description	IfcBuilding	Description
Building ID	IfcBuilding & IfcBuildingStorey	GlobalId

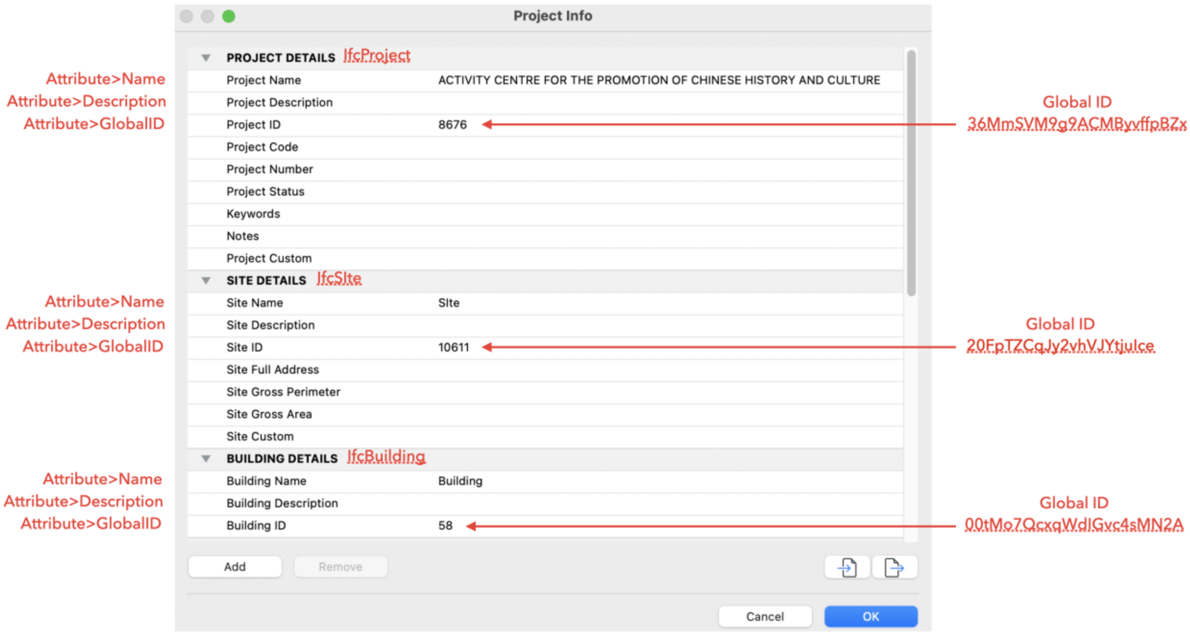
2. Set Project Information

a. Access Project Info:

- Go to File > Info > Project Info in Archicad.

b. Enter Essential Data:

- Project Name (maps to IfcProject Name)
- Project ID (affects IfcProject GlobalId)
- Site Name (maps to IfcSite Name)
- Site ID (affects IfcSite GlobalId)
- Building Name (maps to IfcBuilding Name)
- Building ID (affects IfcBuilding GlobalId)



A.2.2. Customizing GlobalIDs in ArchiCAD

To customize GlobalIDs in Archicad for IFC export, configure Project Info settings and IFC Translator options to control how unique identifiers are assigned to spatial entities. Here's a structured approach:

1. Define GlobalIDs via Project Info

   Access File > Info > Project Info and configure these fields to control GlobalIDs for top-level IFC entities:

   - Project ID: Sets IfcProject.GlobalId.
   - Site ID: Sets IfcSite.GlobalId.
   - Building ID: Sets IfcBuilding.GlobalId.

Scenario	Outcome
Same keyword in Project/Site IDs	IfcProject and IfcSite share the same GlobalID.
Unique Building ID	Each IfcBuilding receives a distinct GlobalID.
Story IDs	Derived from Building ID + Story Number (ensure consistent numbering).

A.2.3. Story Configurations

Setting Up Stories During IFC Export in Archicad

In Archicad, IfcBuildingStorey entities are generated directly from the project's Story Settings. Each story (level) you define in the Archicad project automatically becomes an IfcBuildingStorey in the exported IFC model. The following process occurs:

- Story Creation: When you define a story in Archicad (with a name and elevation), this information is mapped to an IfcBuildingStorey entity.
- Story Naming: The name you assign to the story in Archicad becomes the Name attribute of the IfcBuildingStorey.
- Story Elevation: The elevation value is mapped to the Elevation attribute of IfcBuildingStorey.

Here's a step-by-step guide to ensure your stories are correctly structured for IFC export:

1. Define and Organize Stories in Archicad

   - Create Stories as Logical Building Floors:
     In Archicad, each story represents a logical building floor and will be exported as an IfcBuildingStorey.

- Go to Design > Story Settings to add, name, and set the elevation for each story.
- Ensure story names and elevations match the project's requirements and are coordinated with other disciplines.

2. Story Naming Conventions:
   - Use clear, consistent naming (e.g., "01_Ground Floor", "02_First Floor").
   - Add number prefixes to story names to maintain correct order in platforms like Revit, which sort stories alphabetically.
3. Include Reference Geometry:
   - Only stories containing geometry (elements or zones) will be exported.
   - If a story must appear in the IFC but has no model elements, place a reference object (like a slab or zone) on that story.

A.2.4. Zone Configurations

Setting Up Zones During IFC Export in Archicad

To ensure that zones (which become IfcSpace entities in IFC) are correctly structured and classified during IFC export from Archicad, follow these steps:

1. Model Zones Properly in Archicad
   - Use the Zone Tool to create all spaces/rooms in your project.
   - Assign meaningful names, numbers, and categories to each zone.
   - For best compatibility (especially with Revit), set the subfloor thickness of zones to 0mm to avoid import errors related to enclosure
2. Organize Zones Within the Spatial Hierarchy
   - Each zone is automatically associated with its home story, which maps to IfcBuildingStorey in the IFC hierarchy.
   - The overall spatial structure exported is:
IfcProject → IfcSite → IfcBuilding → IfcBuildingStorey → IfcSpace (Zone)
3. Configure IFC Export Options for Zones

   When exporting zones (Archicad Zones → IFC IfcSpace) via IFC, the IFC Translator's settings are critical to ensure all spatial, classification, and containment data are included and correctly mapped. Here's what you should configure:

   a. Data Conversion Preset

To configure your export settings, navigate to File > Interoperability > IFC > IFC Translators. Select your desired export translator and open the Data Conversion preset (three-dot button).

Zone Categories

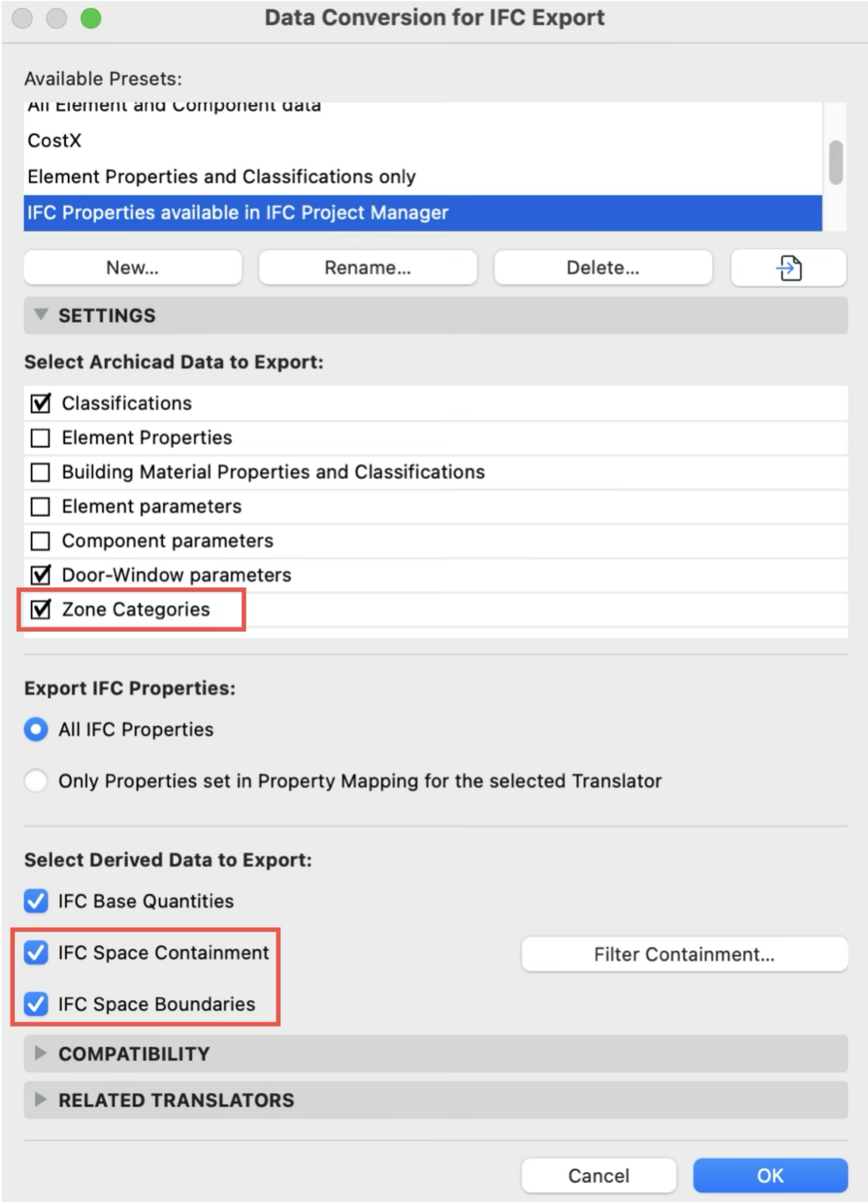
- Enable this option to export the Zone Categories (Code and Name) of Archicad Zones as IFC Space Classification Reference data (ItemReference and Name). This helps with classification and downstream use in FM or analysis tools.

IFC Space Containment

- Enable this option to export data defining the relationship between Archicad Zones and the objects they contain. This is essential for facility management and space allocation workflows.
- You can further filter which elements are included in containment by clicking Filter Containment and limiting by IFC Element Type.

IFC Space Boundaries

- Enable this option to export the geometry data of Archicad Zones, including their relationships (IfcRelSpaceBoundary) with bounding elements (walls, slabs, etc.). This is crucial for energy analysis and thermal calculation applications.



A.2.5. IFC Project Manager

The IFC Project Manager in Archicad is a specialized tool for reviewing, editing, and organizing the spatial structure of your BIM model for IFC export. It provides a hierarchical, tree-based overview of the IFC entities and allows you to customize how elements are grouped and related within the IFC schema, ensuring compliance with openBIM and project-specific requirements.

1. Set Up Which IFC Translator to Use with the IFC Project Manager in Archicad

   The IFC Project Manager in Archicad displays and manages IFC data (attributes, properties, assignments, spatial structure) based on the settings of the currently active Preview IFC Translator. The Preview Translator determines which IFC types, properties, and mapping rules are visible and editable in the Project Manager and other IFC-related dialogs.

   a. Setting the Preview Translator

(i) Open the IFC Translators Dialog: Navigate to File > Interoperability > IFC > IFC Translators.

(ii) Select the Desired Export Translator: In the list on the left, choose the translator you wish to designate as the Preview Translator. Then, click the Set Preview button at the bottom of the dialog.

Once selected, the chosen translator will be highlighted with a blue Archicad icon, indicating that it is now the active Preview Translator.

2. Key Functions of the IFC Project Manager for Spatial Structure Management

a. View and Navigate the IFC Hierarchy:

- Go to File > Interoperability > IFC > IFC Project Manager.
- The left pane displays the full spatial structure (IfcProject > IfcSite > IfcBuilding > IfcBuildingStorey > IfcSpace), listing all IFC entities generated from your Archicad project. You can easily navigate between the IFC model and the native Archicad elements.
- Select any element in the hierarchy to view its properties and attributes. This can include IFC type, classification, properties and other relevant data.

b. Reassign Elements to Spatial Entities:

- By default, Archicad elements are assigned to their home story, appearing under the corresponding IfcBuildingStorey.
- The IFC Project Manager allows you to override this by dragging elements directly to IfcSite or IfcBuilding.
- This is essential for site context elements (e.g., trees, roads, neighboring buildings) that should not be tied to a specific story.

c. Create and Manage Assignments:

- You can create IFC Assignments (e.g., IfcZone, IfcSystem) to group spaces or elements logically. For example, group all IfcSpaces (zones) into an IfcZone for functional or operational purposes. Assignments can be created and managed directly in the IFC Project Manager.
- To create an IFC Assignment
  - Click the "New" button to create an IFC Assignment (e.g., IFC Zone or System).
  - Name and configure the assignment, then drag relevant elements into the assignment's relation folder.
  - Only compatible entities can be grouped (e.g., only IfcSpaces in an IFC Zone)

For comprehensive instructions on the IFC Project Manager, please visit:
https://help.graphisoft.com/AC/28/INT/index.htm#t=_AC28_Help%2F121_IFC%2F121_IFC-10.htm

A.3. Element Categorization

Archicad's IFC exchange workflow is built around a robust system of element categorization and data mapping, ensuring that the non-geometric information are accurately translated between Archicad and other BIM platforms. This process is governed by several key mechanisms: Classifications, IFC Types, Property and Attribute Mapping, and the use of Translators.

A.3.1. Classifications and IFC Type

Archicad's IFC data exchange is fundamentally driven by its classification system, which directly determines both the IFC Type (e.g., IfcWall, IfcSlab) and the IFC Predefined Type (e.g., FLOOR, ROOF, USERDEFINED) assigned to each element during export. This process ensures that exported models are semantically correct and interoperable in openBIM environments.

1. Archicad Classification System:

   Each Archicad element is assigned a classification value from a chosen classification system (such as Archicad Classification, Uniclass, Omniclass, or a custom system) using the *Classification and Properties* panel in the element's settings dialog. This classification is the basis for IFC type mapping during export.

2. Automatic IFC Type Assignment:

   The classification determines which IFC Type the element will be mapped to. For example, a slab classified as "Floor" will be mapped to IfcSlab with the Predefined Type FLOOR; a wall classified as "Curtain Wall" will be mapped to IfcCurtainWall.

3. Type Mapping for Export:

   The IFC Translator's Type Mapping for Export preset defines how each classification is mapped to an IFC Type and Predefined Type. This mapping can be based on:

   - Archicad Tool: (e.g., Wall Tool → IfcWall)
   - Classification Value: (e.g., "Partition" → IfcWall, Predefined Type PARTITIONING).

4. Predefined Types:

   - IFC Predefined Types are subcategories within the main IFC Type, providing more specific semantic meaning (e.g., IfcSlab can have Predefined Types such as FLOOR, ROOF, LANDING, etc.). The translator uses the classification to set this value.

A.3.2. Assigning Classification for IFC Type Mapping

Classification is the key driver for IFC Type mapping during export in Archicad, ensuring your model elements are interpreted correctly in any openBIM workflow. Set the correct classification for each element in the "Classification and Properties" panel of the element's settings dialog.

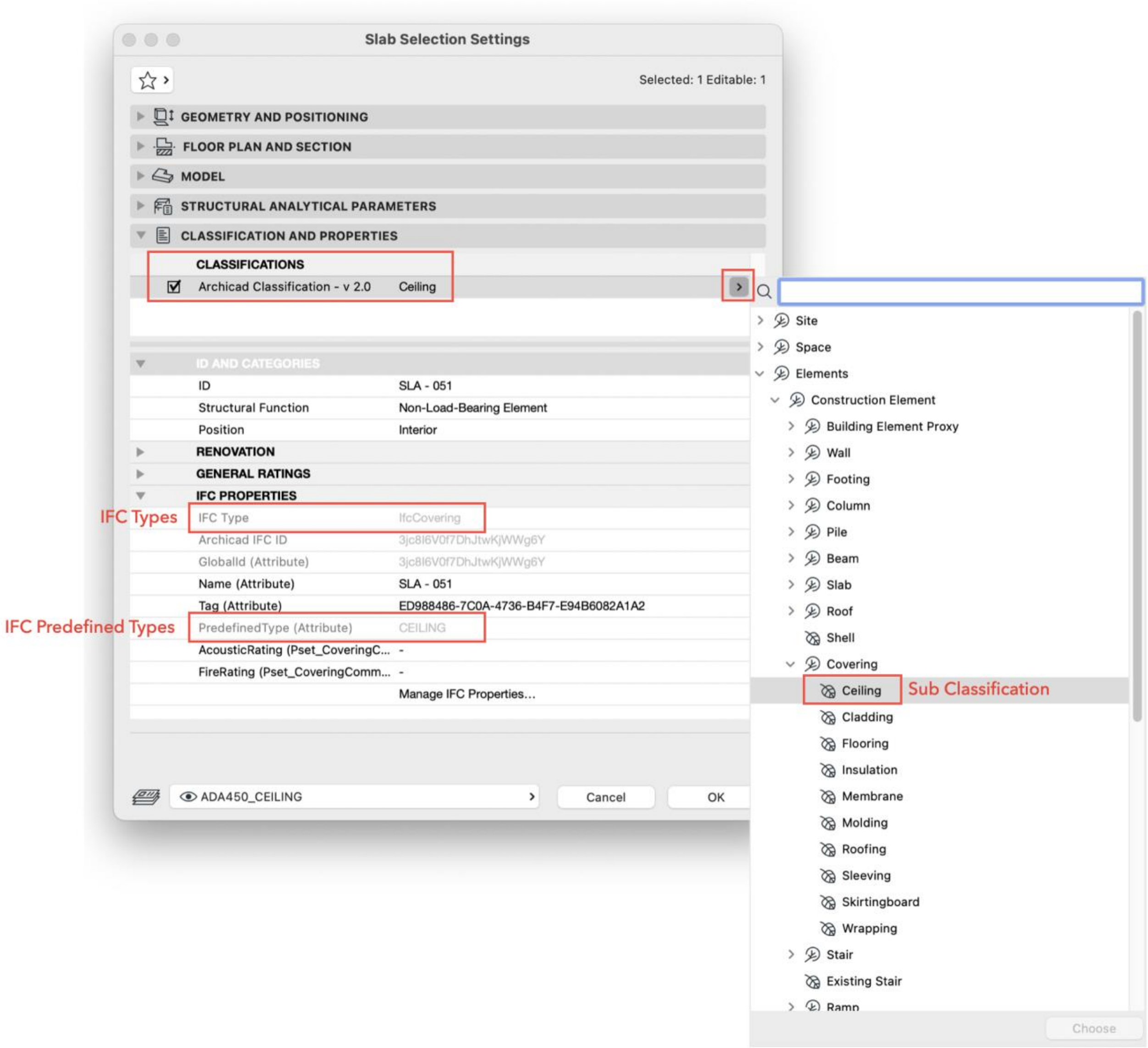
1. Assignment via Element Settings Dialog

   a. Navigate to the 'Classification and Properties 'Panel

- Select the element(s) you wish to classify (e.g., wall, slab, door).
- Open the Element Settings Dialog, locate the “Classification and Properties” panel.
- This panel displays all classification systems available in your project template, such as “Archicad Classification,” Uniclass, Omniclass, or any custom system loaded into the project.

b. Assign the Appropriate Classification

- To change or assign the item’s classification within any of the Systems, click the pop-up arrow at the right.
- For each classification system, select the most accurate classification value that matches the element’s function and intended IFC mapping.
- For example, a slab classified as "Ceiling" will be exported as IfcCovering with the Predefined Type “CEILING”.



c. Confirm the Classification

- After assigning the classification, review the "IFC Properties" section in the same dialog.
- The "IFC Type" and "IFC Predefined Type" fields should now reflect the IFC entity that will be used during export. This mapping is governed by the rules set in the active IFC Translator.

d. How Classification Drives IFC Type Mapping

- The classification you assign is referenced by the IFC Translator's "Type Mapping for Export" setting.

- If “Mapping by Classification” is enabled in the Translator, the element’s classification determines its IFC Type. If no classification is set, the element defaults to a generic IFC type (e.g., IfcBuildingElementProxy).

2. Bulk Assignment via Interactive Schedule

- For multiple elements, use the Interactive Schedule to display the “Classification” field and assign values to many elements at once.

A.3.3. Set Up IFC Type Mapping in the IFC Translator

Setting up IFC Type Mapping in the IFC Translator involves selecting or creating a Translator, choosing the mapping method (by element or classification), assigning the correct IFC Types and Predefined Types for each classification, validating the mapping, and using the Translator during export. This process ensures your Archicad model elements are accurately represented in the IFC schema for downstream use.

1. Access the IFC Translators Dialog

- Go to File > Interoperability > IFC > IFC Translators.
- In the dialog, you’ll see a list of available Translators for both Export and Import.
- Select the Translator for Export you wish to modify or duplicate an existing one to create a custom setup.

2. Open the Type Mapping Preset

- With your Export Translator selected, look to the right panel and find the Type Mapping for Export preset.
- Click the Edit (three-dot) button next to this preset to open the Type Mapping settings. From this preset dialog, click the Map IFC Types for Export to access the mapping definitions.

3. Choose Mapping Method: By Element Type or By Classification

Mapping by Classification is the recommended option for IFC Type assignment in Archicad. This approach allows for detailed and customized IFC Type while the Element Type method simply assigns default IFC Types based on the Archicad tool used (e.g., Wall Tool → IfcWall). The Classification method enables you to map each Archicad Classification to specific IFC Types, Type Products, and Predefined Types, providing much finer control over how your model elements are interpreted in the IFC schema.

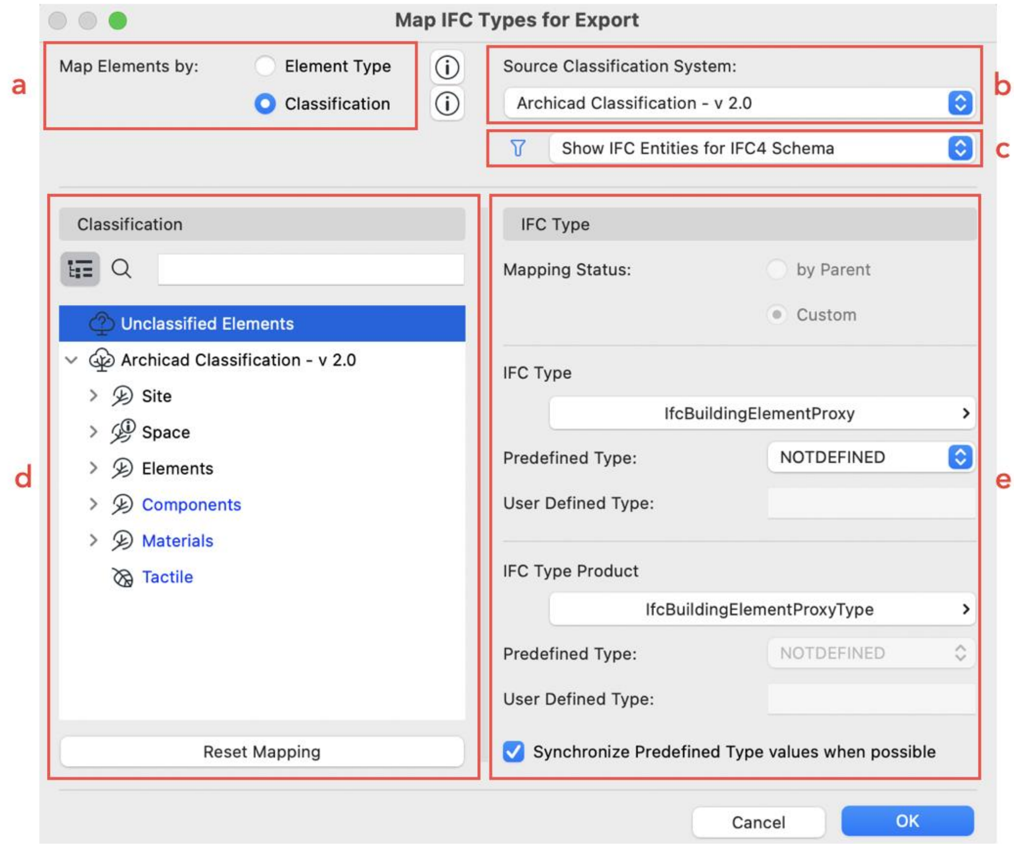
a. Select Mapping by Classification for detailed control.

b. Choose the Classification System used in your project (e.g., Archicad Classification, Uniclass, etc.).

c. Set the Filter to display IFC Type definitions by the IFC4 Schema, the IFC2x3 Schema, or both to ensure compatibility with your target platform.

d. The left side will display all Classification items.

e. The right side shows available IFC Types and Predefined Types.



4. Assign IFC Types and Predefined Types

- For each classification item, select the appropriate IFC Type (e.g., IfcWall, IfcSlab) and, if available, the IFC Predefined Type (e.g., PARTITIONING, FLOOR, ROOF).
- A classification shown in blue means its IFC Type or Predefined Type assignment is inherited from a parent level and can only be changed by editing the parent rule.
- If needed, use the "By Parent" option to inherit the IFC Type from a parent classification.
- For custom requirements, you can assign USERDEFINED as the Predefined Type and specify a custom value.

5. Validate and Save the Mapping

- Check for any warnings or errors in the mapping dialog (e.g., invalid combinations for the selected IFC schema).
- Adjust as needed to resolve issues—Archicad will highlight problematic mappings.

- Save your changes and close the dialog.

6. Use the Translator for Export

- When exporting your model (File > Save As > IFC), select your configured Translator.
- The Type Mapping rules will be applied, ensuring elements are exported with the correct IFC Types and Predefined Types.

For more detailed instructions on Type Mapping for IFC Export, refer to the https://help.graphisoft.com/AC/28/INT/#t=_AC28_Help%2F121_IFC%2F121_IFC-35.htm

A.4. Properties

Properties play a vital role in organizing and facilitating data exchange between various software systems. Archicad offers robust tools for managing these properties during the export of IFC files within an OpenBIM workflow. This process is primarily governed by the IFC Translator and the IFC Project Manager, which enable precise mapping of Archicad elements to IFC entities and properties. These tools provide both pre-defined mapping options and extensive customization capabilities, ensuring flexibility and accuracy in data management.

A.4.1. Property Mapping for IFC Export

Property mapping in Archicad ensures that element properties (both native and custom) are accurately translated into IFC-compliant data during export. This process is managed via IFC Translators, which define how Archicad properties map to IFC schema attributes, enabling interoperability with other BIM platforms. Below is a detailed breakdown of the workflow and key considerations:

1. Accessing Property Mapping Settings

    - Navigate to File > Interoperability > IFC Translators and select a Translator for Export.
    - Open the Property Mapping preset (three-dot button) and click Map IFC Properties for Export to access the mapping interface.

2. Mapping Rules Structure

    - IFC Entities Tree: The left panel lists all IFC entities (e.g., IfcWall, IfcDoor, IfcBuilding). You select the relevant entity to which you want to assign or map properties.
    - IFC Properties List: For each IFC entity, you see a list of available IFC properties and property sets. Properties in bold indicate data already assigned, while blue text shows properties inherited from parent entities.

      For more detailed instructions on Property Mapping, refer to https://help.graphisoft.com/AC/28/INT/#t=_AC28_Help%2F121_IFC%2F121_IFC-38.htm

    - Mapping Rules: Each mapping rule links an Archicad property or parameter to a specific IFC property. You can:
        - Map Archicad properties (from the Property Manager)
        - Map element parameters (such as width, height, fire rating)
        - Map library part parameter (e.g., gs_list_cost)
        - Concatenate multiple fields (e.g., static text + property + parameter)
        - Assign static text or values

3. Creating and Editing Mapping Rules

- Add a New Rule: Select the IFC property, then use the "New Rule" button to define the source Archicad data to export.
- Rule Content: Choose from parameters, properties, library part parameters, or static text. Multiple fields can be concatenated (e.g., for naming conventions).
- Rule Priority: If multiple rules apply, the first valid rule in the list is used. More specific rules (e.g., narrowed by property set or type) take precedence over generic ones.

  For more detailed instructions on Property Mapping, refer to https://help.graphisoft.com/AC/28/INT/#t=_AC28_Help%2F121_IFC%2F121_IFC-39.htm

A.4.2. Unit Conversion

Archicad provides robust and configurable unit conversion during IFC export. This process is managed through the Unit Conversion preset in the IFC Translator for Export, which allows you to set and control how different units are handled for the entire export operation.

1. Centralized Unit Settings

   - In the IFC Translator for Export, the *Unit Conversion preset* allows you to define the export units for all coordinates, geometric parameters, and measure-type IFC properties. This includes length, area, volume, angle, time, and currency units.

2. IFC Schema Compliance

   - When exporting, Archicad recalculates the values of properties and parameters that are unit-dependent to match the selected export units.
   - The exported IFC file will always store measure-type property values in the units defined by the IFC schema. Refer to Table below for Archicad IFC Export Unit Conversion:

Unit Type	Archicad Internal Unit	IFC Exported Unit	Conversion Handled By
Length	Project-defined (e.g., mm)	Meters	IFC Translator preset
Area	Project-defined (e.g., m^2)	Square meters	IFC Translator preset
Volume	Project-defined (e.g., m^3)	Cubic meters	IFC Translator preset
Plane Angle	Degrees	Radians	IFC Translator preset
Time	Project-defined	Seconds	IFC Translator preset
Currency	Project-defined	As set in preset	IFC Translator preset

- Archicad handles the necessary conversion automatically, ensuring that values in the IFC file are schema-compliant, regardless of the working units in the Archicad project.

3. Property Mapping and Unit Conversion

- When mapping Archicad properties to IFC properties, unit conversion is applied automatically to any property whose IFC data type requires a specific unit (e.g., IfcLengthMeasure, IfcAreaMeasure, IfcVolumeMeasure).

- For properties where no unit is defined in the IFC schema (e.g., IfcText, IfcLabel), no conversion is performed.

- Archicad ensures that all unit-dependent data is correctly converted during IFC export according to the Unit Conversion settings in the selected IFC Translator. This guarantees interoperability and accurate data exchange across different BIM platforms and project environments.

A.4.3. Property Mapping Presets in Archicad for IFC Export

Archicad uses Property Mapping Presets within its IFC Translators to control how properties and attributes from the Archicad model are exported as IFC Properties.

For the Property Mapping preset of the IFC Translator for Export defining the assignment of IFC Properties and Attributes when exporting Archicad model to IFC, refer to this link:
https://help.graphisoft.com/AC/28/INT/_AC28_Help/121_IFC/121_IFC-50.htm

A.5. Geometry Management

Managing geometry in Archicad for IFC export during an OpenBIM workflow involves several key steps and considerations to ensure accurate and efficient data exchange. Understanding IFC geometry export options in Archicad is crucial for effective data exchange in OpenBIM workflows.

A.5.1. IFC Geometry Export Options

Boundary Representation (BREP) Geometry

Boundary Representation (BREP) geometry is a method used to represent 3D objects by defining their boundaries through surfaces and edges. In the context of IFC exports from Archicad, BREP is the default and most precise method for exporting geometry.

1. BREP for IFC Export

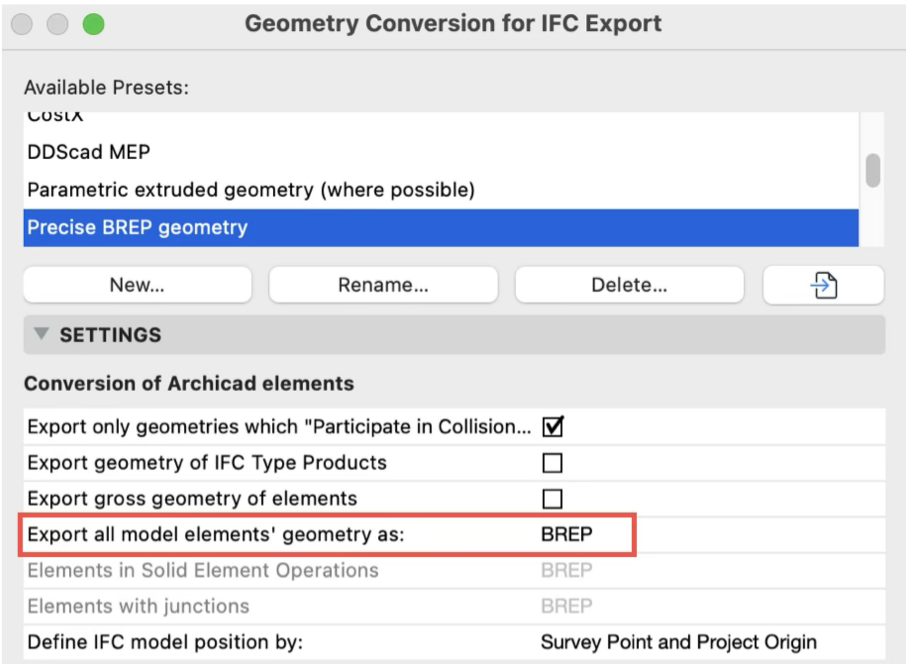
   When exporting IFC files, you can usually find BREP as an option under the geometry conversion settings. Ensure that this option is selected if precise geometry is required.

   a. Access IFC Export Settings:

      - Go to File > Interoperability > IFC > IFC Translators.
      - Choose the appropriate IFC export translator (e.g., IFC2x3 or IFC4).

   b. Geometry Conversion Settings:

      - Open the Geometry Conversion preset within the selected translator.
      - Ensure that BREP Geometry is selected as the geometry export method. This is typically the default setting.



A.5.2. Hierarchical Elements

Hierarchical elements in Archicad, such as Curtain Walls, Stairs, and Railings, can be exported to IFC files in two main ways: as a single element or with their hierarchical structure retained. Here's how to manage these elements during IFC export:

1. Exporting Hierarchical Elements to IFC

   Single Element Export:

   Description: When you choose to export a hierarchical element as a single element, it is exported as one IFC entity. This entity contains the geometry of all its sub-elements, but the hierarchical structure is lost.

   Use Case: This method is useful when you want to simplify the model for visualization or when detailed sub-element data is not required.

   Hierarchical Export (Keep Hierarchy):

   Description: By choosing to keep the hierarchy, the original element's structure is retained. The exported IFC model will contain the main element as an IFC Container, and its sub-elements will be individual IFC elements.

   Use Case: This method is ideal for maintaining detailed geometric and structural information, especially in applications where the hierarchical structure is important, such as in detailed analysis or construction planning.

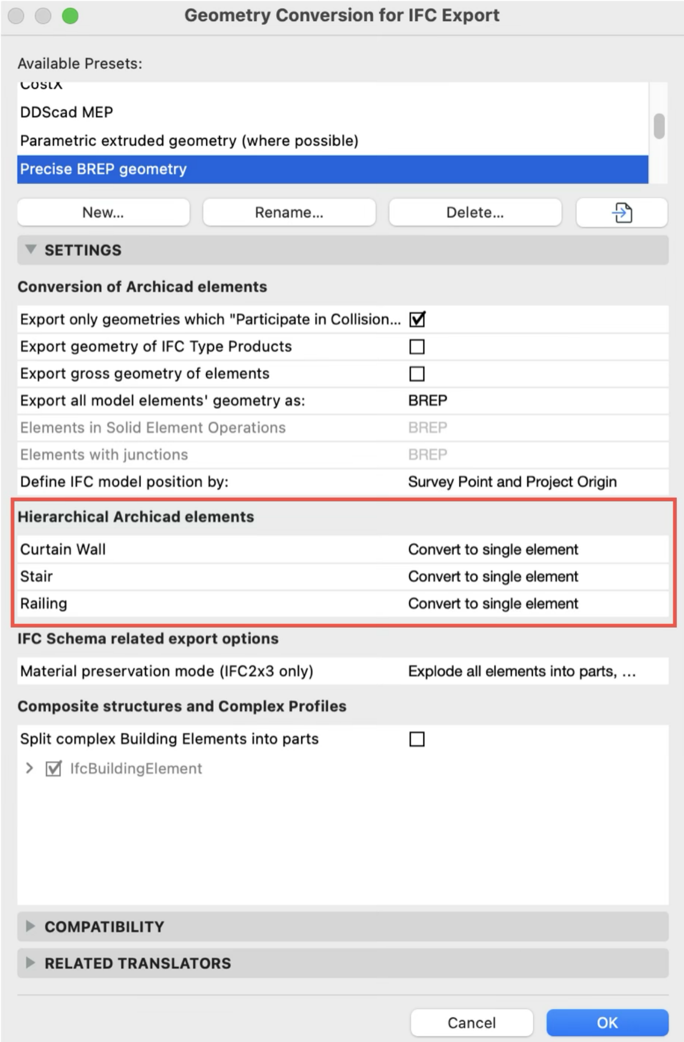
2. Steps to Export Hierarchical Elements

   a. Access IFC Export Settings:

      - Go to `File > Interoperability > IFC > IFC Translators`.
      - Select the appropriate IFC export translator.
      - Open the Geometry Conversion preset (three-dot button)

   b. Configure Hierarchical Elements:

      - In the translator settings, find the section related to hierarchical elements (e.g., Curtain Walls, Stairs, Railings).
      - Choose either Convert to single element or Keep hierarchy based on your project needs.


Geometry Conversion for IFC Export
Available Presets:
CostX
DDScad MEP
Parametric extruded geometry (where possible)
Precise BREP geometry
New...
Rename...
Delete...
SETTINGS
Conversion of Archicad elements
Export only geometries which "Participate in Collision...
Export geometry of IFC Type Products
Export gross geometry of elements
Export all model elements' geometry as:
BREP
Elements in Solid Element Operations
BREP
Elements with junctions
BREP
Define IFC model position by:
Survey Point and Project Origin
Hierarchical Archicad elements
Curtain Wall
Convert to single element
Stair
Convert to single element
Railing
Convert to single element
IFC Schema related export options
Material preservation mode (IFC2x3 only)
Explode all elements into parts, ...
Composite structures and Complex Profiles
Split complex Building Elements into parts
IfcBuildingElement
COMPATIBILITY
RELATED TRANSLATORS
Cancel
OK

A.6. Architectural-Structural Workflow between Archicad and Tekla Structures (IFC-based Model Referencing)

A.6.1. Architectural-Structural Workflow Steps

The exchange of building information models between architectural and structural engineering applications is a critical aspect of modern BIM workflows. This section outlines a process for exchanging structural models between Tekla Structures and ArchiCAD using OpenBIM standards, with a focus on IFC as the data exchange format. This workflow enables seamless collaboration between architects and structural engineers while maintaining data integrity throughout the design process.

Project Workflow for Architectural-Structural Coordination Between Archicad and Tekla Structures (Steps highlighted in **bold** indicate tasks associated with Archicad.)

1. **Export Architectural Model from Archicad as IFC** (Refer to section A.6.2)
2. Import IFC Model into Tekla Structures as Reference
3. Convert or Model Structural Elements Natively in Tekla
4. Export Structural Model from Tekla as IFC
5. **Import Tekla IFC Model into Archicad** (Refer to section A.6.3)
6. **Update Architectural Model in Archicad Based on Structural Input** (Refer to section A.6.4)
7. Re-export Revised Architectural Model from Archicad as IFC
8. Import Updated IFC Model into Tekla Structures
9. Update Structural Model in Tekla
10. Export Updated Structural Model from Tekla as IFC
11. **Import and Compare Updated Structural IFC in Archicad Using Change Detection** (Refer to section A.6.5)
12. Iterate Export-Import-Review Cycles Until Coordination Is Finalized

A.6.1. Export Architectural Model from Archicad as IFC

Exporting an ArchiCAD model for Tekla Structures starts by isolating the relevant model elements and its metadata, then capturing that filtered state in purpose-built 3D views, and finally automating the IFC export via ArchiCAD's Publisher. By combining layer combinations with View Settings and Publisher Sets, you ensure a clean, repeatable workflow that delivers only the necessary geometry and data for the structural engineer.

1. Layer Management
   - Create a Layer Combination that includes all relevant architectural and structural layers.

- Use Layer Intersection Groups to control element interactions (e.g., ensure structural cores intersect correctly with finishes).

2. View Settings

- Model View Options (MVO): Set detail level of library parts, doors, windows, curtain walls, stairs and railing to "Schematic" to reduce the size of the IFC file. Only set the detail level to "Full" if the extra detail of these elements is relevant to the Structural Engineers for design purposes.

- Graphic Overrides: Disable graphic overrides unless specific visual data (e.g., fire compartments) must be retained.

- Partial Structure Display: Set the view to "Entire Model" or "Without Finish" whichever is more appropriate.

- Renovation Filters: If applicable, use a "New Construction" renovation filter to exclude demolished or existing elements from the export.

3. Defining the Export Views

- In the View Map, duplicate an empty 3D view and apply layer combinations and view settings defined in the previous step.

- Rename the view and save.

4. Configuring Publisher Sets

- Add a new Publisher Set and name it.

- Insert the previously created 3D view into the Publisher Set. This ensures only the filtered and properly displayed model is exported.

- In the settings for the view within the Publisher Set:

  - Set the export format to IFC.

  - Select the appropriate IFC Translator (e.g., "Tekla Structures Export")

- In the Publisher palette, select the publisher set and click Publish.

A.6.2. Importing Tekla Structural IFC Models into Archicad

1. Importing Method Options

   a. Archicad provides three primary methods for importing IFC files:

      - Open: This method creates a new Archicad project directly from the IFC file. However, it's not recommended for reference or federation purposes.

- Merge: This method inserts the IFC model directly into your active project as native Archicad elements. However, it's not recommended for iterative workflows because merged elements lose their IFC GUIDs, which can complicate change detection.

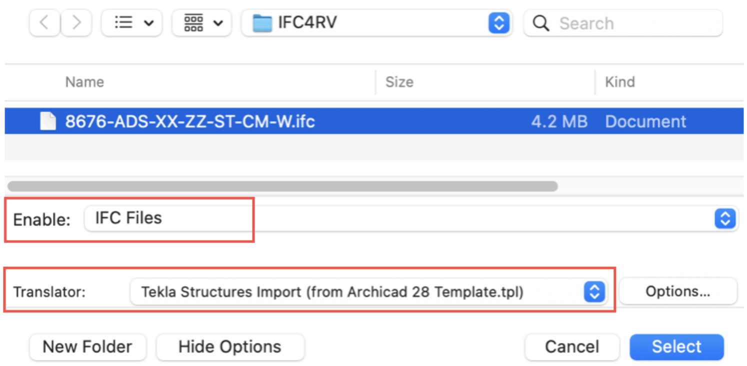
- Hotlink: This method links the IFC as a module, allowing updates and management as an external reference. It's recommended for iterative workflows.

2. Placing the Hotlink Module

a. Select Hotlink file and IFC Translator

- Go to the menu: File > External Content > Place Hotlink.

- In the dialog, Select Select Module > New Module to choose IFC File as the hotlink source and select the appropriate IFC Import Translator (i.e., Tekla Structures Import).



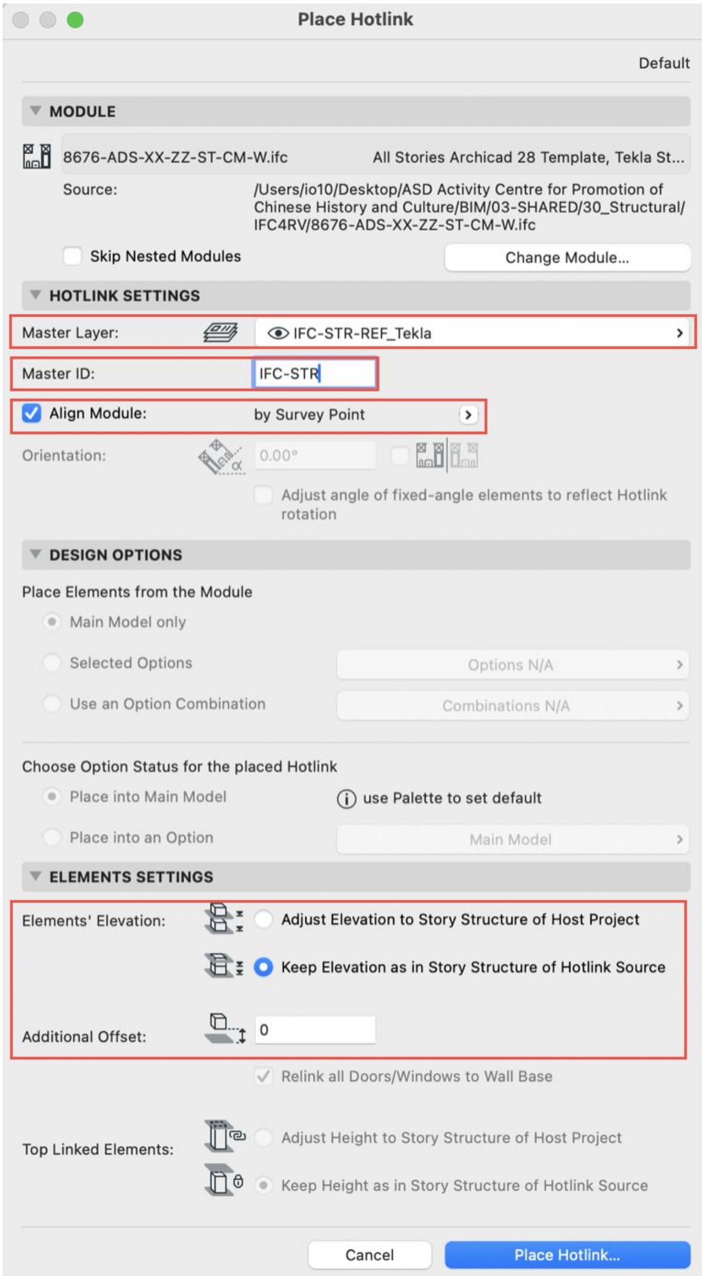
b. Configure Place Hotlink Settings

- Master Layer: Assign a dedicated layer for the hotlinked module (e.g., "IFC-STR-REF_Tekla") with a unique intersection group to prevent clashes.

- Master ID: Set a unique identifier for the hotlink instance.

- Align Module: This setting ensures that the referenced model is accurately positioned in relation to your current project, based on your IFC configuration. Utilize the Project Origin option when both models share the same origin and do not require georeferencing. Conversely, select the Survey Point option when the models are configured with real-world coordinates or geolocation.

- Elevation Settings: Select Keep Elevation as in Story Structure of Hotlink Source and additional offset to 0 to maintain story alignment.



c. Library Part Location

- External vs. Embedded Libraries: Choose whether to reference external libraries (recommended) or to embed objects into the project file. Using external libraries keeps your Archicad file lighter.
- Library Path Consistency: Ensure the path to the external library is accessible to all team members.
- When updating a Hotlinked Module, the Linked Library will not be updated. If Linked Library contents have changed, you must update the library manually.
- In Teamwork, the Linked Library option is not available; generated Library Parts are saved to the Embedded Library.

For more detailed instructions on Hotlink IFC File, refer to https://help.graphisoft.com/AC/28/INT/index.htm#t=_AC28_Help%2F121_IFC%2F121_IFC-8.htm

A.6.3. Update Architectural Model in Archicad Based on Structural Input

1. Comparing Models:

Visual Checks: Use graphic overrides and trace reference tools in Archicad to visually compare the architectural and structural models.

- Use Graphic Overrides to apply discipline-specific color coding for easy visual differentiation (e.g., all structural elements in blue).
- Create view sets: one containing the architectural model and another with the imported structural model. Use ArchiCAD's "Trace and Reference" tool to compare floor plans and sections.

A.6.4. Import and Compare Updated Structural IFC in Archicad

1. Updating the Hotlink

- When a revised Tekla IFC is issued:
  - Use Hotlink Manager in Archicad to update the module.
  - Archicad will replace the previous reference, maintaining placement and settings.

2. Managing Model Changes

When an updated structural model is received from Tekla, ArchiCAD provides tools to compare and visualize changes:

a. Change Detection: Use Archicad's "Model Compare" tool to compare new and previous IFC versions, identifying added, changed, or deleted elements based on GUIDs. The Model Compare palette lists model changes and in the Model Compare window with color coding: blue for new elements, red for deleted elements, and purple for modified elements.

For more detailed instructions on Compare Archicad Models After Hotlink Update, refer to https://help.graphisoft.com/AC/28/INT/#t=_AC28_Help%2F081_ModelCompare%2F081_ModelCompare-5.htm

A.7. Architectural-Building Services Workflow between Archicad and REVIT (IFC-based Model Referencing)

A.7.1. Architectural-Building Services Workflow Steps

The exchange of building information models between architectural and building services engineering applications is a critical aspect of modern BIM workflows. This section outlines a process for exchanging building services models between ArchiCAD and REVIT MEP using OpenBIM standards, with a focus on IFC as the data exchange format. This workflow enables seamless collaboration between architects and building services engineers while maintaining data integrity throughout the design process.

Project Workflow for Architectural-Building Services Coordination Between Archicad and REVIT MEP (Steps highlighted in bold indicate tasks associated with Archicad.)

1. Export Architectural Model from Archicad as IFC (Refer to section A.7.2)
2. Import IFC Model into REVIT MEP as Reference
3. Model Building Services in REVIT MEP
4. Export Building Services Model from REVIT MEP as IFC
5. Import REVIT MEP IFC Model into Archicad (Refer to section A.7.3)
6. Run Clash Detection and Modify Architectural Model (Refer to section A.7.4)
7. Create Opening for Building Services (Refer to section A.7.5)
8. Re-export Revised Architectural Model from Archicad as IFC
9. Import Updated IFC Model into REVIT MEP
10. Export Dwgs from Archicad (Refer to section A.7.6)
11. Import Dwgs to REVIT MEP and Use as Underlay for Drawing Production
12. Update Building Service Design in REVIT MEP
13. Re-export Updated Building Service Model from REVIT MEP as IFC
14. Import and Compare Updated Building Services IFC in Archicad Using Change Detection (Refer to section A.7.7)
15. Iterate Export-Import-Review Cycles Until Coordination Is Finalized

This workflow ensures effective coordination between architectural models in Archicad and building services models in Revit MEP using IFC as the primary exchange format. Below are detailed, step-by-step instructions for each phase of the process, aligned with best practices for OpenBIM interoperability and project management.

A.7.2. Export Architectural Model from Archicad as IFC

1. Exporting an ArchiCAD model for REVIT MEP

   The process for exporting an ArchiCAD model for Revit MEP closely follows the workflow outlined in section B.6.2.

   Separate zones used for calculation purposes (e.g., GFA, UFA, UFS, compartments) should be defined on different layers to distinguish them from physical room spaces.

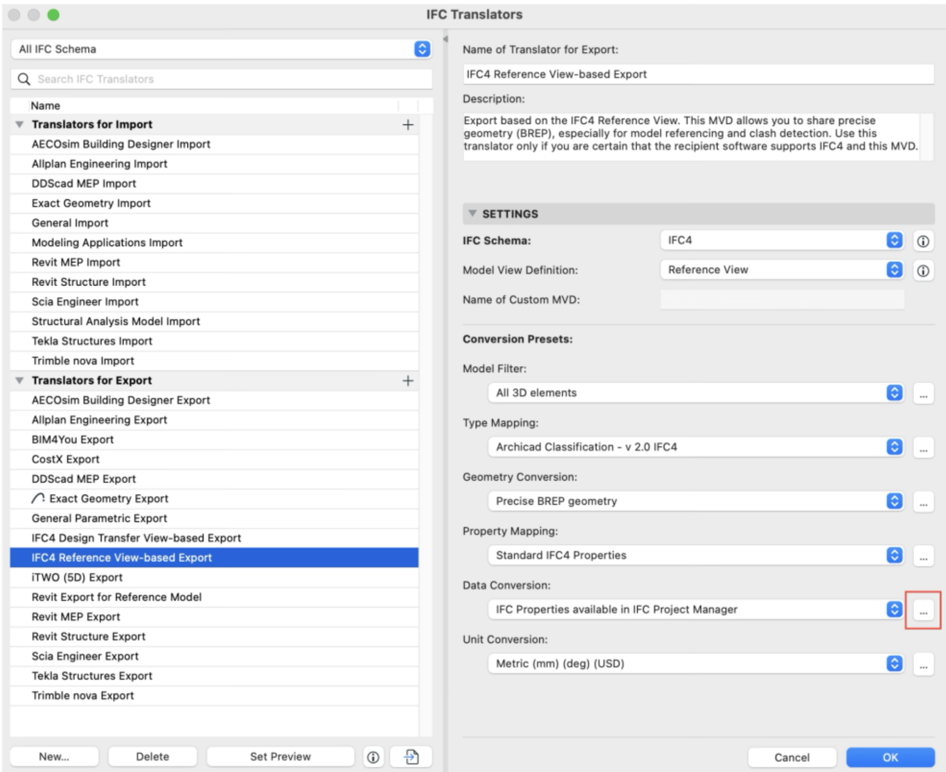
2. Export IfcSpace to REVIT

   a. Configure IFC Translator for Export

      The IFC Translator determines how Archicad elements are interpreted and exported to the IFC format. To modify the settings related to IfcSpaces, please follow these steps:

      (i) Access Data Conversion Settings

      - Navigate to File > Interoperability > IFC > IFC Translators and select the **Data Conversion** preset for the chosen translator.



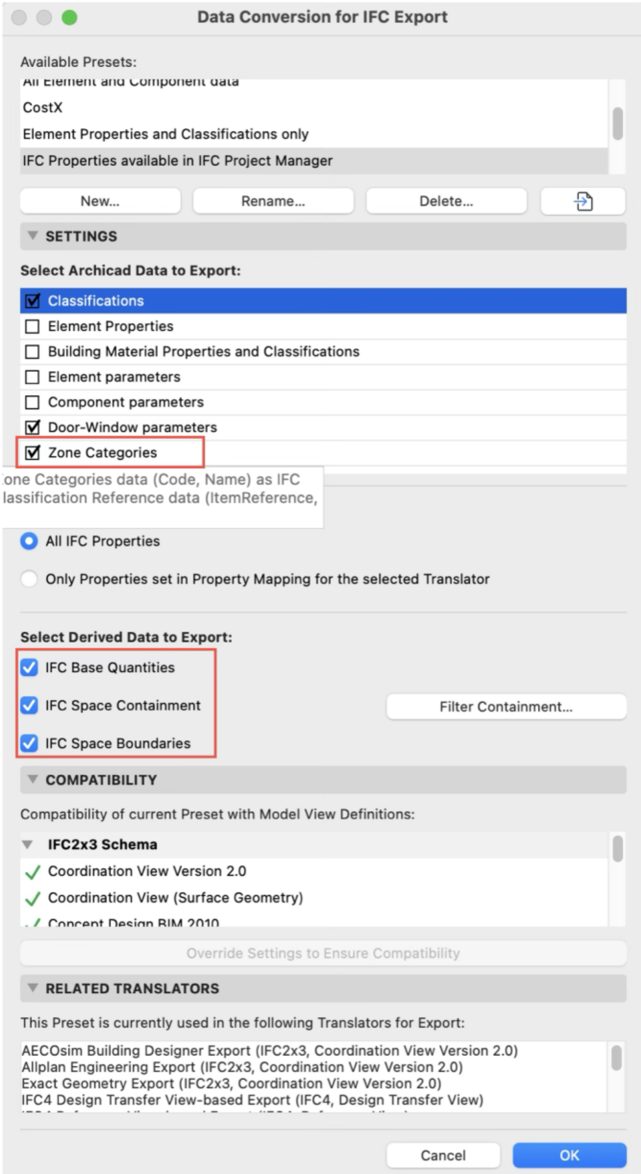
      (ii) Review the Following Options:

      - Zone Categories: This option exports Archicad Zone data, including Code and Name, as IFC Space Classification Reference data.

      - IFC Base Quantities: Enables the export of size, area, and volume parameters for zones, supporting quantity takeoff and cost estimation.

- Space Containment: Ensures that objects within zones are exported while maintaining their spatial containment relationships.

- IFC Space Boundaries: Exports the geometric and logical relationships between spaces (IfcSpace) and their enclosing elements (e.g., walls, slabs), which is particularly beneficial for thermal or energy analysis applications.



(iii) Property Mapping

Open the Property Mapping preset, Map Archicad Zone properties to appropriate IFC properties. For example:

- Zone Name → IfcSpace.Name (Auto)

- Zone Number → IfcSpace.LongName (Auto)

- Custom properties can also be created and mapped if required.

For more detailed instructions on Property Mapping for IFC Export, refer to https://help.graphisoft.com/AC/28/INT/#t=_AC28_Help%2F121_IFC%2F121_IFC-38.htm

A.7.3. Import REVIT MEP IFC Model into Archicad

After obtaining the MEP model from the engineering team, it's essential to integrate it into your architectural model for effective coordination. To manage reference models, import MEP elements onto designated layers for better visibility and intersection management. These elements can be locked to avoid accidental changes. The process for placing the Hotlink Module is similar to a previously described workflow in B.6.3. When configuring the Hotlink Settings for Building Services, assign a specific layer (e.g., "IFC-BSE-REF_REVIT") for each discipline and establish a unique Master ID for each hotlink instance.

A.7.4. Run Clash Detection and Modify Architectural Model

1. Clash Detection:

   Once the MEP IFC Model in imported into the architectural model in Archicad, you need to identify and resolve any spatial conflicts between architectural and MEP elements. Use Archicad's built-in collision detection tools or export both models to a dedicated clash detection software (e.g., Solibri, Navisworks) for comprehensive analysis.

2. Clash Detection Methods in Archicad

   a. Visual Clash Detection

      Archicad provides with several tools that help in visual clash detection workflows:

      - Graphical Overrides: define rules to visually highlight different sets of model elements based on their properties.
      - Minimal Space: doors and some objects have the capacity to show areas that represent a space clearance on plan.
      - Headroom: the stair tool has the option to show a clearance volume required for transit.

   b. Automated Native Collision Detection

      This out of the box function identify collisions between 3D elements in the model. A collision is flagged when two or more elements physically intersect. Checks will be performed between any two groups of elements defined by the filtering criteria.

      To perform collision detection natively in Archicad, follow these steps:-

      (i) Access the Collision Detection Tool:

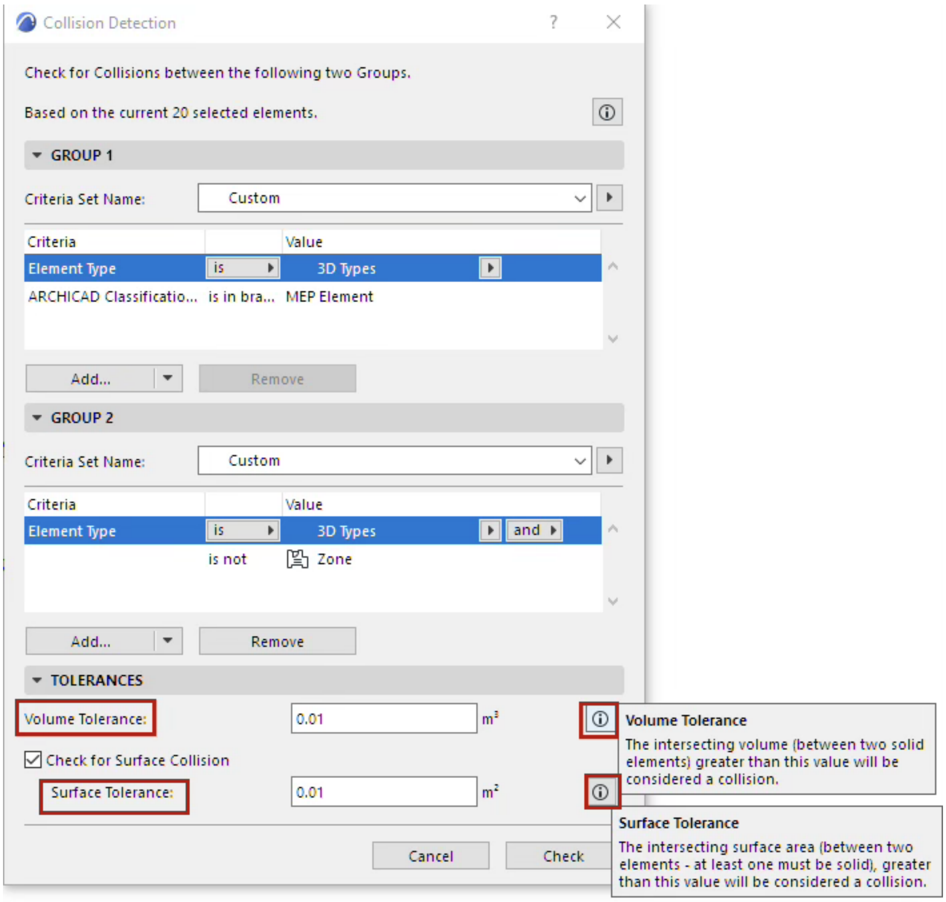
         - Go to the Floor Plan or 3D Window.
         - Navigate to Design > Model Check > Collision Detection

      (ii) Define Collision Groups:

         - In the dialog box, define the criteria for the two groups of 3D elements you want to check for collisions. For example, you can check between MEP elements and Architectural elements.

(iii) Set Tolerances:

- Set Volume and/or Surface Tolerances as needed. This allows you to define what constitutes a collision based on the volume or surface area of the overlapping parts. If the overlap doesn't exceed these tolerances, it won't be considered a collision.



(iv) Run Collision Detection:

- Press Check to run the collision detection. The tool will identify any intersections between the defined groups of elements.

(v) View Results:

- The Collision Detection Report will display any detected collisions. You can click Continue to open the Markup Tools Palette and view the collisions in the model. Each collision is named and assigned a markup style.

The following should be take into account:

- If any elements are selected before starting the check, it will be constrained to those elements. See the top of the dialog for feedback on the selection type.
- Collision Detection is available from the Floor Plan and 3D views only.
- Only 3D geometry on visible Layers is considered.

- 3D Cutting Planes are ignored (the entire model is considered regardless of Cutting Planes).
- Only Building Materials with 'Participates in Collision Detection 'status are considered.
- Elements that are just touching do not count as colliding elements.
- Priority Based Connections do not cause Collisions.
- Solid Element Operations are ignored.

For more detailed instructions on Collision Detection, refer to https://help.graphisoft.com/AC/28/INT/#t=_AC28_Help%2F080_Collaboration%2F080_Collaboration-65.htm

c. External Software Collision Detection

- There are multiple solutions that provide clash detection functionality, some of the most commonly used are: Solibri Model Checker offers model validation based on rules, covering both hard and soft clashes and code compliance.
- The Design Checker (Solibri Inside) plugin uses Solibri online service to run advanced model checks based on IFC export.

3. Model Modification:

a. Address identified clashes in the native Archicad model, adjusting geometry or spatial allocations as needed.

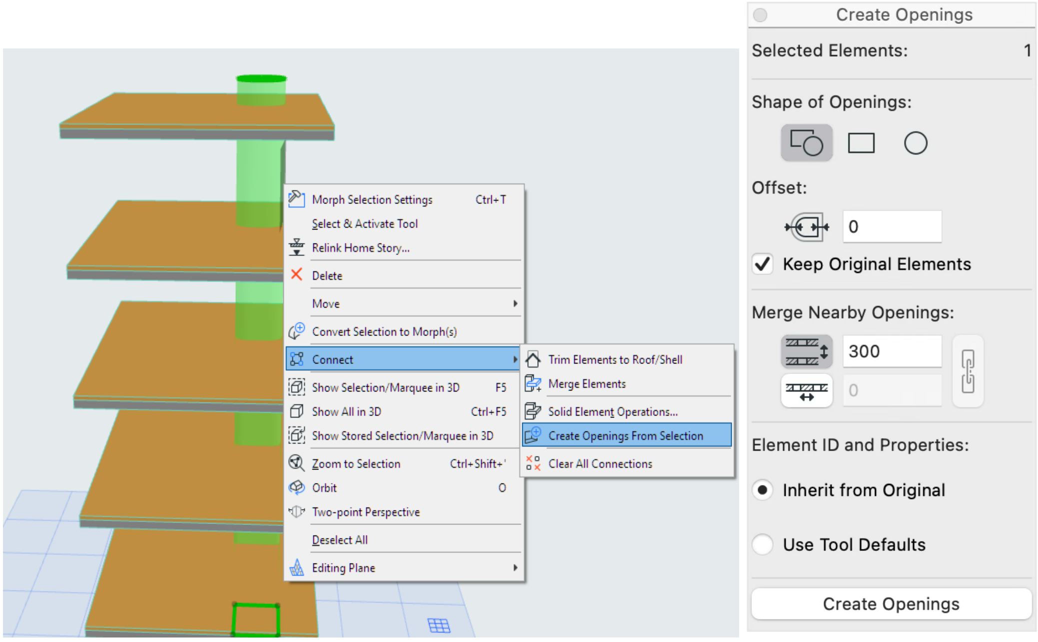
b. Maintain a record of changes and communicate with the MEP team using BIM Collaboration Format (BCF) for issue tracking and resolution.

A.7.5. Create Openings for Building Services

Use Archicad's Opening Tool to create voids for MEP penetrations based on the imported MEP geometry. Openings can be placed manually or generated semi-automatically using collision detection results as guides.

1. Create Openings

- Select the MEP elements (e.g., pipes, ducts) that intersect with walls or floors.
- Use the Create Openings from Selection command from the context menu or from Design > Connect menu
- Set options in the appearing palette, then click Create Openings at the bottom of the palette.



For more detailed instructions on Creating Openings, refer to https://help.graphisoft.com/AC/28/INT/#t=_AC28_Help%2F040_ElementsVB%2F040_ElementsVB-268.htm

2. Placement and Orientation:

Activate the Opening Tool and define its geometry (rectangular, circular, or elliptical) in the Tool Settings. Use the Orientation options (Aligned, Vertical, Horizontal, or Custom) to ensure the opening aligns with the host element's reference plane. For example:

- Aligned: Automatically perpendicular to the host element's face (ideal for slabs and straight walls).
- Custom: Allows manual rotation for angled penetrations.
- Depth Control: Set the opening's Limits to Finite to create niches or recesses for services like electrical conduits or HVAC ducts. Adjust the extrusion depth relative to the host element's reference plane and use the Flip option to control direction.
- Multi-Element Cutting: A single opening can cut through multiple elements (e.g., a suspended ceiling composed of separate slabs). Use the Manage Connections grip to add or remove cut elements post-placement

3. Data Management:

- Assign appropriate classifications and properties to openings (e.g., IfcOpeningElement) to ensure correct export and coordination.
- Link openings to related MEP elements for traceability and future coordination.

A.7.6. Export DWGs for other discipline drawing production

For 3D model exchange workflows, IFC is preferred, but high-quality DWG export remains essential for many coordination tasks (i.e. drawing production)

1. View Setup in Archicad

   - Layer Combinations: Create dedicated layer combinations for Revit export to control visibility of elements like dimensions, annotations, and unneccessary layers.

   - Scale-Specific Views: Define views at appropriate scales (1:50, 1:100) to ensure detail levels match Revit requirements.

2. Export Settings:

   a. Set up Translator for .DWG Export

      DXF-DWG Translators are stored in the Defaults folder of the Archicad program. Thus, translators added or removed in this list will be available for all projects created with the same main Archicad version.

      Note for Teamwork users:

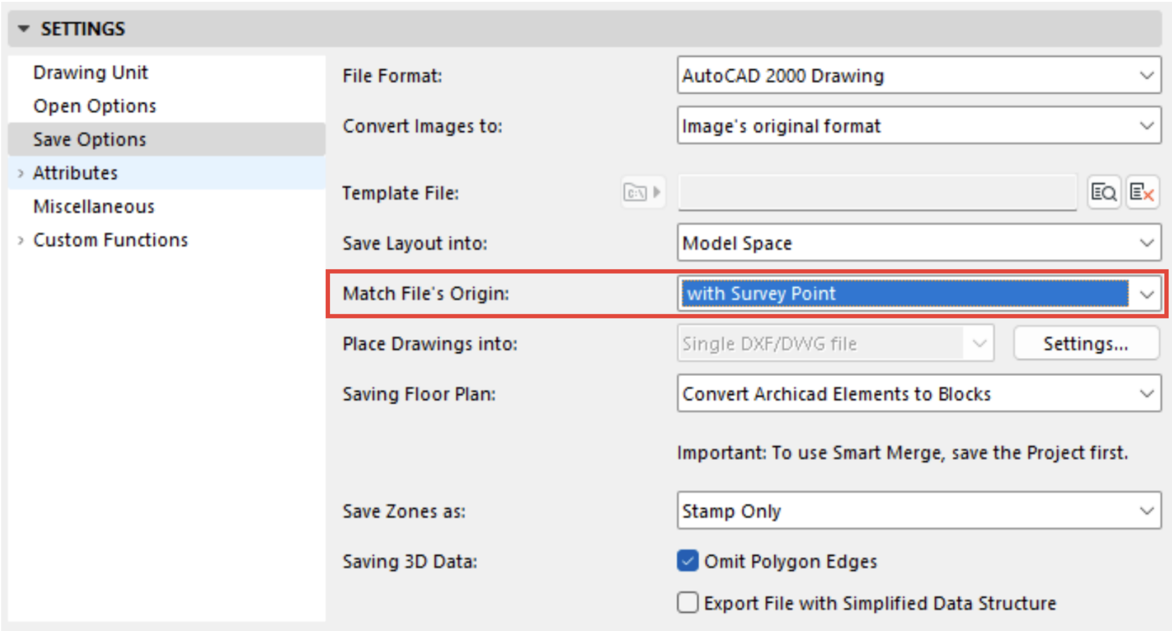
      - Use common Translators ensures that all users have access to the same conversion options.

      - Manage the set of Translators and place them on a server. To use the translators, each user must first add them to their list by using Browse... in the DXF/DWG Translator Setup dialog.

      (i) from File > Interoperability >DXF-DWG > DXF-DWG Translation Setup

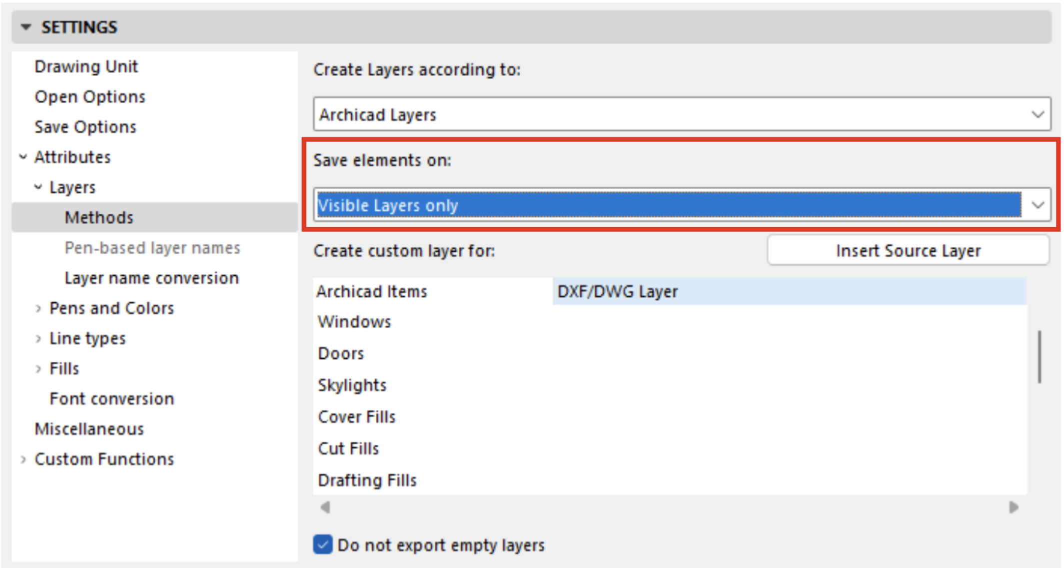
         - Duplicate "03 For as is output" translator and rename it (I.e "03 For export DWGs")

         - Under the SETTINGS subheading, followed by Save Options, ensure that the option Match File's Origin:, is set to with Survey Point as shown below:-



- Note: CAD systems do not have survey point data for geolocation and the standard is to set them up in real world coordinates with the Y axis aligned to north. To perform this translation when exporting into DWG format, select Match File's Origin with Survey Point option in the DWG translator Save Options.

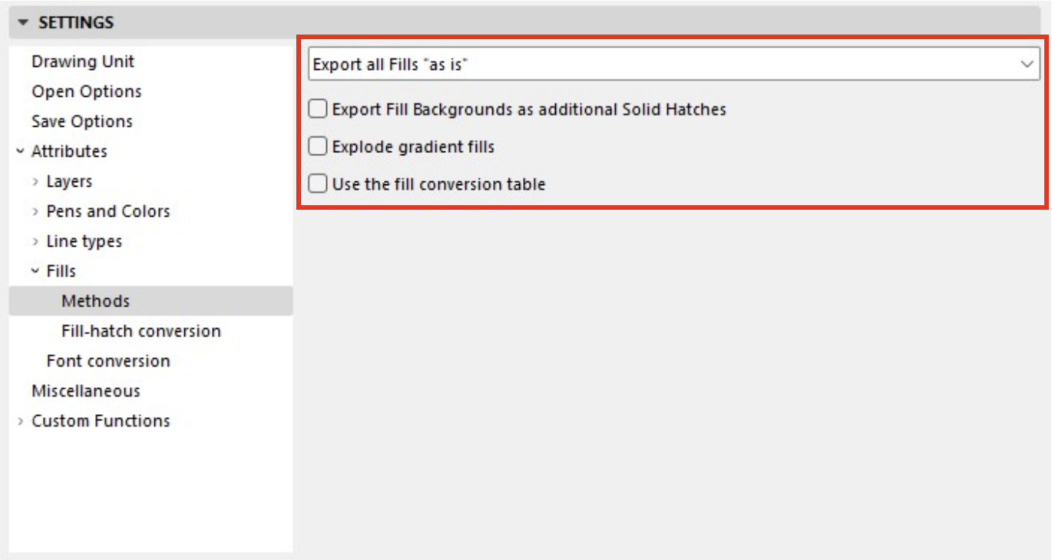
(ii) Under the SETTINGS subheading, followed by Attributes > Layers >Methods, ensure that the Save elements on: option, is set to Visible Layers only as shown below:-



Visible Layers only: Elements on hidden Layers are omitted.

Note: The Visible Layers Only option works for Drawings only if you save the Drawing file separately. If you save a Layout as a DXF/DWG file, all Layers (both visible and hidden) of that Layout's Drawings will be saved.

(iii) Under the SETTINGS subheading, followed by Attributes > Fills >Methods, ensure that the settings as shown below:-

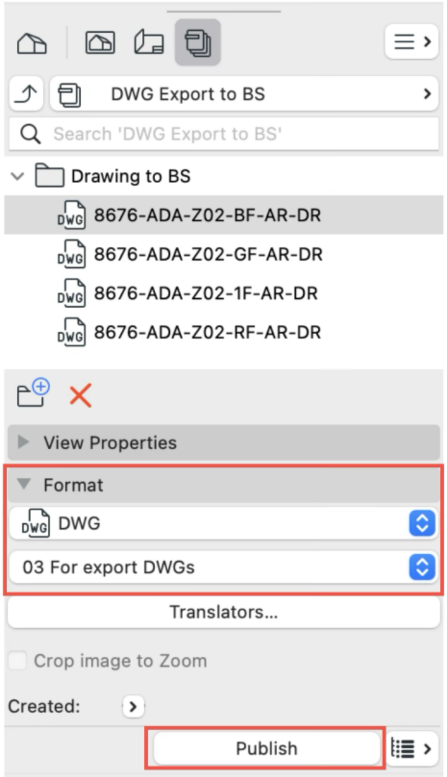


(iv) Save Settings & Close

3. Publisher DWGs

- Create a new Publisher Set
- Add views for export (I.e., all floor plans) to newly created Publisher Set
- Set the file format to DWG and set translator to "03 For export DWGs".

- Run the Publisher set to generate DWGs.



A.7.7. Import and Compare Updated Building Services IFC in Archicad Using Change Detection

Updating the hotlink and managing model changes for the building services IFC follows the same process outlined in section B.6.4.

A.8. BCF Workflow (with Archicad and Trimble Connect)

Archicad seamlessly integrates BCF-based communication through its Issue Management tool, enabling a streamlined, efficient, and flexible collaboration experience that directly incorporates the BIM model. This native BCF support positions Archicad as a viable alternative to solely relying on external platforms like Trimble Connect. By leveraging the BCF workflow, Archicad and Trimble Connect enhance collaboration, allowing users to create, manage, and update issues across various BIM tools effortlessly.

A.8.1. Creating Issues in Trimble Connect

1. Users create issues or topics in Trimble Connect using the BCF Topics feature. This involves describing the issue, assigning it to a responsible party, setting a priority, and adding relevant tags or references.

2. These issues can be linked directly to a BIM model using 3D views and model object references, facilitating model-based communication.

A.8.2. Exporting BCF from Trimble Connect

1. Once issues are created, they can be exported from Trimble Connect as BCF files. This allows for sharing and importing into other BIM tools like Archicad.

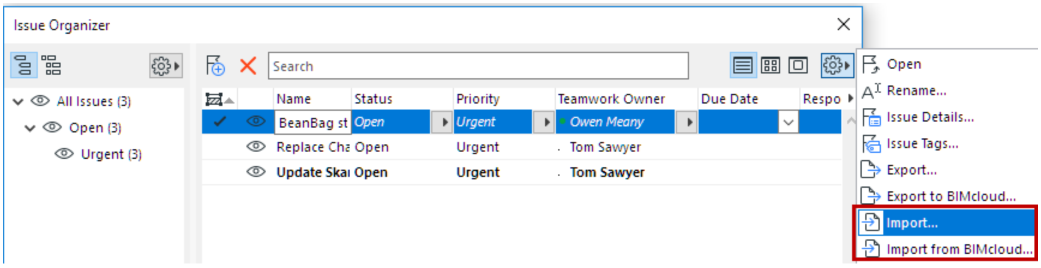
    a. Open BCF Topics Tab and Export Topics (...)

    b. Goto Import & Export History (...) to download BCFs

A.8.3. Importing BCF into Archicad

1. The exported BCF files are then imported into Archicad. This step integrates the issues directly into the Archicad model, allowing users to review and address them within their native BIM environment

    a. Open Issue Organizer Panel and in the Addition Settings button, 'select Import...'



    b. Browse for a file in BCF format. (All BCF formats are supported.)

    c. Click Open.

    d. The Locate IFC Files dialog box appears. Here, you can optionally browse for and locate the IFC files referenced in the BCF entry. The IFC file names are listed: select one and then click Locate to browse for the file location. (IFC files are identified based on their GUID, not their name.)

    e. Click Continue.

f. The imported Issues are added to Issue Organizer. Each is shown in bold type, until you open it.

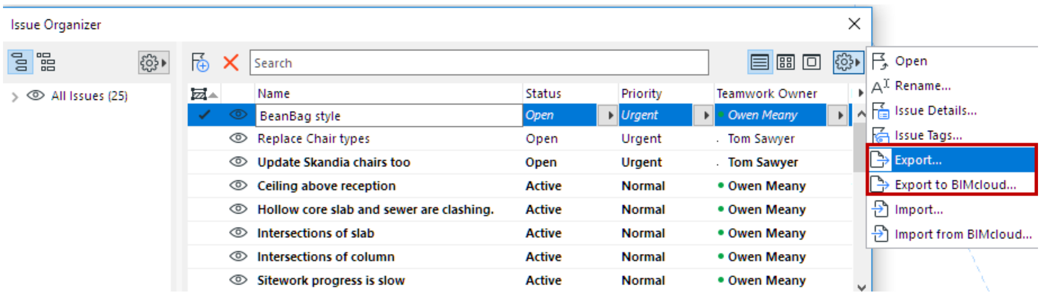
A.8.4. Managing Issues

- Issue Organizer (Document>Issue Organizer): This tool allows users to sort and manage issues based on various criteria like status, priority, and ownership. It provides a comprehensive view of all issues, enabling efficient tracking and management.
- Issue Manager Palette (Document>Issue Manager): Users can edit issue details such as status, priority, type, labels, and descriptions. This helps in categorizing and prioritizing issues effectively.

1. Exporting BCF in Archicad

a. Open Issue Organizer Panel, select the issues for export then click the Addition Settings button, 'select Export…'

b. Save the file in BCF format.



c. choose a BCF version from the Save as type drop-down list.

d. Click **Save**.

e. All the exported Issues are saved in a single .BCF file.

Note: When exporting an Issue in BCF format: if the Issue contains a Floor Plan view, it will generate a 3D axonometry top view, with cutting planes at the top and bottom level of the floor.

A.9. Workflow for IFC SoA Checker (with IDS in Archicad 28)

To meet the Schedule of Requirements established by the end-users, specific custom properties set by ArchSD must be populated within each zones in Archicad. These properties will subsequently be mapped to a set of IFC properties defined during export, creating an IFC file that the SoA Checker will use for verification and compliance assessment.

With an IDS file that includes the SoA data requirements, you can easily import the necessary properties directly into Archicad. Archicad 28 supports the IDS file format, enabling you to load it into Archicad Properties. These properties will then be automatically mapped to the IFC properties during export.

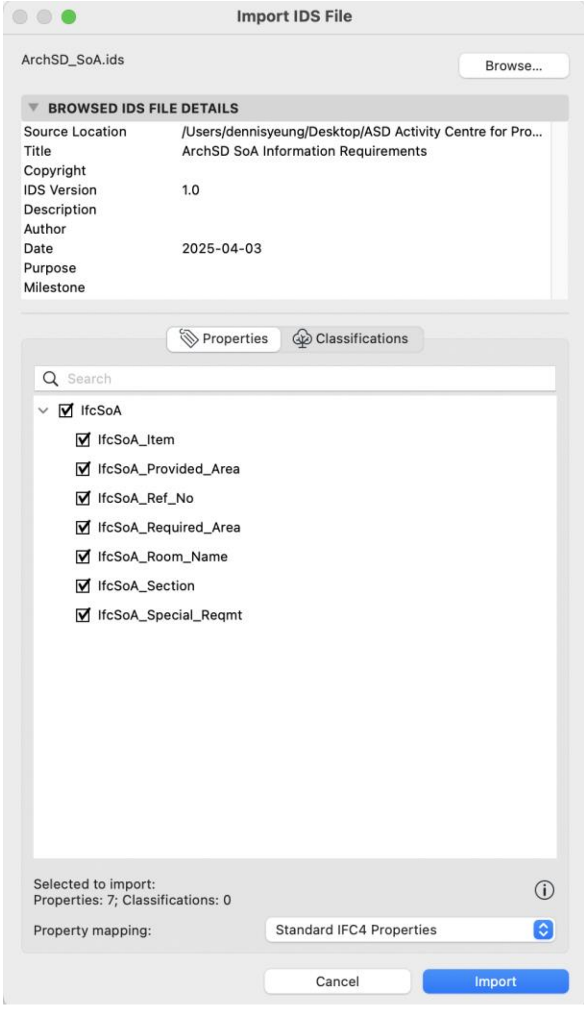
By following the workflow below, you can import a valid IDS file into Archicad and automatically generate classifications and properties, eliminating the need for manual creation. Consequently, when you export to IFC, the exported file will include all the property and classification data that adheres to IDS standards.

A.9.1. Importing an IDS File

1. Access the Import Menu: Navigate to File > Interoperability > IDS > Import IDS and choose the desired IDS file (e.g., in .xml format) from your directory.

2. Review the IDS Content: The Import IDS File dialog will display all valid properties and classifications defined in the selected IDS file. Please note that existing properties and classifications in Archicad will remain unchanged; only new items will be added. Carefully review the available items and select the specific properties (found under the Properties Tab) and classifications (found under the Classification Tab) you wish to import into your project.

3. Select Property Mapping: From the Property Mapping dropdown menu, choose either Standard IFC4 Properties or Standard 2X3 IFC Properties.

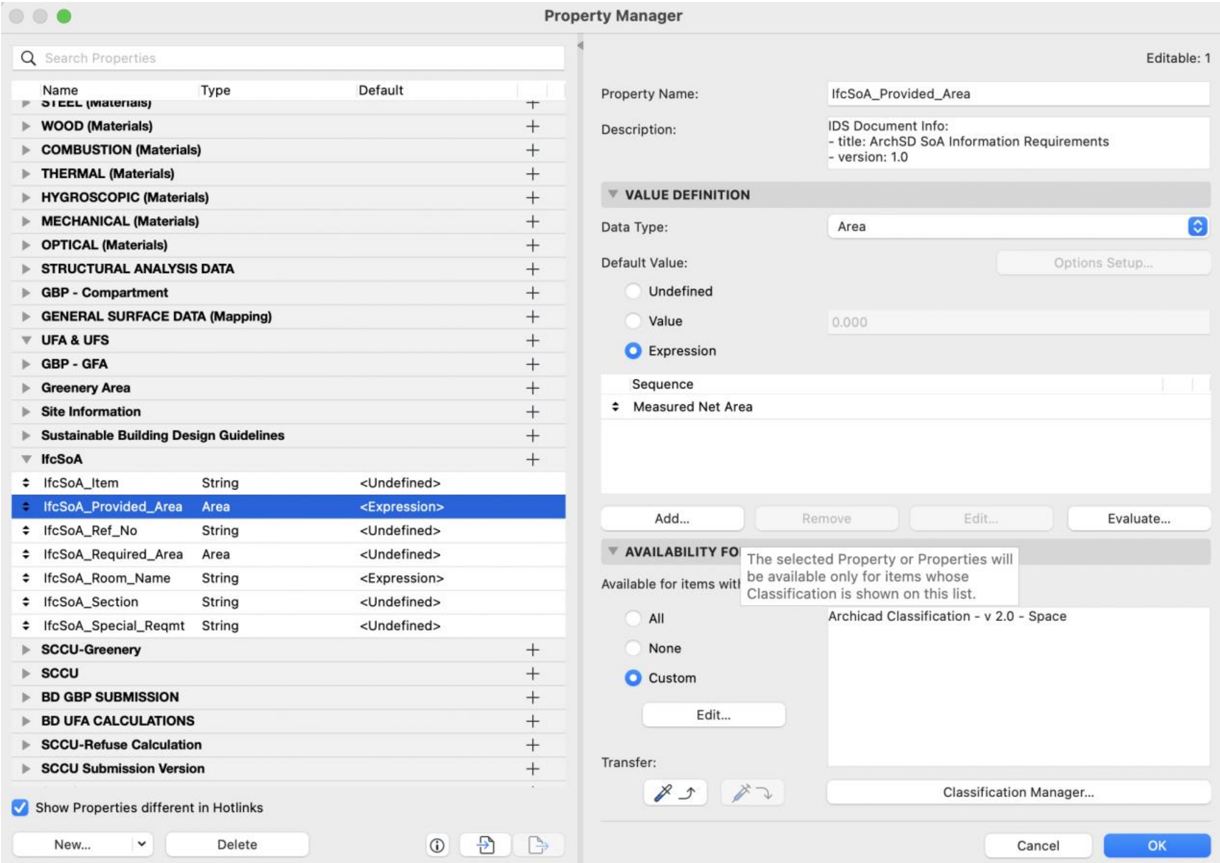


A.9.2. Set IfcSoA_Provided_Area as Measured Net Area

In Archicad, the Measured Net Area parameter refers to the total area of a Zone polygon, excluding the area of bounding walls and other construction elements that intersect or fall within the zone boundary. This calculation is primarily influenced by the construction method and the relationship of elements to the zone.

1. To open the Property Manager dialog, navigate to Options > Property Manager.

2. Next, set the data type of the property to Area. Then, under Default Value Definition, choose Expression and in the expression editor select the parameter, Measured Net Area.



A.9.3. Assign the Property to Zones Classification

1. Still in the Property Manager, look at the left-hand panel under "Availability for Classifications".

2. Click Edit and choose "Space" classification (or any other relevant classification used by your Zone elements).

3. Ensure you check the box next to "Space" to make this property available specifically for Zones.

Your custom property is now successfully created and assigned specifically for use with Zones.

A.9.4. Assigning Values to Individual Zones

1. Select a Zone element in your Archicad model.

2. Open its settings (Ctrl + T) and navigate to "Classification and Properties" tab.

3. Enter the appropriate values for the custom SoA fields. To ensure the SOA Checker functions correctly, make sure to fill in both the IfcSoA_Item and IfcSoA_Required_Area fields.

A.9.5. Validate Data

To ensure that the inputed data comply with IDS requirements, utilize Archicad's Find & Select tool or Interactive Schedules to identify any discrepancies. For any missing or incorrect properties, you can manually adjust them in the zone settings or in the interactive schedules.

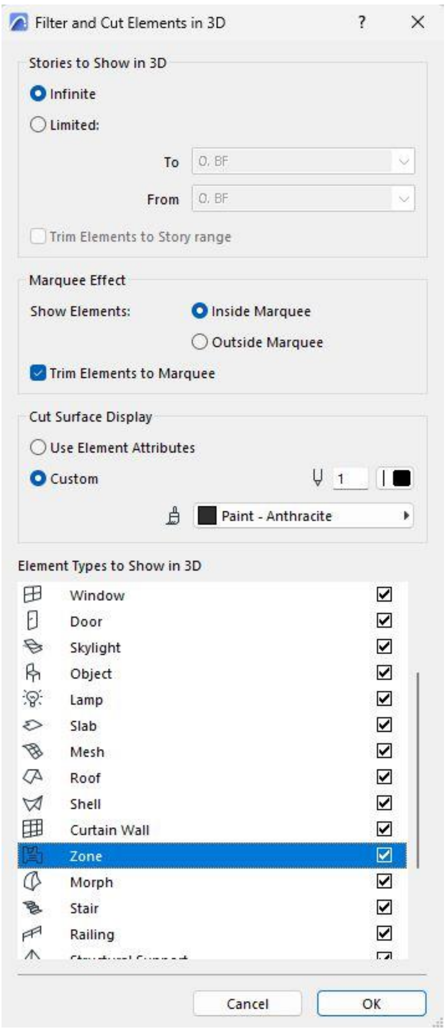
Alternatively, consider using an IDS compliance review application, such as Solibri, for a more comprehensive assessment.

A.9.6. Exporting IFC

To export Zones as IfcSpace entities with custom properties from Archicad, configure the IFC Export Translator settings as follows:

1. Visibility of Zones

    a. To ensure that zones (IfcSpace entities) are visible during IFC export in Archicad, in 3D window and Goto View> Elements in 3D View> Filter and Cut Elements in 3D... (Ctrl+Alt+A) and turn on the "Zones".



    b. Additional, you can ensure the visibility through layers, view settings, and proper translator configurations, to guarantee that zones are correctly included during IFC export.

2. Choose the Correct IFC Translator

    a. Go to File > Save As, choose "IFC Files (.ifc)".

    b. In the "Save As" dialog, select your previously configured IFC Translator from the dropdown menu (this translator should already have your custom property mappings defined).

Upon exporting, your custom Zone properties will appear as mapped IFC properties within each IfcSpace entity in the resulting IFC file.

A.9.7. Workflow with No IDS File

If a IDS is not provided, use the following workflow to add custom-defined zone properties and create mappings to corresponding IFC properties when exported as IfcSpaces.

1. Adding Custom Properties in Archicad

   To add custom properties specifically to Zones in Archicad, follow these detailed steps:

   a. Create a New Property

      (i) Go to Options > Property Manager.

      (ii) In the Property Manager window, click New at the bottom-left corner.

      (iii) Select New Property from the dropdown menu.

   b. Define Your Custom Property

      (i) In the right-hand panel, enter the following property names and respective data types:

Property Name	Data Type
IfcSoA_Item	String
IfcSoA_Provided_Area	Measure Net Area (as an expression)
IfcSoA_Ref_No	String
IfcSoa_Required_Area	Area
IfcSoA_Room_Name	String
IfcSoA_Section	String
IfcSoA_Special Requirement	String

2. Mapping Custom Archicad Properties to IFC Properties

   To ensure your newly created Archicad properties are correctly translated into IFC properties upon export, you must set up a mapping in the IFC Translator:

   a. Property Mapping for Export

      (i) Go to File > Interoperability > IFC > IFC Translators.

      (ii) Select the appropriate IFC Translator and click on the Property Mapping preset.

      (iii) Click on Map IFC Properties for Export to access the mapping definitions

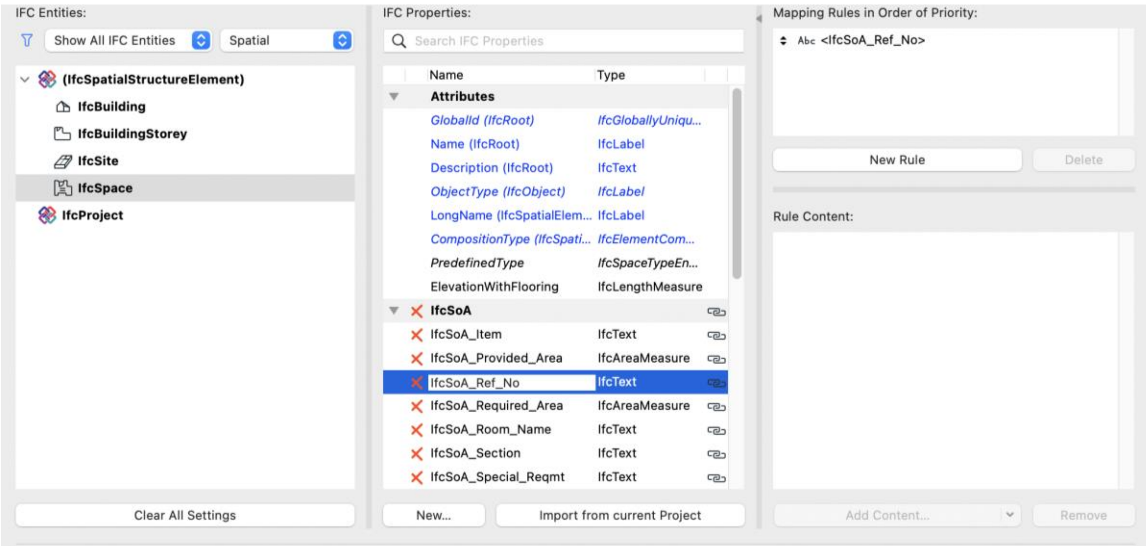
   b. Creating New IFC Properties

To create a new custom IFC property for mapping the custom Schedule of Areas (SoA) archicad property for export to IFC, follow these steps:

(i) Under the **IFC Entities** panel, select **IfcSpace** as the entity type for which you want to create the property.

(ii) Under the IFC Properties panel, Click **New** and in the dialog box, enter a name for your new custom Property Set (i.e., IfcSoA), then enter the **Property Name** and corresponding **Value Type** as per table below:-

Property Name	Value Type	Archicad Property
IfcSoA_Item	ifcText	IfcSoA_Item
IfcSoA_Provided_Area	ifcAreaMeasure	IfcSoA_Provided_Area
IfcSoA_Ref_No	ifcText	IfcSoA_Ref_No
IfcSoA_Required_Area	ifcAreaMeasure	IfcSoA_Required_Area
IfcSoA_Room_Name	ifcText	IfcSoA_Room_Name
IfcSoA_Section	ifcText	IfcSoA_Section
IfcSoA_Special_Requirement	ifcText	IfcSoA_Special_Requirement



c. Creating a New Mapping Rule

(i) In the **Map IFC Properties for Export** dialog, Select the IFC property you want to create a new mapping rules, then click **New Rule**.

(ii) Click Add Content and select the archicad property you want to map from the available properties list (Refer to the above table for the correct mapping). Repeat the process for each custom property that needs mapping.

For more detailed instructions on Creating a Mapping Rule, refer to https://help.graphisoft.com/AC/28/INT/#t=_AC28_Help%2F121_IFC%2F121_IFC-39.htm

Appendix B

B. COMPREHENSIVE GUIDE TO OPENBIM WORKFLOW IN REVIT MEP

To effectively support the openBIM workflow for MEP (Mechanical, Electrical, and Plumbing) design, collaboration, and drawing production, several key considerations must be addressed. MEP using Revit design workflows are inherently dependent on the architectural and structural layouts, which act as the foundation or underlay for MEP systems. These layouts may come from different BIM systems, often introducing challenges in interoperability, coordination, and data exchange.

B.1. Importing IFC from other Authoring tools

B.1.1. Key Considerations Before Importing IFC into Revit

Understand IFC Model Structure:

- IFC models are organized hierarchically with entities like:
- IfcProject: The overall project.
- IfcSite: The project's location/georeferenced origin.
- IfcBuilding: Represents individual buildings.
- IfcBuildingStorey: Represents levels within the building.
- IfcSpace: Defines rooms or spaces.

Purpose of Importing IFC:

- Determine whether the IFC model is being imported:
- As a reference for MEP system design (recommended).
- For conversion into native Revit elements (limited functionality in Revit).

Coordinate System Alignment:

- Ensure the IFC model uses the same Survey Point or Shared Coordinates as the Revit project to avoid alignment issues.

IFC Version Compatibility:

- Revit supports IFC4 (preferred for OpenBIM workflows) and IFC2x3.

Confirm that the architectural and structural teams export models using compatible Model View Definitions (MVDs), such as:

- IFC4 Reference View: For coordination.
- IFC4 Design Transfer View: For editable elements (limited support in Revit).

B.2. Workflow for Importing IFC into Revit MEP

B.2.1. Preparing the Revit MEP Project

1. Set Up Project Coordinates:
2. Ensure the Survey Point and Project Base Point in Revit are aligned with the IFC model's coordinate system.
3. For georeferenced projects (e.g., HK1980 Grid), confirm the shared coordinate system matches across all disciplines.
4. Create a Dedicated Workset:
5. Create a new Workset for the imported IFC model (e.g., "Architectural IFC" or "Structural IFC"). This allows better control over visibility and model management.

B.2.2. Importing the IFC Model

1. Link the IFC Model:
   a. Navigate to Insert > Link IFC.
   b. Select the IFC file provided by the architectural or structural team.
2. In the Link IFC Options dialog, configure the following:
   a. Positioning: Choose Auto – By Shared Coordinates if the IFC model is georeferenced. Otherwise, use Auto – Center to Center.
   b. Open Workset: Assign the IFC link to the newly created workset (e.g., "Architectural IFC").
   c. Verify Import Options:
   d. Ensure the following options are enabled:
   e. Preserve Geometry: Retains the original geometry of the IFC model.
   f. Load Nested Links: Includes nested models within the IFC file (if applicable).
   g. Spatial Structure: Retains the hierarchy of the IFC model (e.g., IfcBuildingStorey).
3. Load the Linked Model:
4. After linking, the IFC model will appear in the Revit project as a reference.

B.2.1. Managing the Linked IFC Model

1. Control Visibility:
   a. Use Visibility/Graphics (VG) settings to control the display of the linked

2. IFC model:
    a. Go to View > Visibility/Graphics (VG).
    b. Navigate to the Imported Categories tab and manage the visibility of specific IFC categories (e.g., IfcWall, IfcSpace).
    c. Apply filters to isolate elements relevant to MEP design.
3. Assign View Templates:
    a. Create and apply View Templates to manage the visibility of the linked
4. IFC model across multiple views:
    a. Example: Hide unnecessary categories like furniture or site elements.
5. Pin the Linked Model:
    a. Pin the linked IFC model to prevent accidental movement or rotation.
    b. Use Linked Model for Coordination:
6. Reference the linked IFC model for placing MEP elements:
    a. Align ducts, pipes, and conduits with architectural walls and structural beams.
    b. Use IfcSpace data from the architectural model for defining HVAC zones.

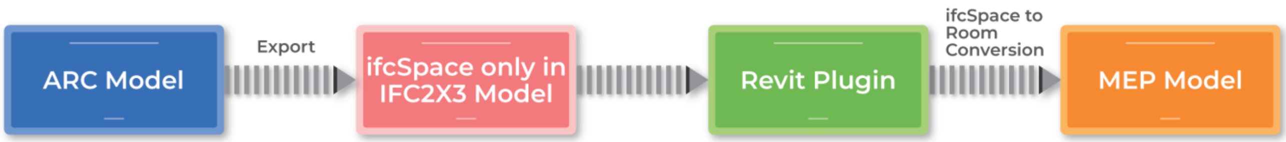
B.3. Best Practices for Importing IFC into Revit MEP

1. Use Linked IFC Models for Reference:
    a. Avoid converting IFC models into native Revit elements unless absolutely necessary.
2. Optimize IFC Files Before Import:
    a. Request clean, filtered IFC files from the architectural and structural teams to minimize file size and complexity.
3. Coordinate Regularly:
    a. Update the linked IFC models periodically as new revisions are provided.
    b. Notify the architectural and structural teams if issues are identified during coordination.
4. Document Import Settings:
    a. Record the import settings (e.g., positioning method, workset assignment) in the project's BIM Execution Plan (BEP).

B.4. Converting IfcSpace to Rooms / Space in Revit

In some cases, a room/ zone created in an architectural model, say archiCAD need to be incorporated into Revit (MEP) for location identification purposes, there are 2 possible ways to do so

Method 1



This workflow describes the process of converting IfcSpace entities from an architectural model (exported in IFC format) into usable Rooms/Spaces in Revit for MEP design purposes. The workflow ensures seamless coordination between disciplines by maintaining spatial data integrity and alignment across platforms.

B.4.1. Exporting the Architectural Model in IFC Format

1. Source: The architectural model, created in software such as ArchiCAD, is exported in IFC2x3 format.
2. Key Element: The exported IFC model must include IfcSpace entities that define rooms and spaces.
3. Checklist for Export:
   a. Ensure IfcSpace entities are properly defined and enclosed by architectural elements (walls, floors, ceilings).
   b. Validate that the exported IFC model aligns with the agreed coordinate system (e.g., Project Base Point or Survey Point).

B.4.2. Verifying IfcSpace Entities in the IFC Model

1. Intermediate Step: The exported IFC file should be reviewed to confirm that it contains spatial data (IfcSpace) with the following attributes:
   a. Name (e.g., Room Name)
   b. Number (e.g., Room Number)
   c. Dimensions (e.g., Area, Volume)
2. Tools for Verification:

a. Use IFC viewers like Solibri Model Viewer or BIMcollab Zoom to check the presence and accuracy of IfcSpace entities.

B.4.3. Importing the IFC Model into Revit with Plugins

1. Revit Plugin: Use a specialized plugin or add-in (e.g., Autodesk IFC Importer or third-party tools) to import the IFC model into Revit.

2. Conversion Process:

   a. The plugin translates IfcSpace entities into Revit-compatible Rooms or Spaces.

   b. Ensure that the plugin maps IfcSpace attributes (Name, Number, Area) to corresponding Revit Room/Space attributes.

3. Settings and Options:

   a. Use the plugin's settings to preserve the spatial hierarchy (e.g., IfcBuildingStorey).

   b. Align imported spatial data with the Revit project's coordinate system.

B.4.4. Converting IfcSpace to Usable Rooms/Spaces in Revit

1. Automatic Conversion: The plugin automates the process of translating IfcSpace into Revit Rooms or Spaces.

2. Manual Adjustments: Post-conversion, validate the following:

   a. Room boundaries align with architectural elements.

   b. Room names and numbers match the architectural model.

   c. Additional attributes, such as HVAC zones or occupancy, are correctly populated.

B.4.5. Utilizing Converted Rooms in the Revit MEP Model

1. Integration with MEP Design:

   a. The converted Rooms/Spaces are used as the basis for defining HVAC zones and other MEP systems.

   b. Revit tools, such as the Zone Tool, can group Rooms into functional zones for HVAC analysis.

   c. Coordination with Other Disciplines:

   d. Use the spatial data to coordinate MEP systems with architectural layouts and structural elements.

B.4.6. Quality Assurance and Coordination

1. Validation: Perform a thorough check of the imported Rooms/Spaces to ensure:

   a. Proper alignment with the architectural model.

b. Integrity of spatial data (no missing or misaligned Rooms).

c. Clash Detection: Use federated models to detect and resolve conflicts between MEP systems, architectural elements, and structural components.

2. Feedback Loop: Share observations with the architectural team if issues in IfcSpace definitions are identified.

Method 2

In IFC4, an add-in can be installed by directly converting IfcSpace to Revit room

https://github.com/agnBIM/ifc2room

1.) Open an empty Revit file (version 2020, 2021, 2022).

Caution: The app deletes all levels (and the associated views) from the underlying Revit file and then creates new levels based on the Ifc file.

2.) Start the program.

3.) Selection of the Ifc file to be read (Double click in text field).

4.) Selection of the Revit view type, which shall be the basis for the rooms.

5.) Press "Start".

B.4.7. 7. Final Output for OpenBIM Workflow

1. The final Revit MEP model, complete with Rooms/Spaces derived from IfcSpace, is shared as an IFC4 Reference View for further coordination with the architectural and structural models.

2. This ensures that all disciplines work with accurate and consistent spatial data within the OpenBIM framework.

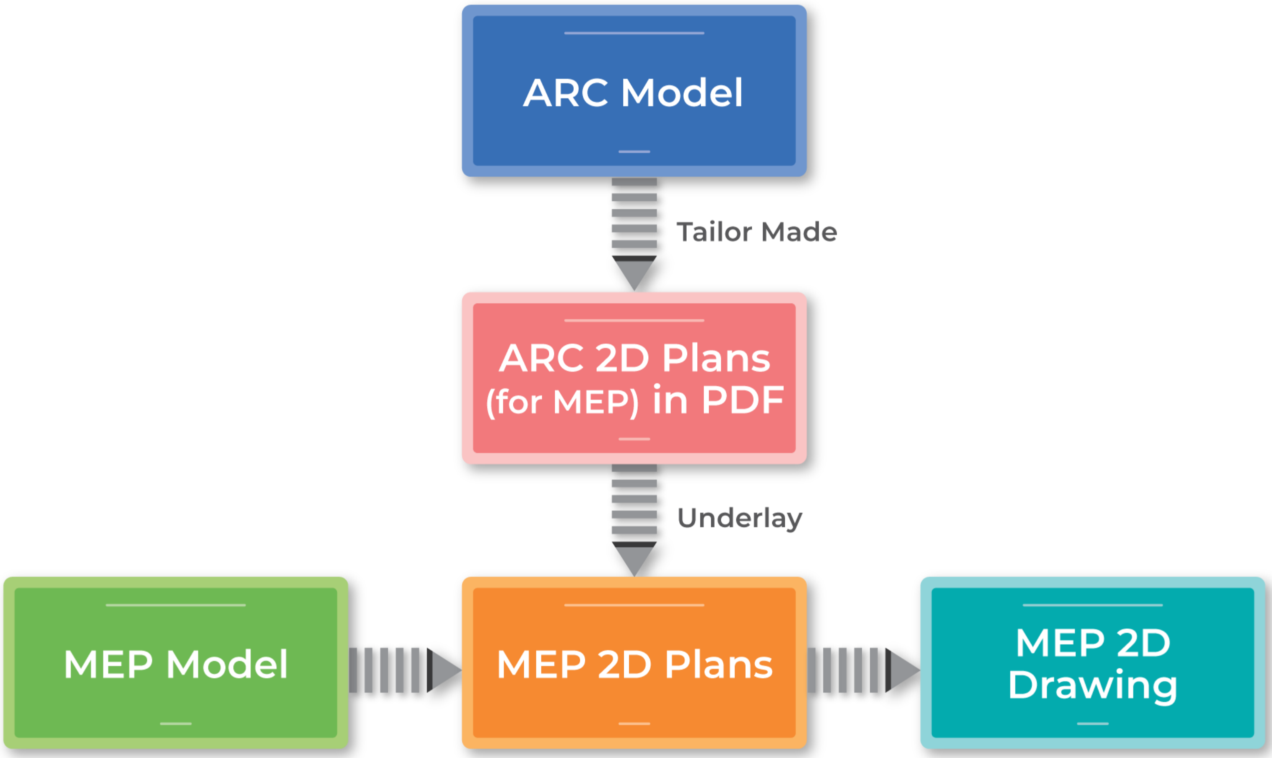
B.5. Architectural plans underlay for Revit MEP drawings

Since MEP drawings do not only indicated MEP layouts, but also Architectural layouts with conventional way of presentation, eg. a sprinkler pipe located at the underside of a stair, the architectural stair needs to be indicated partial dotted line and partial solid line with cut symbol to clearly indicate the stair configuration. A pure stair geometry through IFC model showing half cut stair geometry cannot satisfy the drawing presentation and will cause misunderstanding and confusion, thus a PDF workaround workflow is needed for MEP drawing production.

This workflow describes the process of using architectural 2D plans as underlays for MEP design and documentation in Revit. The goal is to integrate tailored architectural PDF plans with MEP models for creating clear and coordinated 2D MEP drawings.

B.5.1. Exporting Tailored Architectural Plans for MEP

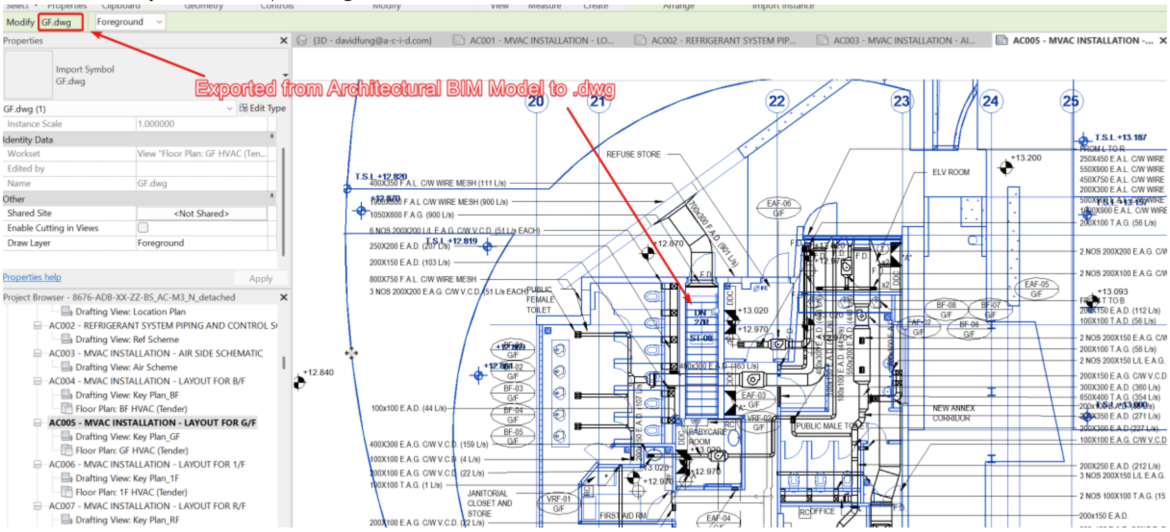
1. Source: The architectural model is prepared in software such as ArchiCAD.
2. Tailor-Made Plans:
    a. Extract 2D plans specifically tailored for MEP purposes, ensuring that they include all relevant architectural elements (e.g., walls, doors, ceilings, and shafts).
    b. Exclude unnecessary details like furniture and decorative elements to reduce visual clutter.
3. Export Format:
    a. Save the tailored architectural plans as dwg files for easy integration into Revit.
4. Ensure sheet sizes, scales, and annotations align with the project's BIM Execution Plan (BEP).



B.5.2. Using Architectural Plans as Underlays in MEP drawings

1. Linking or Importing dwg Files:
    a. In Revit, navigate to Insert > Import dwg and select the tailored architectural dwg files.
    b. Align the imported dwg underlays with the Revit MEP model using shared coordinates or manual alignment tools.

c. (Linking is preferred to Importing as there will be sequential changes, linking can be updated automatically while importing cannot.)



2. Underlay Configuration:

   a. Adjust the visibility settings of the imported dwg/ PDF underlays to ensure they do not obstruct the visibility of MEP elements.

   b. Use the transparency feature to emphasize MEP systems over the architectural layout.

B.5.3. Developing MEP 2D Plans

1. Linking the MEP Model:

   a. Overlay the MEP model onto the architectural dwg/ PDF underlay.

   b. Use Revit tools to generate system-specific 2D plans for Mechanical, Electrical, Plumbing, and Fire Protection systems.

2. Annotation and Detailing:

   a. Add annotations, tags, and labels for ducts, pipes, conduits, and equipment.

   b. Include room names and numbers from the architectural underlays for context.

3. Coordination with Architectural Layouts:

   a. Ensure alignment of MEP elements with architectural features such as walls, ceilings, and shafts.

   b. Address clashes or discrepancies in coordination meetings.

B.5.4. Producing MEP 2D Drawings

1. Sheet Creation:

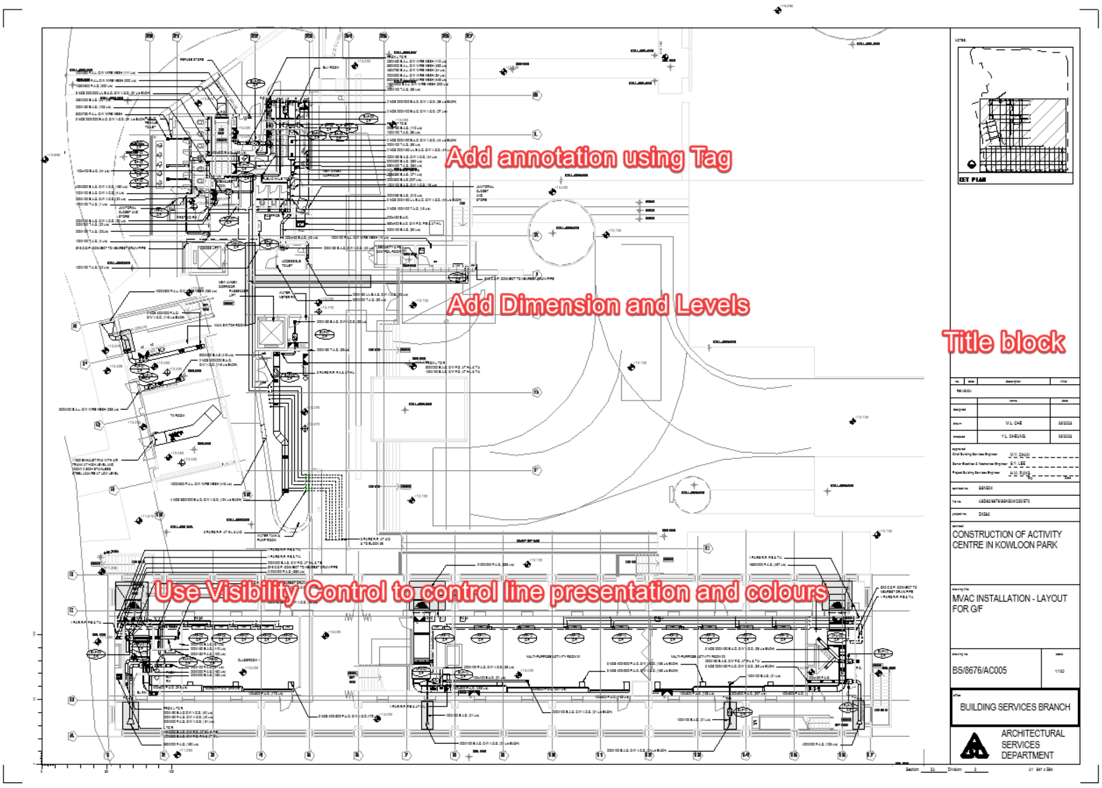
   a. Organize MEP 2D plans into discipline-specific sheets using Revit's Sheet Manager.

b. Add legends, schedules, and notes to the sheets as per project standards.

2. Exporting Final Drawings:

   a. Export the 2D MEP drawings as PDF for distribution and coordination.

   b. Ensure the exported drawings include both the MEP systems and the architectural underlays for comprehensive documentation.



B.5.5. Quality Assurance

1. Verification of Underlays:

   a. Confirm that the architectural PDF underlays are properly aligned with the MEP model. Updating Architectural layouts requires the update of the PDF.

   b. Check for consistency in annotations, scales, and sheet layouts.

2. Coordination Review:

   a. Conduct review sessions with the architectural team to ensure that the MEP systems align with the architectural designs.

   b. Resolve any spatial or layout conflicts before finalizing the drawings

B.6. Exporting to IFC

B.6.1. IFC Scheme and Model View Definition

1. Preparing the Revit MEP Model for IFC Export

   a. Model Cleanup:

      (i) Review the MEP model to ensure all elements are properly categorized and connected if possible (e.g., cable tray, ducts, pipes, and equipment).

      (ii) Delete unused views, families, and elements to reduce file size and data clutter.

   b. Parameter Validation:

      (i) Check that required parameters (e.g. System, System Type) are properly populated.

      (ii) Add custom parameters if required by the BIM Execution Plan (BEP).

   c. Coordinate System Setup:

      (i) Align the model using the Project Base Point / Internal Orgin and Survey Point to match the agreed coordinate system (e.g., HK1980 Grid).

      (ii) Ensure the model origin matches the IfcSite origin defined in the architectural and structural models.

2. Configuring IFC Export Settings

   a. To configure the export settings for your Revit model:

   b. Step-by-Step Navigation:

   c. In Revit, go to the main menu and follow this path:

   d. File > Export > IFC > Modify Setup.

   e. This will open the IFC Export Setup dialog box, where you can customize various settings to control how your model is exported.

   f. Understanding the Export Setup Interface:

      (i) The Modify Setup dialog includes a range of options that allow you to tailor the export process:

      (ii) IFC Schema Selection: Choose the desired IFC schema, such as IFC4 Reference View 1.2, for compatibility with your project requirements.

      (iii) Category Mapping: Map Revit categories to appropriate IFC classes for accurate classification.

      (iv) Property Set and Parameter Mapping: Define how Revit parameters map to IFC property sets to ensure data is correctly exported.

g. Save Custom Export Configurations:

h. After configuring the settings, you can save your setup as a template for future use. This reduces the need to reconfigure the settings for similar projects.

3. Key Export Settings:

a. Proper configuration of export settings is critical to ensure compatibility and data accuracy. Key settings to configure include:

(i) Geometry handling: use BREP/tessellation suitable for IFC4 Reference View; set faceting/tolerance to balance fidelity and file size.

(ii) Element scope/filters: export only visible/required categories, split by levels where helpful, exclude temporary or non-deliverable items.

(iii) Coordinates: use Shared Coordinates; confirm IfcSite/IfcProject alignment with survey data; include true north and elevation.

(iv) Spatial structure: ensure correct IfcProject → IfcSite → IfcBuilding → IfcBuildingStorey assignments.

(v) Classification: map to IFC Types and PredefinedTypes; add external classifications (e.g., local codes) when required.

(vi) (Properties: apply property set/parameter mapping for mandatory LOIN/IDS fields; set units consistently.

(vii) Metadata: populate project/file metadata (author, phase, version) and consistent naming conventions.

b. Mapping Revit Categories to IFC Classes:

(i) Assign Revit categories to appropriate IFC classes for proper classification:

1) Ducts → IfcFlowSegment

2) Pipes → IfcFlowSegment

3) Air Terminals → IfcFlowTerminal

4) Equipment → IfcEquipmentElement

4. Property Set Mapping:

a. Property set mapping ensures that project-specific data is properly carried over into the IFC file. To configure property set mapping:

   (i) Open the IFC Export Setup and go to Property Set/ Parameter Mapping.

   (ii) For each IFC entity (e.g., IfcDuctSegment, IfcPipeSegment, IfcFlowTerminal), choose the target IFC property set and property.

   (iii) Map Revit parameters (project/shared/family) to the chosen IFC properties; use consistent names, units, and data types (e.g., IfcText, IfcAreaMeasure).

   (iv) Prefer standard Psets (e.g., Pset_DuctSegmentCommon, Pset_PipeSegmentCommon) and only add custom Psets where no standard exists.

   (v) Validate the export using an IFC viewer/Validation Service and correct any unmapped or type-mismatched fields.

b. Understand Property Sets in IFC:

   (i) IFC property sets (Psets) are predefined collections of properties assigned to specific elements in an IFC model. Examples include material properties, performance characteristics, and system classifications. Ensuring accurate mapping preserves these details in the exported file.

c. Use the Parameter Mapping Tool:

   (i) The Autodesk IFC Exporter includes a Parameter Mapping Tool that allows you to assign Revit parameters to corresponding IFC property sets. This ensures that critical data is included in the exported file. Examples include:

      1) System Classification → IfcSystem

      2) Fire Rating → Pset_WallCommon.FireRating

      3) Load Classification → IfcLoadGroup

      4) Customize Property Set Mapping

d. Depending on project requirements, you may need to create custom mappings for parameters that are specific to your workflow. This can be done by defining additional parameters in the Modify Setup dialog.

e. Verify Data Accuracy:

   (i) After mapping properties, review the exported IFC file to confirm that all necessary data has been correctly transferred. This step is critical to ensure compliance with project requirements and interoperability with other software tools.

f. Save Property Mapping Templates:

(i) For future projects, save your property mapping configurations as templates. This will streamline the setup process for similar projects and maintain consistency across multiple exports.

5. Exported Elements:

   a. Choose to export only visible elements in the view or all elements in the model.

   b. Exclude temporary or non-essential elements (e.g., placeholders or construction aids).

B.6.2. Exporting the Revit Model to IFC

1. Start the Export Process:

   a. Navigate to File > Export > IFC and select the desired export setup.

2. File Naming and Metadata:

   a. Use a naming convention that follows the BEP (e.g., ProjectName_Discipline_Level_IFC4.ifc).

   b. Add metadata such as project name, author, and organization in the export settings.

3. Export Location:

   a. Save the IFC file in the shared project directory or Common Data Environment (CDE) for access by other disciplines.

B.6.3. Coordinate System Mapping for IFC

1. Export Positioning Options:

   a. Go to File > Export > Options > IFC Options and review the coordinate export settings.

2. Ensure Export Base Point is set to either:

   a. Project Base Point: For local project coordinates.

3. Survey Point: For global georeferenced coordinates.

4. Set IFC Model Origin:

   a. In the IFC Export Settings, specify the positioning:

      (i) Shared Coordinates: Ensures the IFC file aligns with the global Survey Point.

      (ii) Internal Coordinates: Exports the model using Revit's internal origin (not recommended for OpenBIM workflows).

5. Define the IfcSite Location:

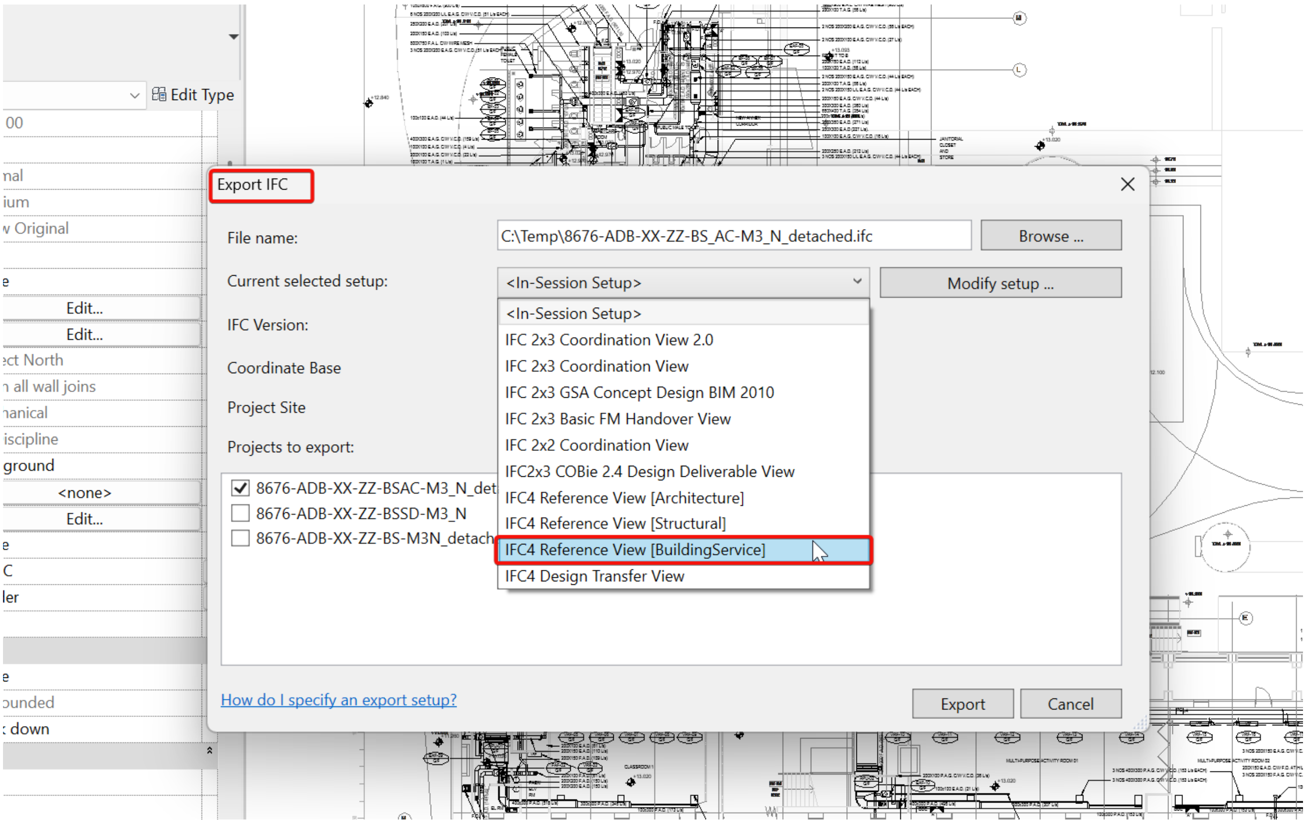
   a. During export, Revit will use the Survey Point as the IfcSite origin by default.

6. Confirm that the IfcSite properties (e.g., latitude, longitude, elevation) are correctly defined in:
    a. Manage > Project Location > Location.

B.6.4. Validating the Exported IFC File

1. Use IFC Viewers:
    a. Open the exported IFC file in an IFC viewer like Solibri, BIMcollab Zoom, or Navisworks to review its contents.
2. Check for Key Elements:
    a. Verify the presence and accuracy of exported MEP elements, such as ducts, pipes, equipment, and spaces.
    b. Confirm that spatial structure (e.g., IfcBuildingStorey) and geometry align with the original Revit model.
3. Property Validation:
    a. Ensure that all required parameters have been transferred correctly.
    b. Validate compliance with the Information Delivery Specification (IDS) and Level of Information Need (LOIN) defined in the BEP.
4. Sharing the IFC File
    a. Collaborate in OpenBIM Environment: Share the IFC file with architectural and structural teams using the agreed CDE or cloud-based platforms like BIM 360, Trimble Connect, or BIMcollab.

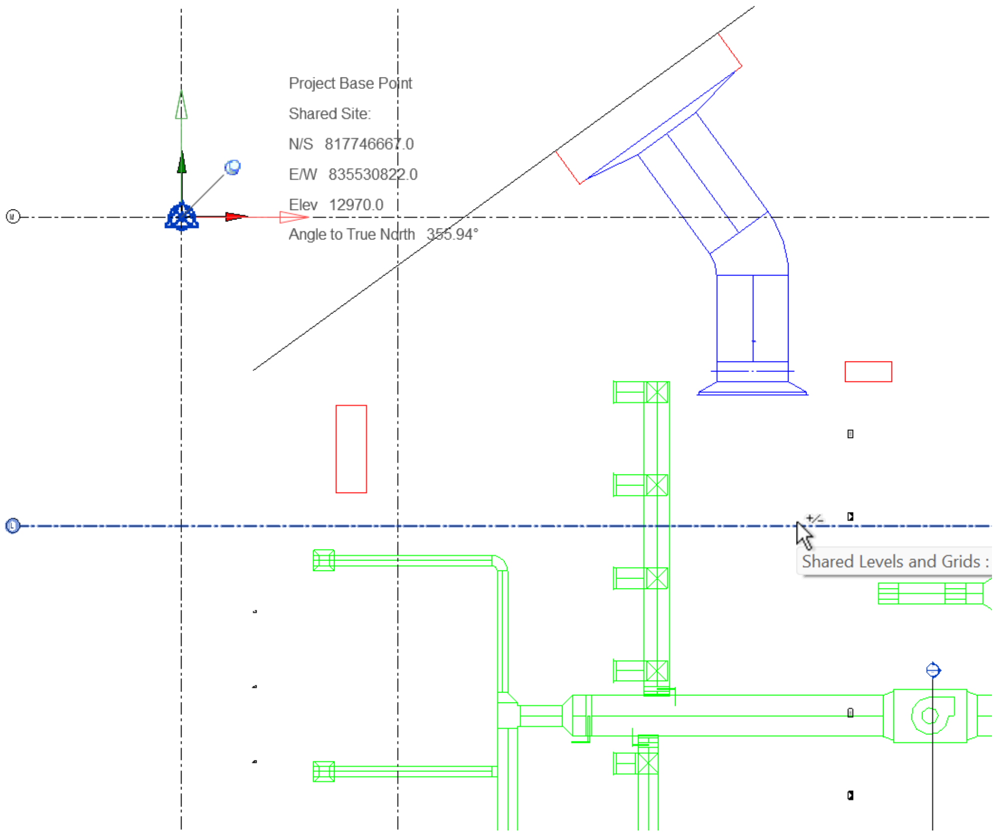
b. Issue Tracking: Use BIM Collaboration Format (BCF) tools if issues arise during coordination,



enabling efficient communication across disciplines.

B.6.5. Best Practices and Tips

1. Use IFC4 Whenever Possible: IFC4 offers improved geometry handling and better support for MEP



elements compared to IFC2x3.

2. Regular Model Reviews: Perform periodic audits of the Revit MEP model to ensure it meets export requirements before sharing.

3. Stay Updated: Keep the Autodesk IFC Exporter plugin updated to leverage new features and compatibility improvements.

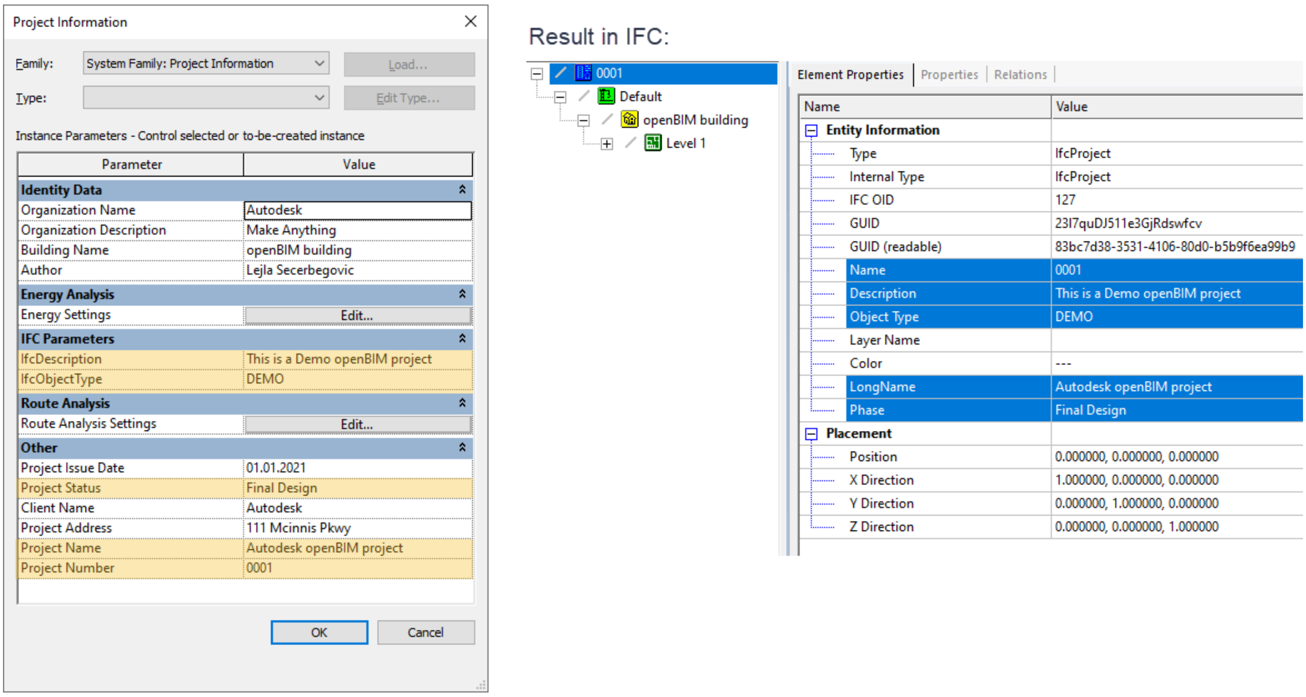
4. Collaborate Early: Coordinate with architectural and structural teams early to establish export standards and resolve potential conflicts

B.6.6. IFC Spatial Aggregation Hierarchy

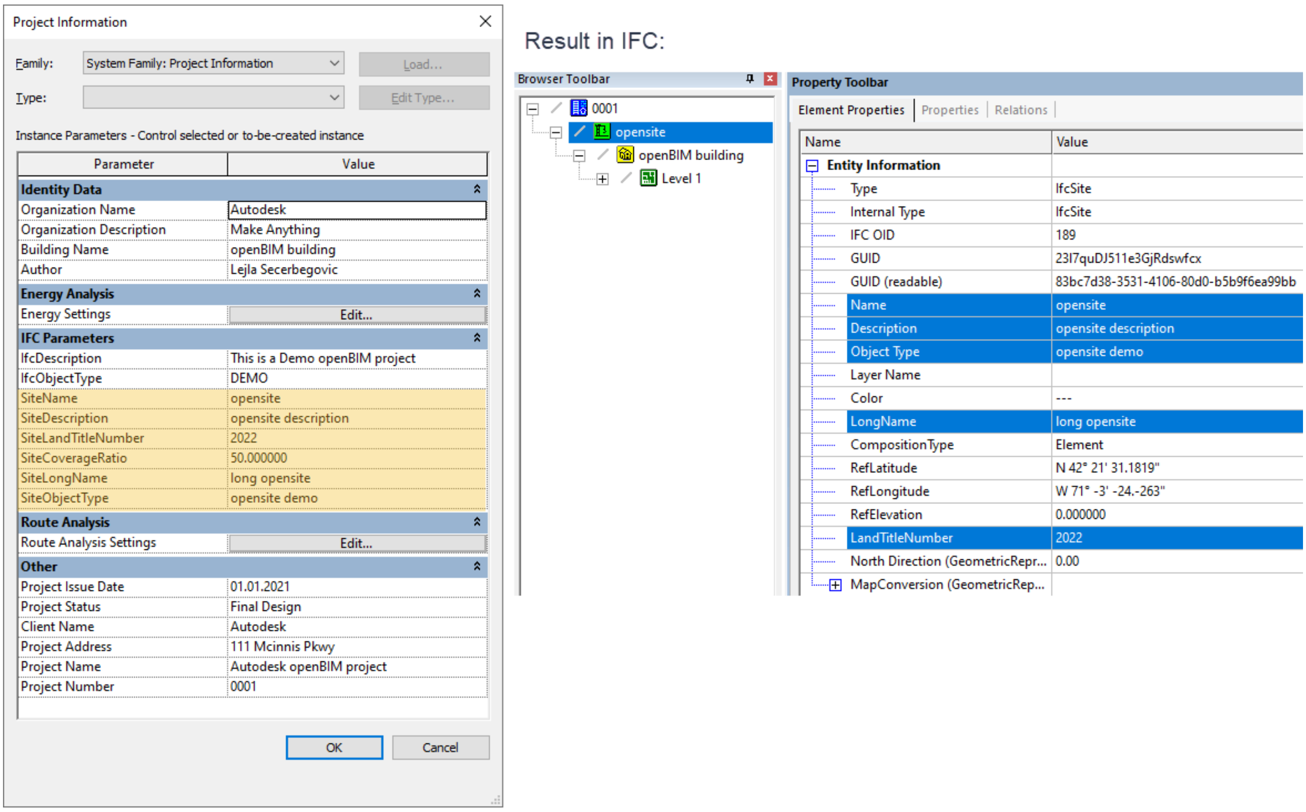
Official Revit IFC Manual on Autodesk website should always refer for detail operation for up-to-date Revit version.

https://autodesk.ifc-manual.com/revit/ifc-export-category-mapping/basic-ifc-structure

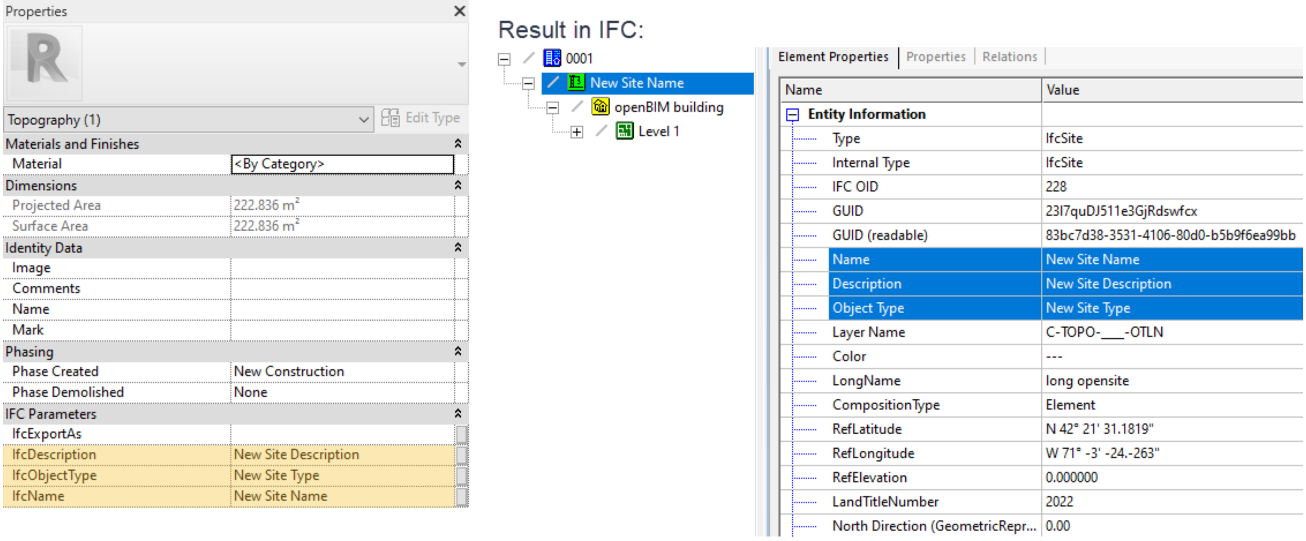
The key screenshots showing the settings is extracted below:



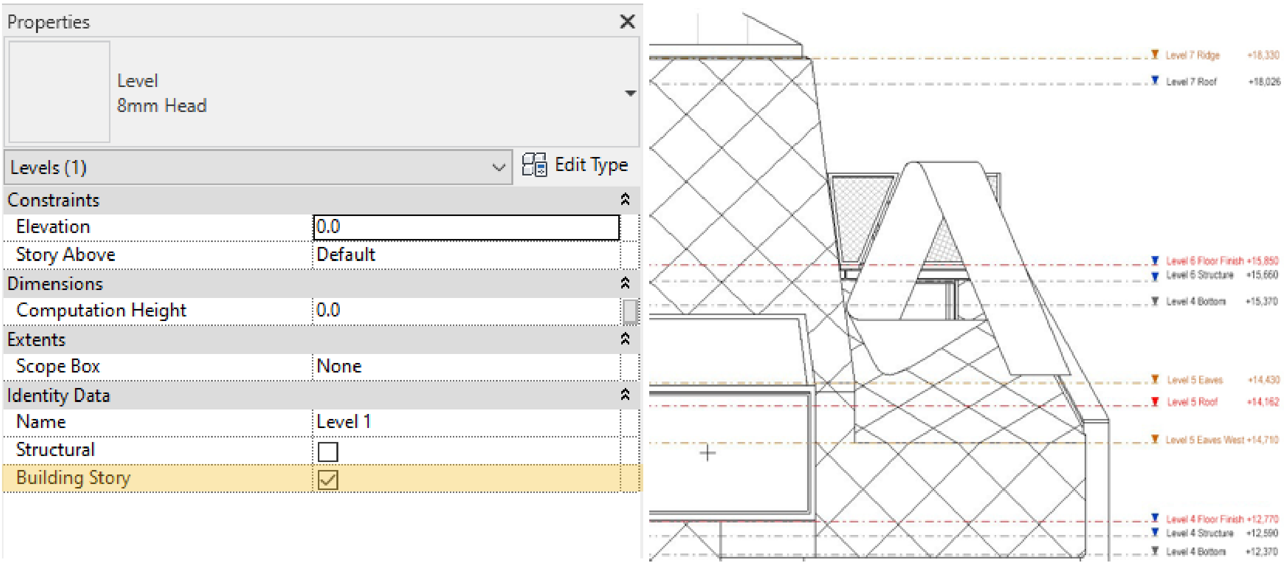
Properties for IfcProject



Properties for IfcSite



Properties for IfcSite



Properties for IfcBuildingStorey

B.7. Common Challenges and Solutions

B.7.1. Misaligned Linked Models

1. Issue: The linked IFC model does not align with the Revit MEP model.

2. **Solution**:

   a. Verify that the correct positioning method (Shared Coordinates) is used.

   b. Align the Survey Point and Project Base Point with the IFC model's coordinate system.

B.7.2. Missing Elements

1. Issue: Some elements from the IFC model are not visible in Revit.

2. Solution:
    a. Check the Imported Categories tab in Visibility/Graphics (VG).
    b. Ensure that all required IFC categories are enabled.

B.7.3. Performance Issues

1. Issue: Large IFC models slow down Revit performance.
2. Solution:
    a. Request filtered IFC files containing only relevant elements.
    b. Use Section Boxes and Worksets to limit the displayed geometry.

B.7.4. Misaligned Models in IFC

1. Issue: Architectural and structural models misalign with the MEP model when federated.
2. Solution:
    a. Ensure all disciplines use the same coordinate origin (e.g., Survey Point).
    b. Confirm that the Survey Point and Project Base Point are consistent across all models.

B.7.5. Rotation or Scaling Issues

1. Issue: The IFC model appears rotated or scaled incorrectly.
2. Solution:
    a. Verify that the True North and Project North are correctly configured.
    b. Ensure the base units in Revit (e.g., meters) match those in the IFC export settings.

B.7.6. Loss of Georeferencing in IFC

1. Issue: Georeferenced data is missing in the exported IFC file.
2. Solution:
    a. Use Shared Coordinates during export.
    b. Confirm that the IfcSite entity includes georeferenced data (e.g., latitude, longitude).

B.8. Best Practices for Coordinate System Management

1. Set Coordinates Early:
    a. Define the Survey Point and Project Base Point at the start of the project.
    b. Share these settings with all disciplines to ensure consistency.
2. Use Shared Coordinates:

   a. Always use Shared Coordinates for IFC exports to maintain alignment with the overall project.

3. Test Model Alignment:
   a. After exporting the IFC file, test the model's alignment in different IFC viewers (e.g., Solibri, BIMcollab Zoom) or a federated model (e.g., in Navisworks).

4. Document the Coordinate Setup:
   a. Include detailed descriptions of the coordinate system in the BIM Execution Plan (BEP).
   b. Specify whether local (PBP) or global (SP) coordinates are used.

5. IFC Export Settings:
   a. Positioning: Shared Coordinates
   b. Units: Meters
   c. Split by Levels: Enabled

6. Verifying Coordinates in IFC file
   a. Use an IFC viewer (e.g., Solibri, Trimble Connect) to verify the following:
   b. The IfcSite origin matches the georeferenced Survey Point.
   c. The spatial hierarchy (IfcBuildingStorey, IfcSpace) aligns with the project levels.

7. Check Georeferenced Data:
   a. Confirm that latitude, longitude, and elevation data are included in the IfcSite entity.

Appendix C

C. COMPREHENSIVE GUIDE TO OPENBIM WORKFLOW IN TEKLA STRUCTURES

C.1. Exporting to IFC

C.1.1. IFC Scheme and Model View Definition

Tekla Structures supports various IFC schemes and Model View Definitions (MVD). IFC4 Reference View (IFC4 RV) is recommended for general project exchanges due to its broad acceptance and verified interoperability across software platforms.

IFC2x3 Coordination View 2.0 (CV2.0) is available for legacy project compatibility. Users must select the appropriate IFC schema and MVD within Tekla's IFC export settings, ensuring alignment with project-specific BEP/EIR requirements.

C.1.2. Coordinates

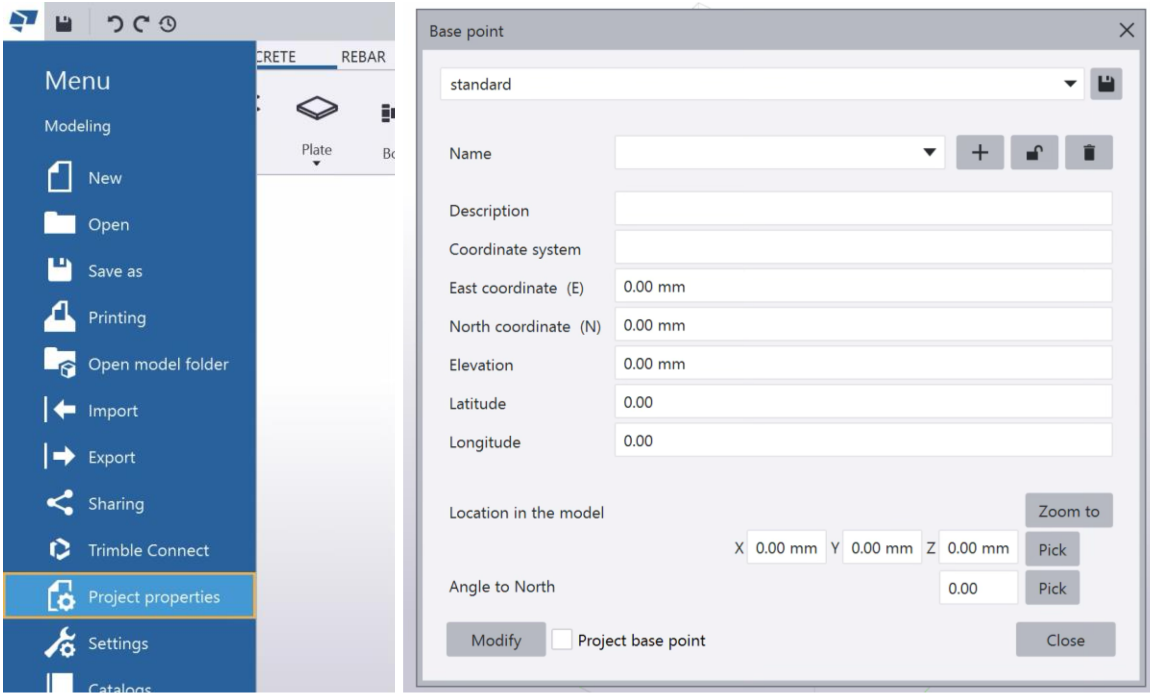
Coordinate management in Tekla Structures must follow the standard IFC spatial hierarchy:

- IfcProject: aligns with the HK1980 grid system, fixed at coordinates (0,0,0).
- IfcSite: aligns with Tekla Structures' "Global Point of Origin." Positioning should ensure models are near the origin point to avoid performance degradation and maintain IFC round-trip compatibility.

Practically, models should be established with a site-specific reference point, using Tekla's "Base Point" feature. Base Points must accurately represent the project's survey data and IFC Site coordinates. Proper configuration ensures seamless integration when federated with models from other disciplines.

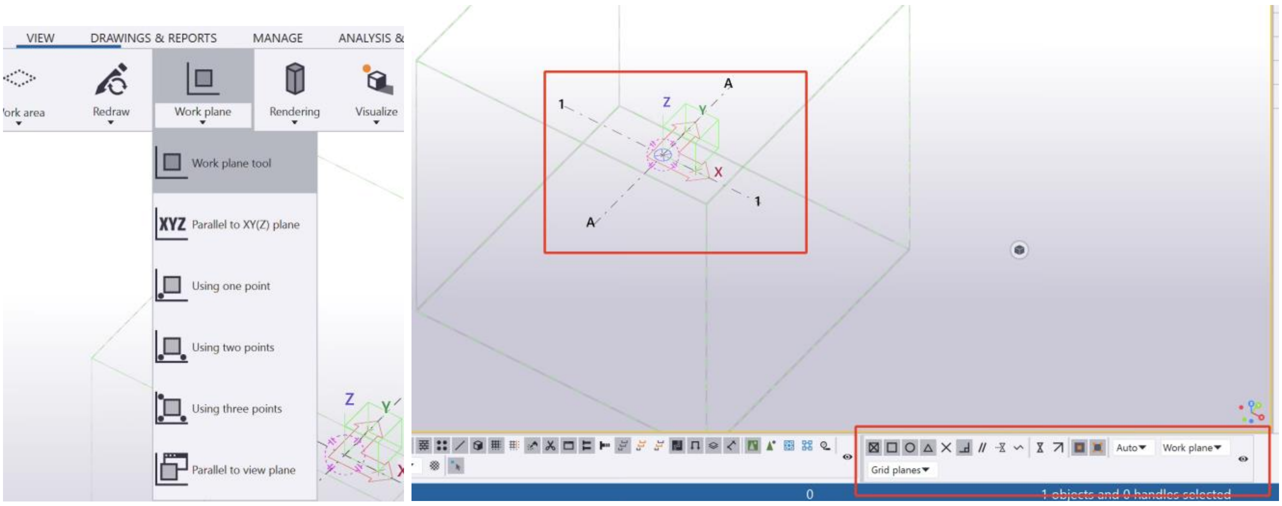
Global Coordinate System:

- Represented by a green cube symbol.
- Located at the global origin (x=0, y=0, z=0).
- It is static and cannot be changed.
- Avoid placing the model far from the origin, as this can reduce accuracy and make snapping to points difficult.
- For inserting reference models or exporting IFC models, you can use base points to work with other coordinate systems while keeping coordinates manageable.



Local Coordinate System (Work Plane):

- Represents the local coordinate system and is model-specific.
- Commands like creating points, part positioning, and copying comply with the work plane coordinates.
- The coordinate symbol in the lower-right corner of the model view follows the work plane.
- The work plane can be shifted to change the local coordinate system.
- The red arrow symbol indicates the xy plane, and the z direction follows the right-hand rule.
- The work plane has its own red grid, which can be shown or hidden for positioning parts.



C.1.3. IFC Spatial Aggregation Hierarchy

Tekla Structures must correctly set up IFC spatial aggregation (IfcBuilding, IfcBuildingStorey). Structural elements should be accurately placed according to their associated storey using Tekla's object hierarchy. Each element must be linked via spatial containment relationships clearly defined in the IFC export settings.

The IFC Spatial Aggregation Hierarchy is a structure used to organize and classify building elements in an IFC model. It defines the spatial relationships between objects, such as how elements are grouped into buildings, floors, and other spatial entities. This hierarchy is essential for managing and exchanging data in BIM workflows.

Key Levels in the IFC Spatial Aggregation Hierarchy:

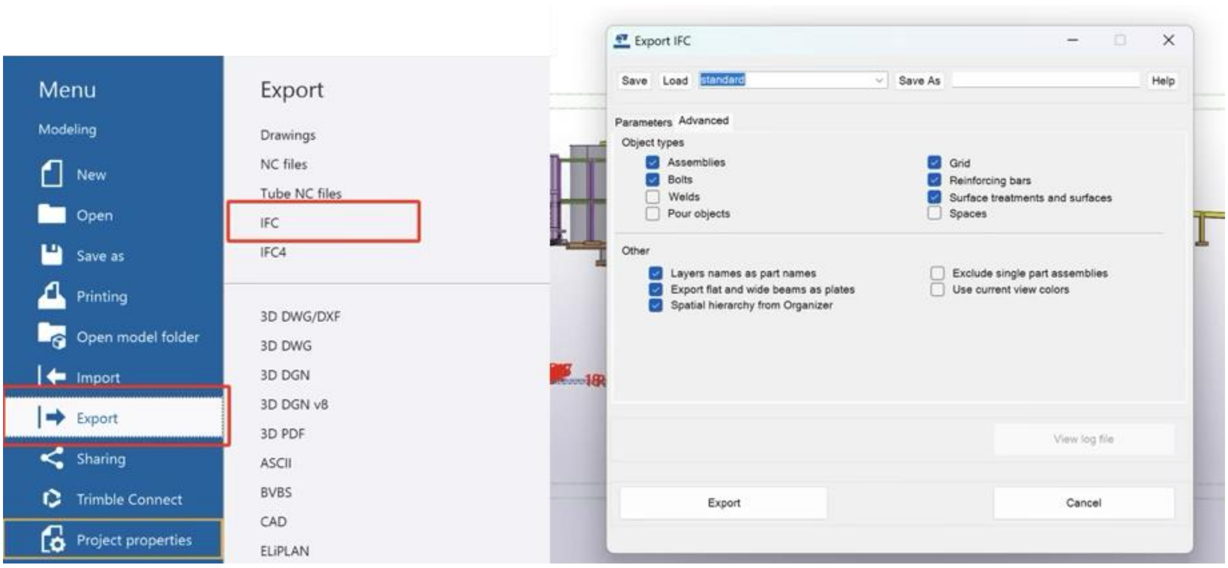
1. IfcProject: The topmost level representing the entire project.
2. IfcSite: Represents the site or location of the project.
3. IfcBuilding: Represents a single building within the site.
4. IfcBuildingStorey: Represents individual floors or levels within the building.
5. IfcSpace: Represents specific spaces or rooms within a building storey.

Usage in Tekla Structures:

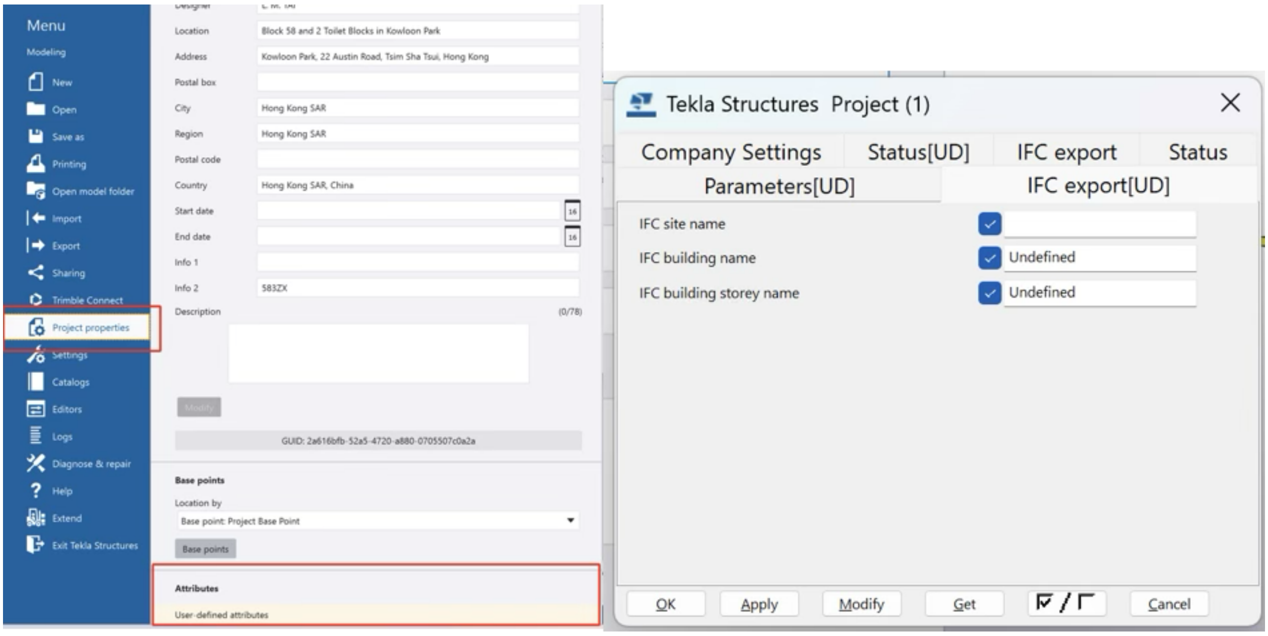
- Tekla Structures allows users to define and export the spatial hierarchy for IFC models.
- Users can use the Organizer to create a spatial hierarchy (Building-Site-Section-Floors) and export it with the model.
- If users do not use the Organizer, the hierarchy can be defined using part UDAs (User-Defined Attributes) or project property UDAs.

Exporting Spatial Hierarchy:

- From Organizer: Select "Spatial hierarchy from Organizer" in the IFC export settings. Ensure the hierarchy is created and synchronized in the Organizer before export.



- From UDAs: If not using the Organizer, the hierarchy is defined in part UDAs or project property UDAs.



Example of Spatial Hierarchy in IFC:

IfcProject

└── IfcSite

└── IfcBuilding

└── IfcBuildingStorey

└── IfcSpace

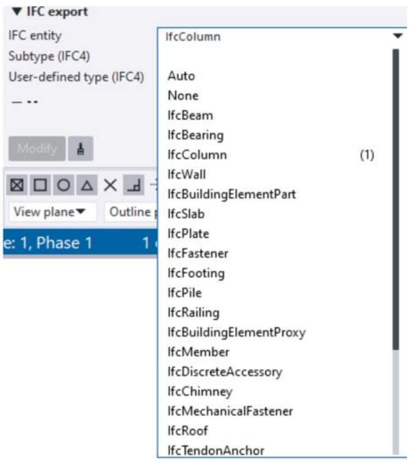
└── Building elements (e.g., beams, columns, slabs)

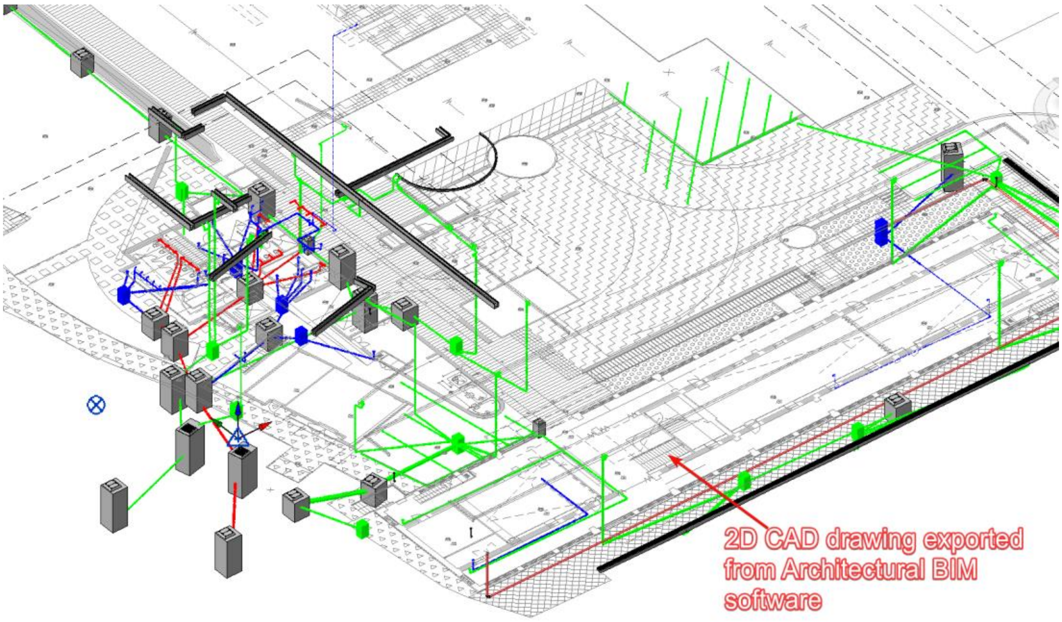
This hierarchy ensures that all elements are properly categorized and can be easily understood and managed in other BIM tools.

Before exporting Tekla Structures model objects to IFC, users need to define the resulting IFC entities for the exported model objects in the object properties.

1. Double-click an object in the model, for example a column, to open part properties.

2. In the IFC export section in the property pane, select an option in the IFC entity list to define the IFC entity for the exported model object.The default entity for the object type is selected by default.





C.1.4. IFC Types and Predefined Type Mapping

Tekla Structures allows explicit mapping between native objects and IFC Types/Predefined Types through the IFC Export Configuration dialog. Users should ensure accurate mappings, minimizing generic types like IfcBuildingElementProxy. Typical mappings include:

- Steel Beam → IfcBeam
- Reinforced Concrete Column → IfcColumn
- Steel Connection → IfcMechanicalFastener or IfcPlate

Tekla Structures has predefined mapping settings, but project-specific mappings should be verified and adjusted per BEP/EIR requirements.

In Tekla Structures, IFC types and predefined types are used to define the classification and purpose of objects when exporting to IFC. These classifications ensure that the exported IFC model is compatible with other BIM tools and follows the buildingSMART standards.

IFC Types:

- IFC Entity Types: These are the main classifications of objects in the IFC schema, such as IfcBeam, IfcColumn, IfcSlab, etc.
- User-Defined IFC Entity: If the desired IFC entity is not available in Tekla Structures, the user can define a custom entity using the UserDefinedEntity option.

Predefined Types:

- Predefined Types: These are subcategories or specific roles of an IFC entity. For example:

- IfcBeam can have predefined types like T_BEAM, LINTEL, or USERDEFINED.
- IfcColumn can have predefined types like COLUMN, PILASTER, or USERDEFINED.

- User-Defined Predefined Type: If the predefined type is not available, the user can select USERDEFINED and manually input the desired type.

Mapping in Tekla Structures:

1. Default Mapping:
   - Tekla Structures automatically maps objects to IFC types based on their object type in the model (e.g., beams are mapped to IfcBeam).
   - Predefined types can also be automatically assigned based on object properties.
2. Custom Mapping:
   - Users can override the default mapping by defining custom IFC types and predefined types in the User-Defined Attributes (UDA) dialog under the IFC Export tab.
   - For example:
     - Set IFC entity user to UserDefinedEntity and input the desired entity in IFC entity user free field.
     - Specify the predefined type in the Predefined type field.
3. IFC Extender (ML126):
   - The IFC Extender application provides additional control over IFC type and predefined type mapping.
   - Users can set predefined types for all objects of a specific IFC entity in the IFC Part Type and IFC Assembly Type tabs.

Example of Mapping:

Tekla Object Type	IFC Entity	Predefined Type	Custom Mapping Example
Beam	IfcBeam	T_BEAM	UserDefinedEntity: IfcBeam, Predefined Type: USERDEFINED
Column	IfcColumn	COLUMN	UserDefinedEntity: IfcColumn, Predefined Type: PILASTER

Plate	IfcPlate	BASE_PLATE	UserDefinedEntity: IfcPlate, Predefined Type: USERDEFINED

Notes:

- The mapping ensures that the exported IFC model is properly classified and can be used effectively in downstream workflows.
- If no predefined type is set, the default type is used.
- The IFC Extender allows users to map additional properties and customize the export further.

IFC Extender (ML126)

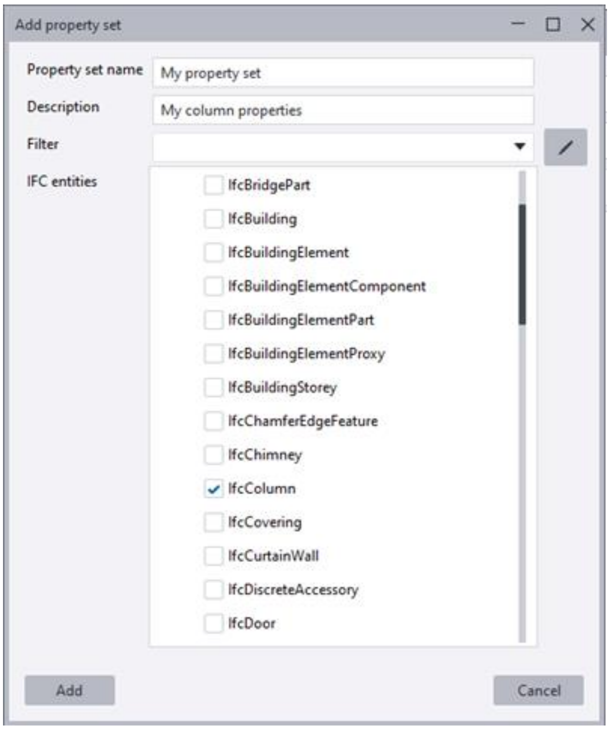
Properties Mapping

Property mapping in Tekla Structures is facilitated through IFC Property Sets (Psets) which can be predefined or custom. Users must ensure properties required by the project's BEP/EIR are explicitly mapped. Use Tekla's custom property definitions (UDA - User Defined Attributes) to comply with departmental requirements and standards, like ArchSD's Elemental Codes or EMSD's maintenance standards.

In Tekla Structures, properties mapping refers to the process of associating IFC properties with Tekla Structures attributes during IFC export or import. This ensures that the correct data is transferred between Tekla Structures and other software using the IFC format.

Key Aspects of Properties Mapping:

1. Mapping IFC Properties to Tekla UDAs:
   - User-defined attributes (UDAs) in Tekla Structures can be mapped to IFC properties.
   - This mapping allows the export of Tekla-specific data into IFC files.
2. Property Set Configuration Files:
   - Property sets for IFC export are defined in XML configuration files.
   - These files specify which Tekla attributes or UDAs are included in the IFC property sets.
   - You can modify or create custom property set configuration files to meet project-specific requirements.

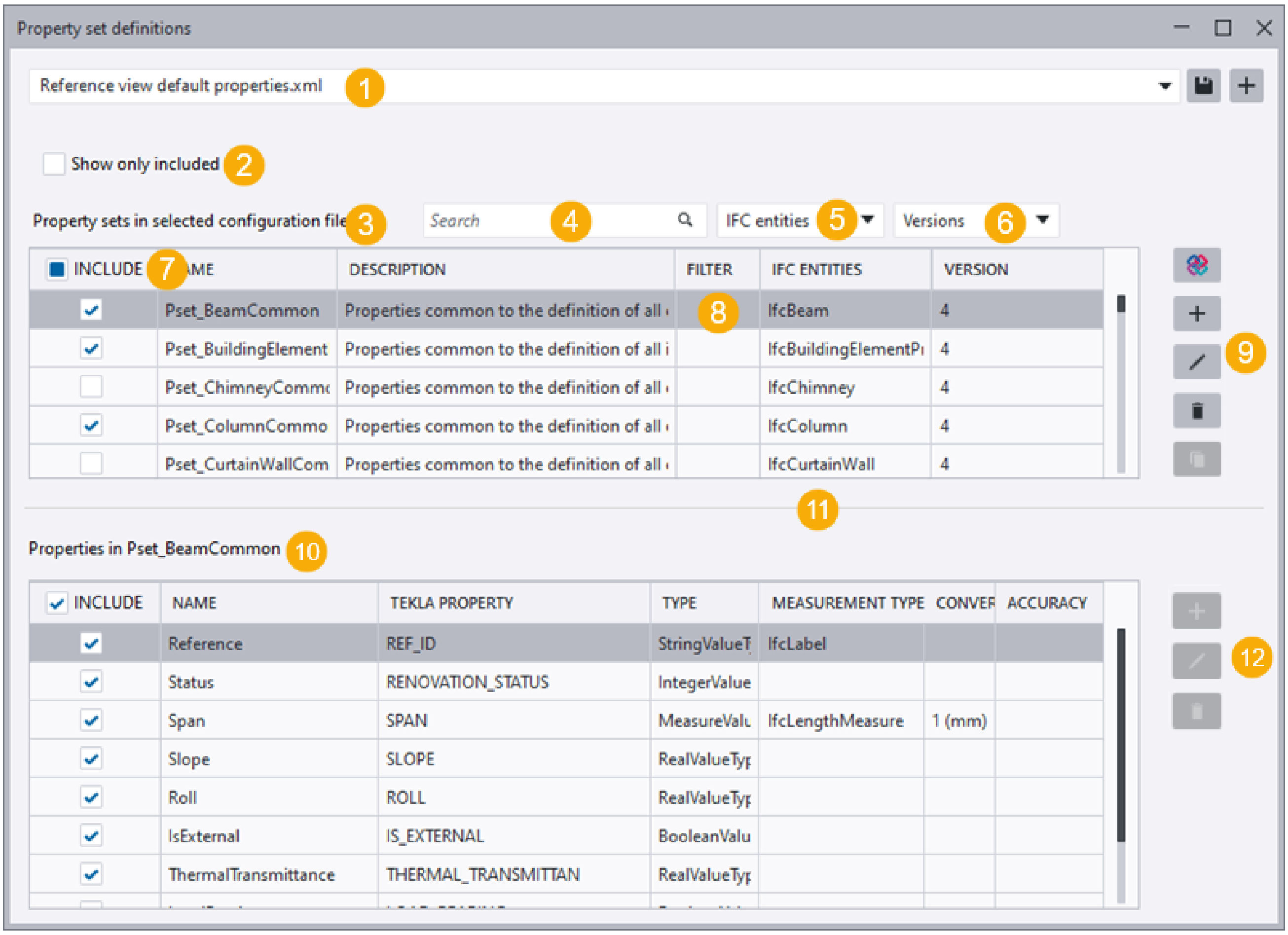


3. Default and Custom Property Sets:
    - Tekla Structures provides default property sets for IFC export.
    - Users can create custom property sets or modify existing ones using the **Property set definitions** dialog.
4. Steps for Mapping Properties:
    - Open the **Property set definitions** dialog from the IFC export settings.
    - Add or modify property sets and properties.
    - Map Tekla attributes or UDAs to IFC properties.
    - Save the configuration file in the \AdditionalPsets folder under the current model folder or in the XS_FIRM, XS_PROJECT, or XS_SYSTEM folders.
5. Supported Data Types:
    - Tekla Structures supports various data types for mapping, including:
        - String

- Integer
- Boolean
- Measurement (e.g., length, area, volume)
- Timestamp

6. BuildingSMART Property Sets:

  - Tekla Structures supports importing and using predefined BuildingSMART property sets (e.g., Pset_, Qto_).
  - These property sets are protected and cannot be modified but can be included or excluded during export.



Example of Property Mapping:

- Tekla UDA: ASSEMBLY_POS

- Mapped IFC Property: Assembly/Cast unit Mark
- IFC Property Set: Tekla Assembly

This mapping ensures that the assembly position in Tekla Structures is exported as the "Assembly/Cast unit Mark" property in the IFC file.

C.1.5. Geometry Conversion

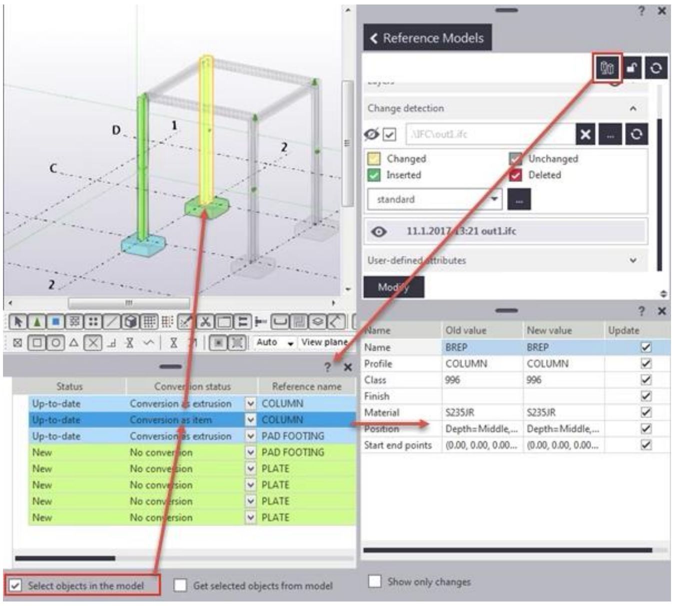
Tekla Structures exports geometry predominantly in Boundary Representation (BREP), ensuring accurate visual representation in IFC viewers. Geometry should be carefully simplified where possible to manage polygon count effectively. Tekla's internal tools (e.g., Polygon Reducer) assist in optimizing geometry for IFC exports, balancing visual fidelity and performance.

Geometry conversion in Tekla Structures refers to the process of converting imported geometry, such as IFC objects, into native Tekla Structures objects. This is particularly useful for integrating models from other software into Tekla Structures for further detailing, clash detection, or other workflows.

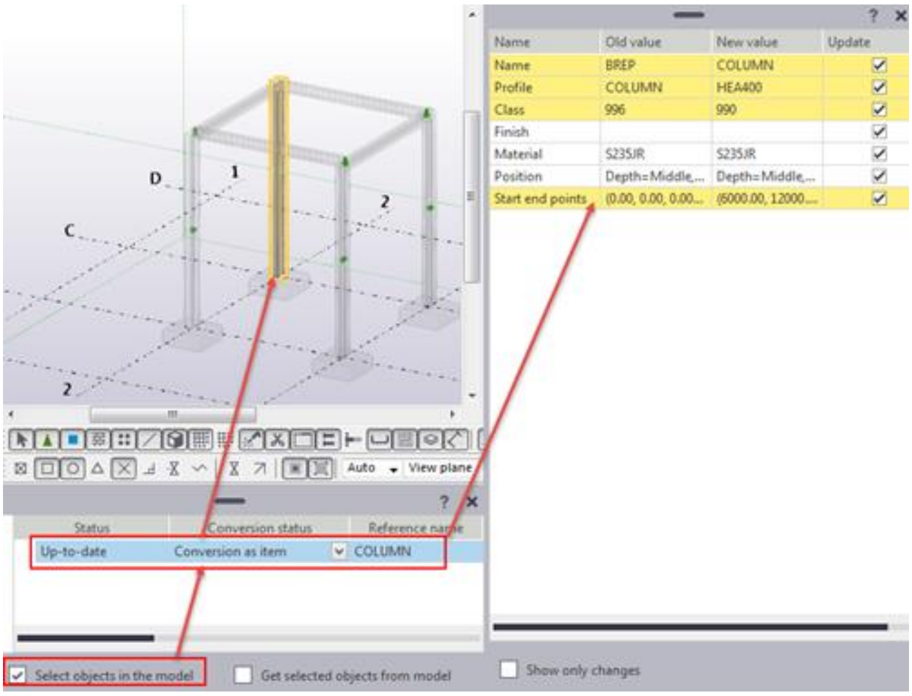
Key Points About Geometry Conversion:

1. Supported Object Types:
   - Most linear IFC objects, such as beams, columns, braces, plates, slabs, footings, and walls, can be converted.
   - Polybeams with curved sections and objects exported from Tekla Structures are also supported.
   - Conversion supports string, integer, and double-type UDAs (User-Defined Attributes).
2. Conversion Methods:
   - Conversion as Item: Converts an IFC object into a Tekla Structures item where the 3D shape defines the geometry.
   - Conversion as Extrusion: Converts an IFC object into a part (e.g., column, beam, plate) with a profile extruded to create the length.
3. Steps for Conversion:
   - Ensure profiles and units in the IFC reference model are compatible with the Tekla Structures environment.
   - Check and adjust the IFC object conversion settings.
   - Convert objects using one of the following methods:

- Convert IFC Objects Command: Converts all selected objects at once.
- Conversion Change Management: Provides change detection and allows selective conversion.



4. Profile Conversion Logic:
   - If a parametric profile is used in the IFC model, Tekla Structures attempts to match it with profiles in its catalog.
   - If no match is found, a default parametric profile or a new profile based on the IFC description is created.
5. Limitations:
   - IFC4.0, IFC4.1, and newer formats are not supported for conversion.
   - Bolts, reinforcement, and welds cannot be converted.
   - Only certain IFC representations (SweptSolid, Brep, CSG, Clipping) are supported.
6. Checking Conversion Results:
   - After conversion, use the changes list to review the status of converted objects.



- Statuses include New, Changed, Deleted, Up-to-date, or Error.

Example:

To convert beams and columns from an IFC model:

1. Import the IFC model as a reference model.
2. Hide unnecessary layers to focus on relevant objects.
3. Use the Convert IFC Objects command to convert the selected objects.
4. Check the conversion results in the changes list and verify the geometry in the model.

C.1.6. Model Filtering

IFC model exports from Tekla Structures should filter unnecessary elements based on BEP-defined modeling responsibilities. Common rules include:

- Exclude temporary or construction-phase elements.
- Exclude duplicate elements modeled by other disciplines (e.g., architectural finishes, unless explicitly structural).

- Clearly define structural scope, including critical openings and load-bearing walls, beams, and columns.

Tekla's IFC Export filter tool supports complex filtering rules, enabling precise definition of exportable elements.

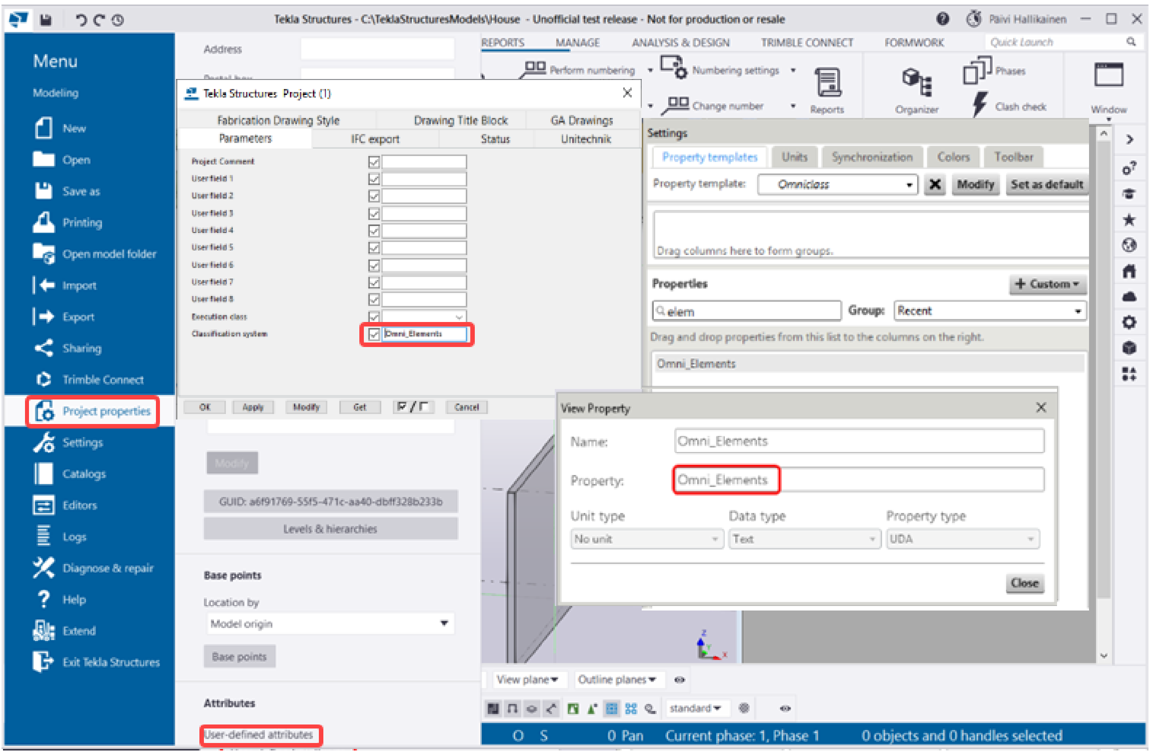
User Assistant

In Tekla Structures, users can filter model objects for IFC export by using selection filters or defining specific criteria in the export settings. This allows users to control which objects are included in the exported IFC file.

Steps for Filtering Model Objects for IFC Export:

1. Use Selection Filters:
    - Users can create and apply selection filters to choose specific objects in the model.
    - Filters can be based on object properties, such as part name, material, or phase.
2. Define Object Types in Export Settings:
    - In the IFC export dialog, users can select specific object types to include, such as assemblies, bolts, grids, reinforcing bars, or surface treatments.
    - For example, users can choose to export only assemblies or exclude single-part assemblies.
3. Use Organizer for Spatial Hierarchy:
    - Users can create a spatial hierarchy in the Organizer and use it to filter objects for export.
    - This is particularly useful for exporting objects based on their location in the building structure (e.g., specific floors or sections).
4. Modify User-Defined Attributes (UDAs):
    - Users can define UDAs for objects to control their inclusion in the IFC export.
    - For example, users can set the "IFC export type" or assign specific classification codes to objects.
5. Export Selected Objects:
    - Users can manually select objects in the model and choose the "Selected objects" option in the IFC export settings.
    - This method is useful for exporting a specific subset of the model.

By using these filtering methods, users can ensure that only the required objects are included in the exported IFC file, improving efficiency and reducing file size.



C.2. Importing IFC

C.2.1. Import settings

IFC import settings in Tekla Structures must strictly preserve original IFC geometry, properties, and spatial hierarchies without unintended transformation. IFC files should be imported using Tekla's IFC object import dialog, setting geometry type to "BREP" to maintain accuracy.

To import IFC models into Tekla Structures, users can adjust the import settings to ensure proper handling of the model. Below are the key details related to importing IFC models:

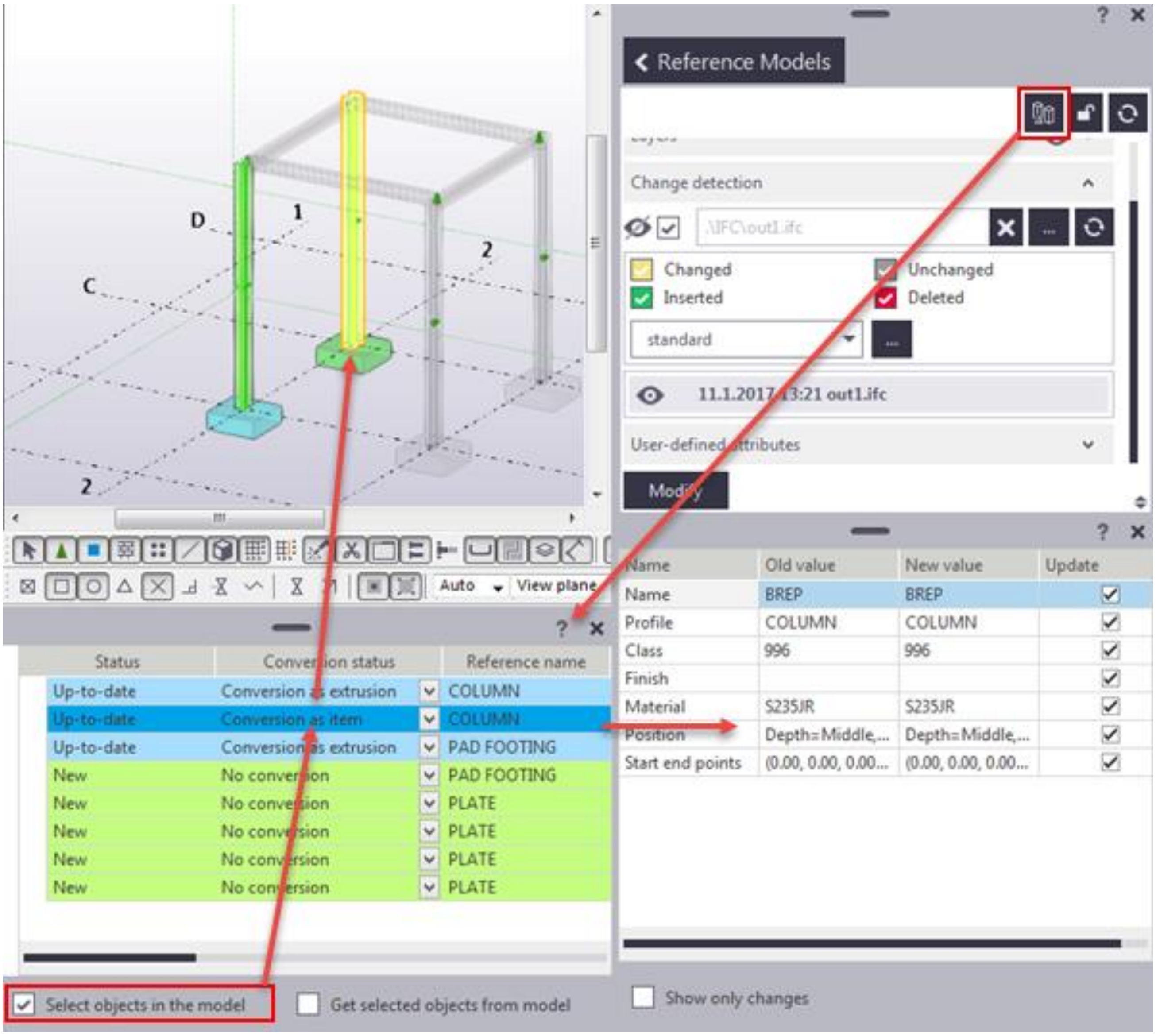
Importing IFC Models

1. Importing as Reference Models:
   - IFC models can be imported as reference models for coordination and visualization.
   - Reference models are not editable but can be used for clash detection, reporting, and scheduling.
2. Importing IFC Categories to Organizer:
   - Users can import the location breakdown structure of an IFC model as IFC categories into the Organizer.
   - Steps:
     1. Open Organizer by clicking Manage > Organizer.

2. Select a project, right-click, and choose New IFC Project.
3. Select the IFC model and click Import.
4. The IFC categories are imported under location categories, and objects are automatically included.

3. Conversion of IFC Objects:
   - Users can convert IFC objects into native Tekla Structures objects for further modeling and editing.
   - Conversion supports linear objects like beams, columns, slabs, and walls.
   - Before conversion, users should check the profiles and units in the IFC model for compatibility with the Tekla Structures environment.
4. Settings for Import:
   - Users can manage layers and visibility of the imported IFC model.

- Irrelevant layers can be hidden to focus on specific parts of the model.



5. Updating Imported IFC Models:

   - If the IFC model is updated, users can replace the existing model with the updated version while retaining the imported categories and settings.

Notes:

- The quality of the imported IFC model depends on the compliance of the IFC file with the IFC standard.
- Tekla Structures supports IFC2x3, IFC4, and IFC4.3 schemas for importing.

C.2.2. Coordinates & Positioning

Imported IFC models must align with Tekla Structures' global origin and base points. Verify alignment immediately post-import to ensure that IFC coordinates match native project setups. Corrective actions, if necessary, should adjust base points or IFC positioning rather than altering geometry.

When working with IFC models in Tekla Structures, coordinates and positioning play a crucial role in ensuring proper alignment and collaboration between different disciplines. Tekla Structures provides several options to manage coordinates and positioning for IFC import and export.

Key Concepts for IFC Coordinates & Positioning:

1. Global Coordinate System:

- The global coordinate system in Tekla Structures is fixed and represents the origin (0,0,0).
- It is used as the default reference for positioning objects in the model.

2. Base Point:

- A base point is used to define a local coordinate system for the model.
- It allows you to align the Tekla Structures model with external coordinate systems, such as those used in other software or on-site.
- Base points include values like East coordinate, North coordinate, Elevation, Angle to North, Latitude, and Longitude.
- Base points are particularly useful for IFC export to ensure proper alignment with other models.

3. Positioning Options for IFC Export:

- Model Origin: Exports the model relative to the global origin (0,0,0).
- Work Plane: Exports the model relative to the current work plane coordinate system.
- Base Point: Exports the model relative to a defined base point. This is the most common method for aligning models in collaborative projects.

4. Base Point Export Systems:

- IfcMapConversion: Converts the model's local coordinate system into the global coordinate system. This is required when the IFC4 schema is used.
- IfcSite Coordinate System: Converts coordinates for each object separately. This method is compatible with most IFC viewers, including Trimble Connect.

5. Checking Exported Coordinates:

- After exporting an IFC model, you can check the coordinates by inserting the exported IFC model back into the original Tekla Structures model as a reference model.

- Use different colors for the original model and the IFC reference model to visually verify alignment.

Recommendations for IFC Positioning:

- Always define a base point when working on collaborative projects to ensure proper alignment with other disciplines.
- Use IfcMapConversion for IFC4 exports when global coordinate systems are required.
- Verify the exported IFC model by inserting it back into Tekla Structures or using an IFC viewer like Trimble Connect.

C.2.3. Visibility Controls

Imported IFC models should utilize Tekla's visibility groups and object representation settings. Clear visibility rules facilitate model management, allowing toggling IFC elements independently of native Tekla Structures objects. This significantly enhances clarity during coordination.

In Tekla Structures, visibility controls for IFC models allow users to manage how IFC reference models are displayed and interacted with. These controls are essential for reviewing, coordinating, and managing imported IFC models effectively.

Key Visibility Controls for IFC Models:

1. Layer Visibility:
    - Users can hide or show specific layers in the IFC model.
    - To manage layers:
        - Open the Reference models list.
        - Double-click the reference model to open its details.
        - Use the Layers list to toggle the visibility of layers by clicking the eye icon next to each layer.
2. Color and Transparency:
    - Users can apply different colors or transparency settings to distinguish the IFC model from the Tekla Structures model.
    - This is particularly useful for visual comparison and clash detection.
3. Clip Planes:

   - Clip planes can be used to isolate specific sections of the IFC model for detailed inspection.
   - This helps in visually checking the alignment and integration of the IFC model with the Tekla Structures model.
4. Spatial Hierarchy:
   - If the IFC model includes a spatial hierarchy (e.g., buildings, floors, spaces), users can navigate and control the visibility of elements based on this hierarchy.
   - This can be managed through the Organizer or directly in the reference model settings.
5. Selection and Highlighting:
   - Users can select and highlight specific objects in the IFC model for detailed inspection.
   - This is useful for identifying and managing specific elements within the reference model.
6. Use Current View Colors:
   - When exporting IFC models, users can choose to export objects using the colors defined in the current view settings. This ensures that the exported model reflects the visual representation in Tekla Structures.

These visibility controls allow users to efficiently manage and interact with IFC models, ensuring better coordination and integration in BIM workflows.

C.3. Issue Tracking

C.3.1. Local BCF manager Workflow

Tekla Structures integrates native support for BCF. Users should adopt Tekla's built-in BCF manager to track design coordination issues directly within the structural model environment. The workflow includes:

- Creating and annotating BCF issues linked to specific structural components.
- Exporting BCF files for exchange with architectural and MEP teams using compatible platforms like BIMcollab, Trimble Connect, or BIM Track.

The BCF (BIM Collaboration Format) Manager in Tekla Structures allows users to manage project-based issues and collaborate effectively using BCF topics. Below is the workflow for managing BCF topics locally in Tekla Structures:

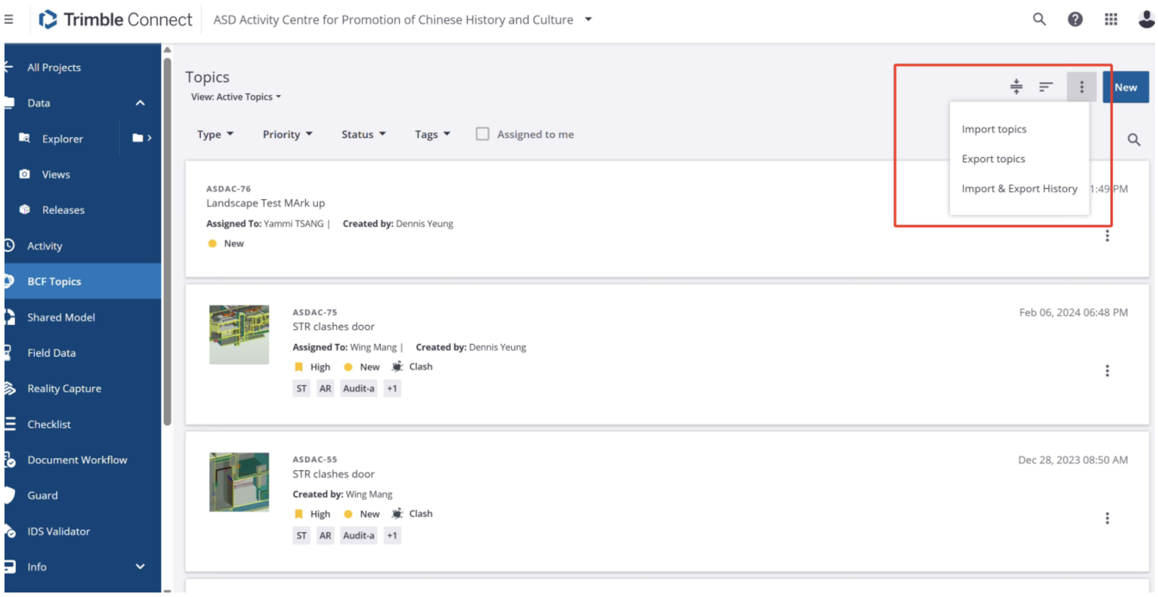
Workflow for Local BCF Manager:

1. Open the BCF Topics Side Pane:

- In Tekla Structures, go to the Trimble Connect ribbon tab.
- Click BCF Topics to open the BCF Topics side pane.

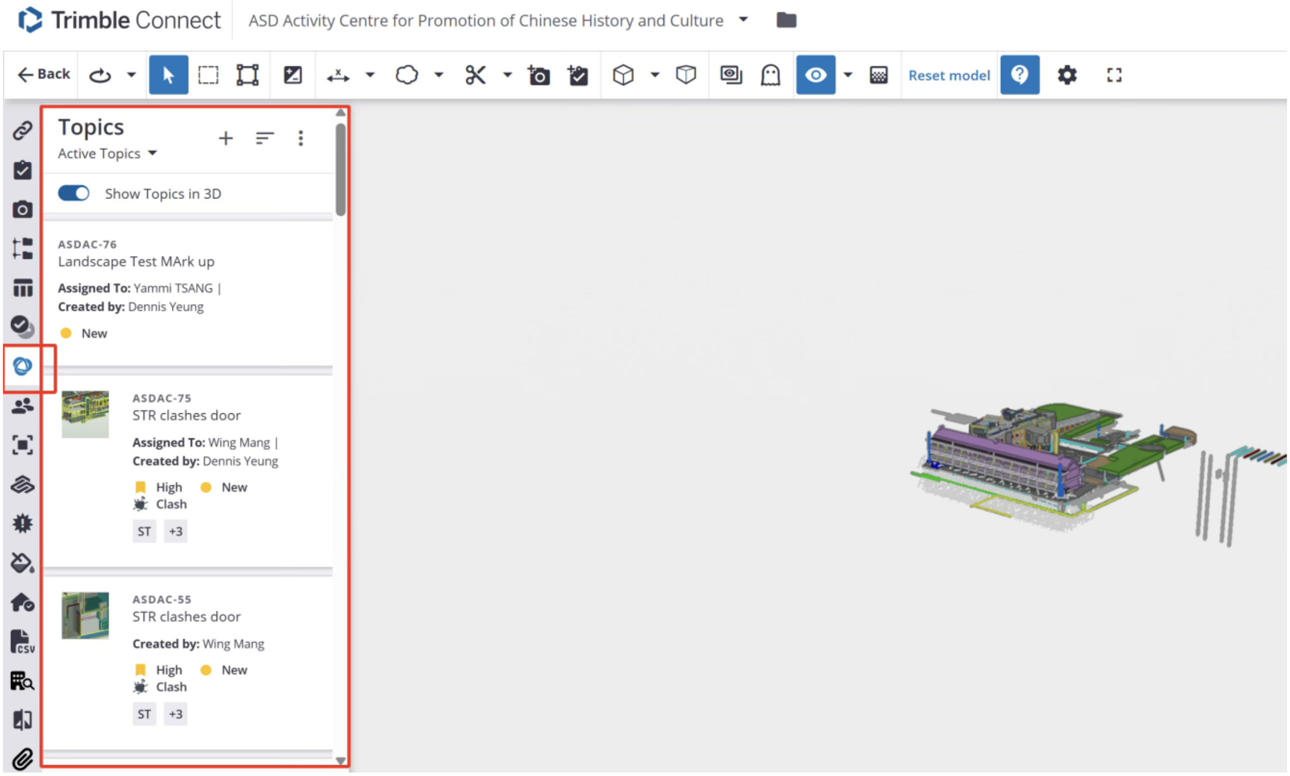
2. Create a New BCF Topic:
   - Click the plus button in the BCF Topics side pane.
   - Fill in the required fields on the New Topic page:
     - Title (mandatory).
     - Assign the topic to a user or user group if needed.
     - Add references (documents, links, or models) if applicable.
   - Click Save to create the topic.
3. Add Measurements and Markups:
   - Use the measurement and markup tools in the BCF Topics side pane to add graphical annotations or measurements to the model.
   - These annotations are saved as part of the topic view.
4. Add Comments:
   - Select a topic and click Comments.
   - Add comments, attachments, or images to the topic.
   - Save the comments to update the topic.
5. Link Model Objects:

- Select the objects in the model that are relevant to the topic.
- Click **Link selected** in the Models page of the BCF topic to associate the objects with the topic.



6. Export BCF Topics:
   - Select the topics to export and click the **Export** button.
   - Choose the file format (.bcf or .pdf) and the details to include in the export.
   - Download the exported file from the **Export history** section.
7. Import BCF Topics:
   - If you receive a .bcf file, you can import it into the BCF Topics side pane to view and manage the topics.
8. Synchronize with Trimble Connect (Optional):
   - If the model is linked to a Trimble Connect project, the BCF topics are automatically synchronized with the project.
9. Manage Topics:
   - Edit, assign, or delete topics as needed.
   - Use sorting and filtering options in the BCF Topics side pane to organize the topics.
10. Collaborate with Team Members:
    - Share the exported BCF files with other team members or use Trimble Connect for real-time collaboration.

This workflow ensures efficient issue tracking, communication, and resolution within a project.



C.3.2. Working with online BCF server

1. Automatic syncing

Tekla Structures supports synchronization with online BCF servers for real-time collaboration. Teams must establish regular synchronization intervals (e.g., daily) to ensure coordinated updates across disciplines. Choose a BCF server (such as BIMcollab or BIM Track) that seamlessly integrates with Tekla's BCF manager for enhanced productivity.

Tekla Structures allows users to work with BCF (BIM Collaboration Format) topics using an online BCF server.

If you are looking to manage BCF topics in Tekla Structures, you can use the **BCF Topics** side pane, which integrates with Trimble Connect. This allows you to create, edit, assign, and track issues in a collaborative BIM workflow.

2. Manual import and export

The manual import and export workflow using the BIM Collaboration Format (BCF) is essential in Tekla Structures to enable seamless collaboration with project team members using different BIM platforms. This workflow allows project issues, comments, and status updates to be exchanged effectively, even without a direct real-time server connection.

Workflow Steps for Manual BCF Import

Step 1: Downloading Issues from Online Server

- Log into the online BCF server platform.
- Navigate to your project workspace and select relevant issues or issue lists for download.
- Export selected issues in `.bcfzip` format to a local or network directory accessible to Tekla Structures users.

Step 2: Importing BCF into Tekla Structures

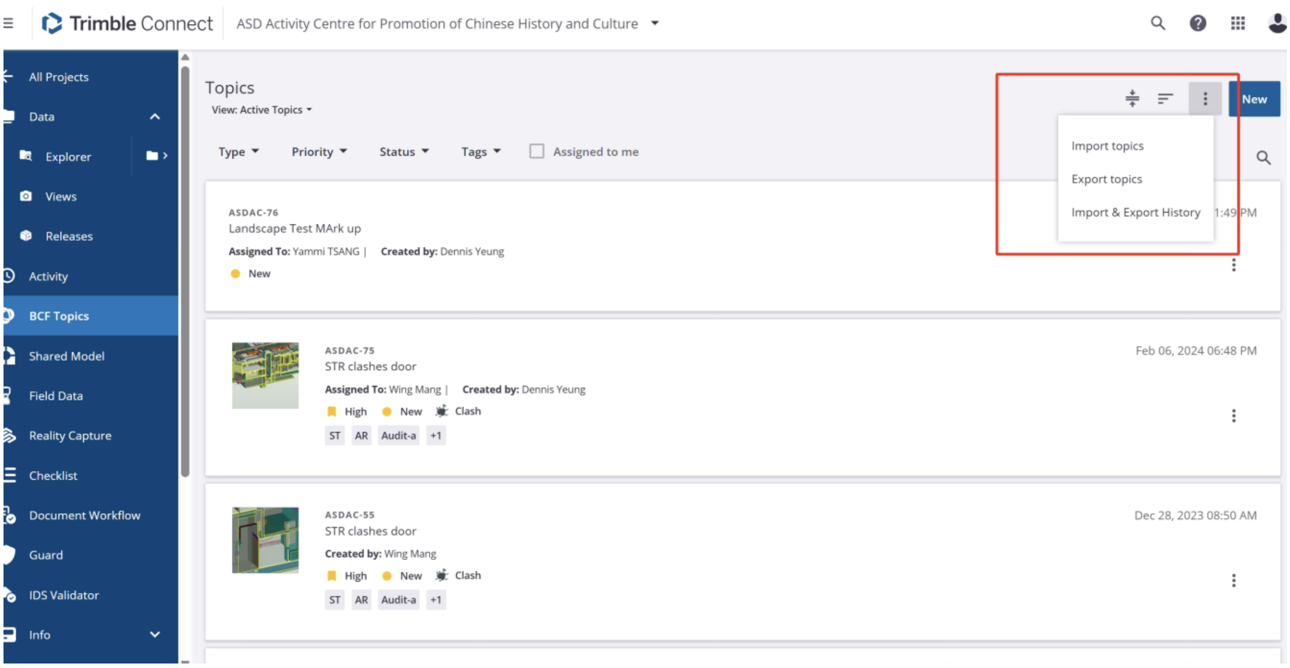
- In Tekla Structures, open the Issue Manager:
    - Select the Import option, and browse to the downloaded `.bcfzip` file.
    - Confirm the import selection. Tekla Structures imports and processes all included issues.
- Issues appear directly in the Tekla Structures Issue Manager with original viewpoints, annotations, and status data maintained from the server platform.

Step 3: Resolving and Updating Issues in Tekla Structures

- Review imported issues thoroughly within Tekla Structures.
- Make necessary model adjustments, structural revisions, or provide additional information as responses.
- Update issue statuses accordingly (e.g., change from Open to Resolved) and document solutions within the Issue Manager.

Step 4: Re-export and Re-upload Issues

- Once issues are addressed, export the updated issues again as described previously in the Exporting Issues as a BCF File section.
- Upload the updated `.bcfzip` file back to the online server for synchronization and visibility by all project stakeholders.



Recommended Best Practices

- Adopt a consistent naming convention for BCF files to maintain clarity and ease of tracking (e.g., `[Discipline]_Issues_[Date].bcfzip`).
- Clearly document the frequency of manual BCF exchanges in the project's BIM Execution Plan (BEP) or Exchange Information Requirements (EIR) document.
- Regularly verify data integrity after manual exchanges to ensure viewpoints, markups, and associated metadata remain accurate and actionable across all platforms.

Limitations and Considerations

- This manual method requires disciplined coordination efforts, clear responsibilities, and regular communication among team members.
- Real-time synchronization is not available, potentially introducing a time lag in issue tracking updates.
- Manual workflow is effective for smaller projects or those with clearly defined coordination intervals but may become cumbersome for larger projects with extensive, frequent coordination needs. For such projects, direct online integration with the BCF server is highly recommended.

C.4. Special workflow among disciplines

Tekla Structures models are central in structural and construction detailing. Coordination with architectural and MEP teams must follow structured workflows:

- Structural IFC models provided as base models for architectural detailing.
- Continuous clash checks with MEP to coordinate penetrations and services integration.
- Regularly updated IFC files for external referencing and integration into federated models.

C.4.1. Underlaying dwg

DWG underlays from other disciplines (architecture, MEP) can be imported into Tekla Structures as reference layers to guide structural detailing. Ensure DWGs are placed accurately according to IFC Site coordinates to maintain cross-disciplinary alignment. DWG underlays are for reference only; Tekla Structures native geometry should not trace directly from DWGs to maintain model accuracy and IFC fidelity.

To use a DWG file as an underlay in Tekla Structures, you can insert it as a reference model. This allows you to view the surfaces of the objects in the DWG file and use it as a guide for modeling. Below are the key points and steps:

Key Points:

1. DWG/DXF Tool:
    - If you import DWG/DXF files using the DWG/DXF tool, only construction lines or lines converted to part profiles are shown. Surfaces are not displayed.
    - For surfaces, you must insert the DWG file as a reference model.
2. Supported Versions:
    - Tekla Structures supports DWG files up to AutoCAD 2023 format.
3. Coordinate System:
    - Ensure the DWG file's coordinate system aligns with your Tekla Structures model. You can use base points for alignment if needed.

Steps to Insert a DWG File as a Reference Model:

1. Open your Tekla Structures model.
2. Go to the File menu and select Reference Models.
3. In the Reference Models dialog box, click Add model.
4. Browse to the DWG file you want to use and select it.
5. Adjust the insertion point, scale, and rotation as needed to align the DWG file with your model.
6. Click OK to insert the DWG file as a reference model.

Advanced Options for DWG Files:

- XS_DWG_IMPORT_IGNORE_UNITS:
    - Use this option to control how units are handled when importing DWG files. For example, set it to TRUE to ensure all coordinates are interpreted in meters.
- XS_IMPORT_DWG_TEXT_AS_POLYGON:
    - If you encounter font issues in DWG files, set this option to TRUE to import text as polygons instead of fonts.