[**BLD227223**]

Dynamo and Parts to leverage design BIMs for downstream construction and FM workflows

Marjan Sadeghi
Colorado State University

Jeremy Tolmen
PCL Construction

Learning Objectives

- Identify the inefficiencies of fragmentation in BIM workflows and the fundamental concepts of connected BIM from a project lifecycle standpoint
- Leverage connected-BIM workflows through computational design with AEC Collection
- Utilize Dynamo to script in Revit API to verify and augment design models for downstream construction and FM workflows
- Create and utilize Revit Parts to maintain control over the content and convention of linked models

Description

This class provides insight on concepts of a project-lifecycle view towards connected BIM within the AECFM industry. You will be introduced to a foundational framework that, exploring computational modeling, leverages design models for downstream construction, handover, and operation use. This includes using parametric conditioning to augment the Level of Development (LOD) and modeling conventions according to your specific BIM purposes. Utilizing such powerful tools as Revit Parts and Dynamo (to script in Revit API), efficient solutions will be provided to streamline collaboration through AEC Collection and augment the content and convention of linked models to avoid remodeling. We will showcase how you will be able to establish model-based workflows for such tasks as constructability reviews, clash detections and management, scheduling, estimating, building information hand-over, and operation. You will also be able to explore other implementations that align with your own practices.

Speaker(s)

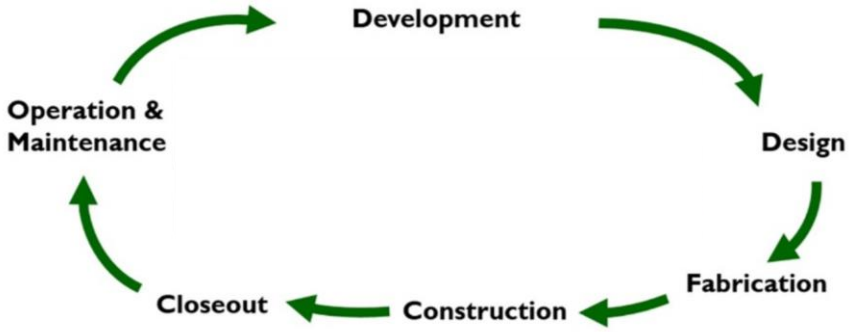
Marjan Sadeghi (msadeghi@colostate.edu) is a PhD candidate and Graduate Teaching Assistant in the Construction Engineering and Management program at Colorado State University. In her research, she focuses on development of BIM-intensive workflows for building lifecycle information management. She is also a Virtual Construction intern at PCL Construction, where she is engaged in streamlining connected BIM with AEC Collection. Exploring computational modeling, she utilizes the Dynamo platform to develop Python scripts that interact with the Revit

API to automate model development and verification for efficient downstream construction, handover, and operation implementation.

Jeremy Tolmen
As the department head of Operations Support at PCL Construction – Denver district, Jeremy Tolmen (jtolmen@pcl.com) directs the integration and collaboration between all preconstruction and operations processes and directly oversees the virtual construction (VC), surveying/reality capture, scheduling, quality and lean departments and services. Complementing Jeremy's BIM experience are years of onsite construction management and preconstruction activities on a wide variety of project types. Combining his BIM and field experience with his passion for quality, efficiency, and innovation provides him with the proper tool set to streamline and integrate otherwise segregated processes and systems within the AEC industry.

Design-, vs. Construction-,vs. FM-intent BIM

- The Architecture, Engineering, Construction, and Facilities Management (AECFM) is an information-intensive but mostly fragmented industry
- Multiple stakeholders involved in various stages of a building's lifecycle have different information requirements (both in terms of content and format)



BUILDING LIFECYCLE INFORMATION FLOW

- (Computational) BIM TO THE RESCUE: object-oriented, parametric, metadata – beyond pure visualization
- BIM-based building lifecycle information management and decision making

Design-intent BIM	Fabrication BIM	Construction-intent BIM	FM-intent BIM
Design code compliance		Construction sequence	Facility repository
Design alternatives		Construction Cost	Maintenance Schedule
Design coordination		Logistics and site layout	Space Management
Design analysis		Building Info. handover	Renovations

- BIM requirements for each implementation: building systems, geometry, semantics
- BIM interoperability issues: software interoperability, exchange protocols, common understanding of the content
- Resulting inefficiencies in BIM workflows:
  - Over-, under-, and re-modeling
  - Comprehensive, accurate, reliable information

➔ There is the need to bridge this gap

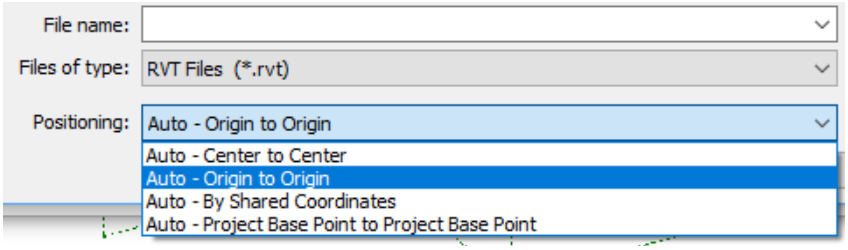
Connected BIM from a project lifecycle standpoint

- BIM Execution Plan: The project BIM standard
- Early involvement of BIM end users:
    - Determine project-specific BIM implementations.
    - Set model requirements.
    - Map the process: develop, verify, implement, exchange, maintenance.

Design-intent BIM:

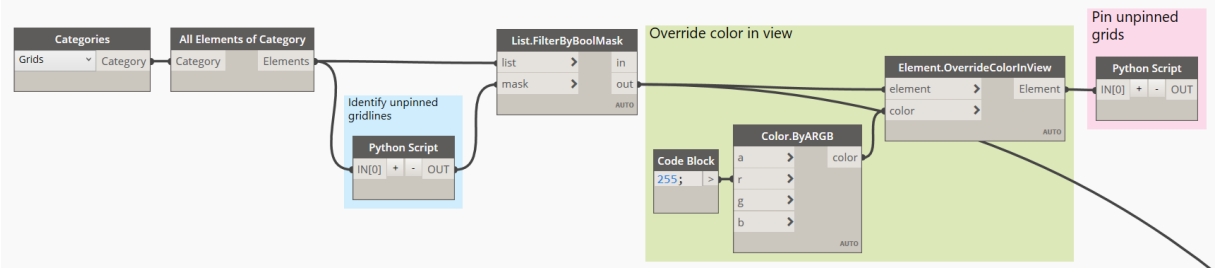
Project-level requirements:

Project coordinate system



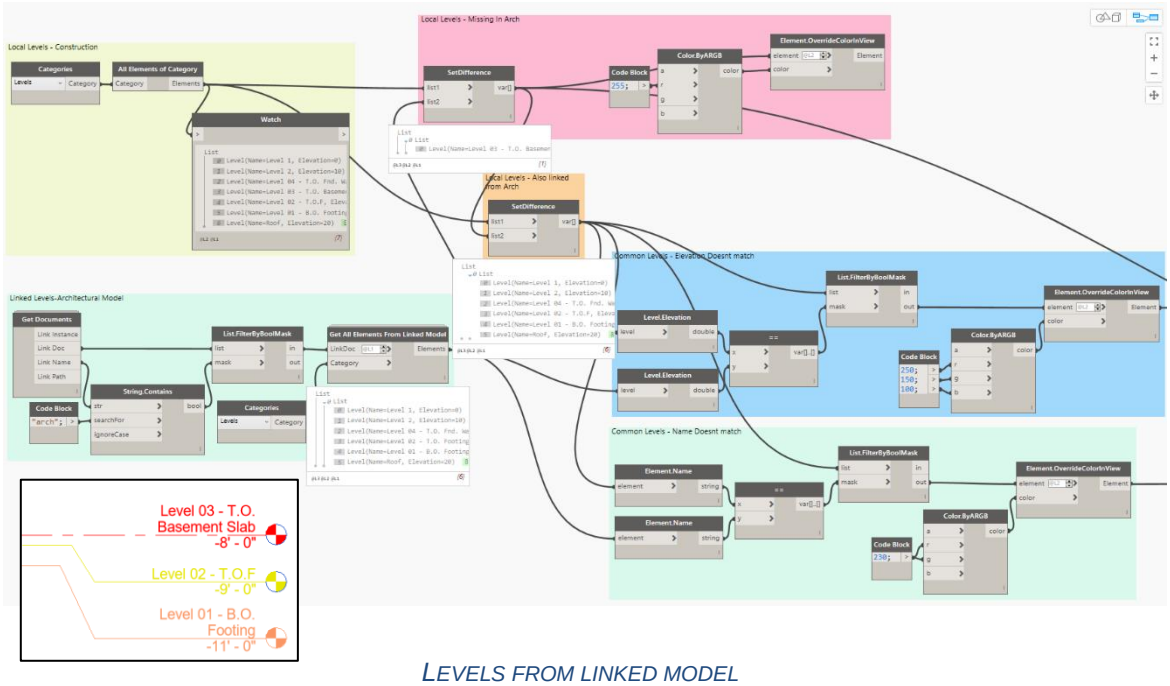
REVIT LINKED MODELS - OPTIONS

Shared elements – Gridlines



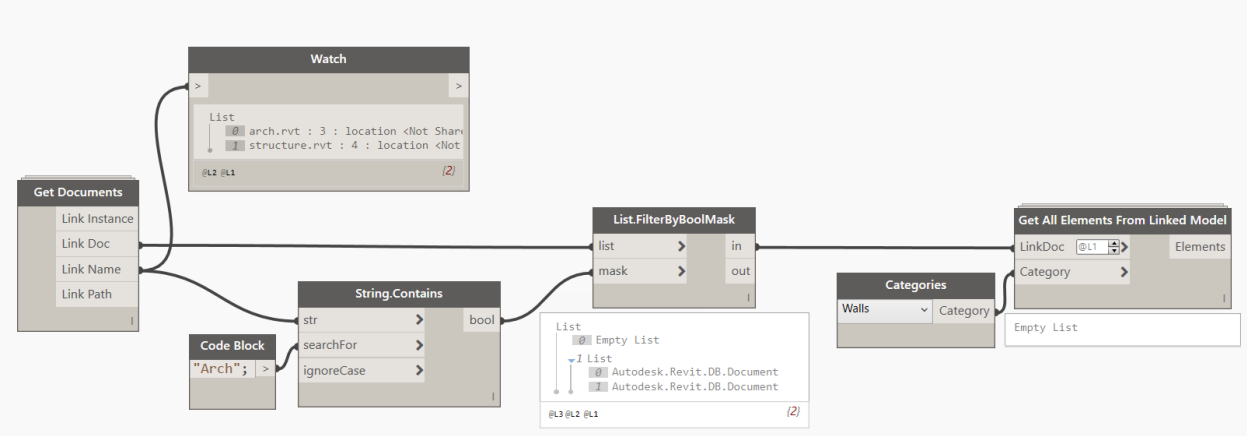
IDENTIFY AND PIN UNPINNED GRIDLINES

Shared elements - Levels



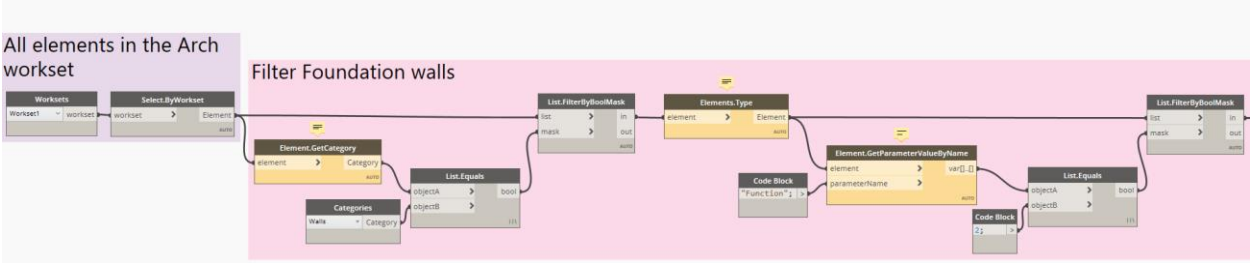
LEVELS FROM LINKED MODEL

Naming Convention – Linked models naming



STANDARD NAMING FOR LINKED FILES

Discipline-model content:



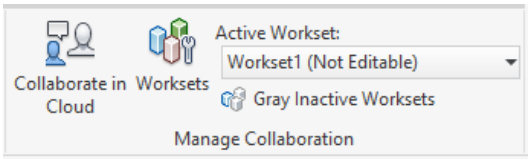
FOUNDATION WALLS MODELED IN THE ARCHITECTURAL MODEL

* Null values cause warnings, but the graph works fine for the elements with value

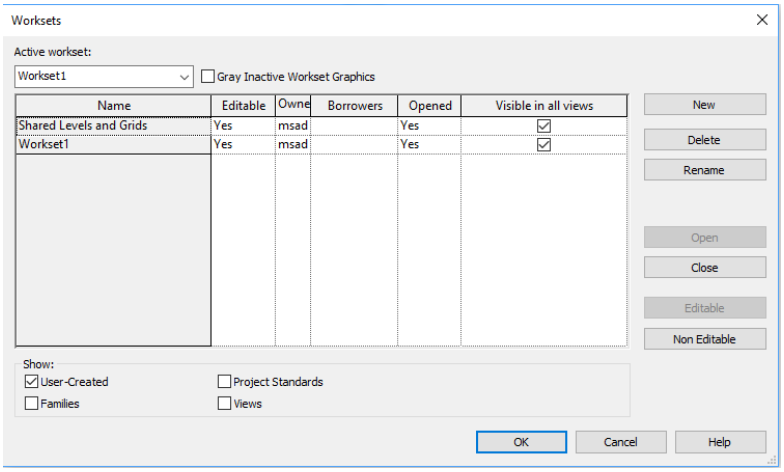
Model content organization

Worksets:

- Enables file-based or cloud-based worksharing
- To set up a workshared project in Revit:

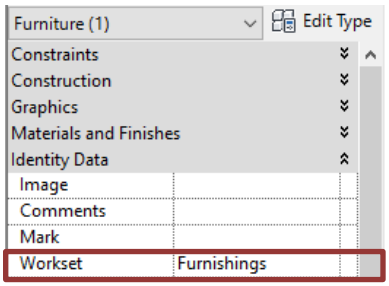


- By default, Revit creates two user-created worksets:



- Create user-defined worksets to organize discipline model elements.
- To be able to display a workset from a linked model in a host view, the workset must be open when you load the linked model into the host model.

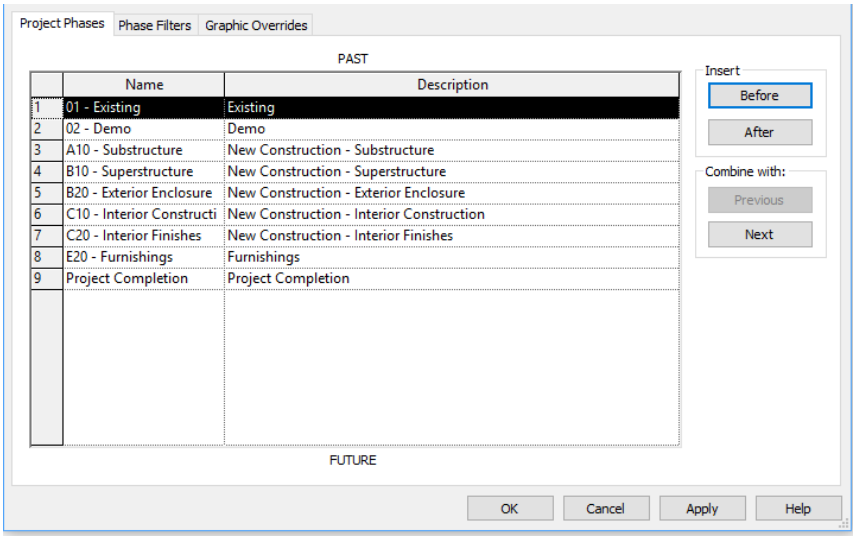
- Workset is an instance parameter in Revit. You can edit the value if required.



- You can filter elements in Dynamo by their respective workset:

Phasing

- Each phase represents a distinct time period in the life of the project
- Each model element has a “Phase Created” and “Phase Demolished” parameter
- Each view in Revit has a “Phase” and a “Phase Filter property”



DEFAULT PHASES IN REVIT CONSTRUCTION TEMPLATE

Content requirements:

Naming Conventions

- Project participants need to agree on standard, meaningful naming convention for all components of the model – including objects, properties, materials, and values.
- Here is an example provided in the “BIM Object Standard”.

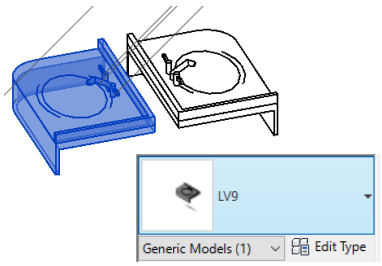
<Originator>_<Source>_<Type>_<Subtype/Product code>_<Differentiator>

Type	Description	Example
Originator	Used to convey the **object** provider by a 3–6 character code. Where an **object** is provided through an **object** library but developed by another party, include a **code** to convey the library provider.	NSWPH
Source	Used to identify the library object manufacturer. The manufacturer name shall not be abbreviated. For a generic object, this field may be omitted.	BettaWindows
Type	Used to identify the object type.	Window
Material	Used to identify the material type.	Plastic
Subtype/ Product Code	Used to convey additional information to further define the construction product such as the product range. The manufacturer product range shall not be abbreviated. This field can also be used to identify the predefined (Sub) type.	Skylight
Differentiator	Used to convey additional information required to adequately identify the **object**, or not otherwise captured in the attribute data.	600x900mm
Image type	Used to convey the image type, e.g. bump, cut-out, render.	Bump

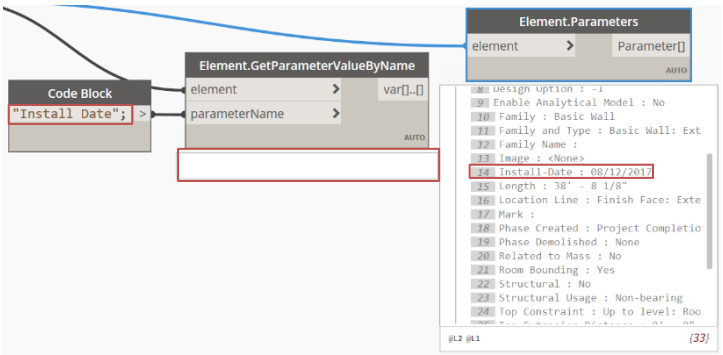
NAMING FIELDS – FROM BIM OBJECT STANDARD

- You can download the standard here:
https://www.nationalbimlibrary.com/en/nbs-bim-object-standard

- Two examples of inefficient naming conventions:



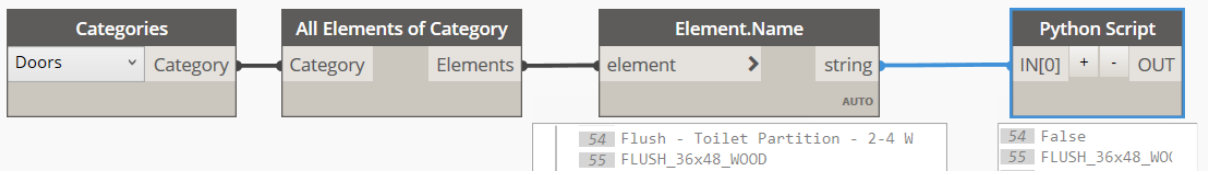
OBJECT NAMING - ABBREVIATIONS



PARAMETERS NAMING: COMPOSITION

Dynamo can verify naming patterns through regular expressions. The graph below does so assuming a standard naming has been agreed upon upfront for door elements.

"<Subtype>_<Differentiator>_<Material>"



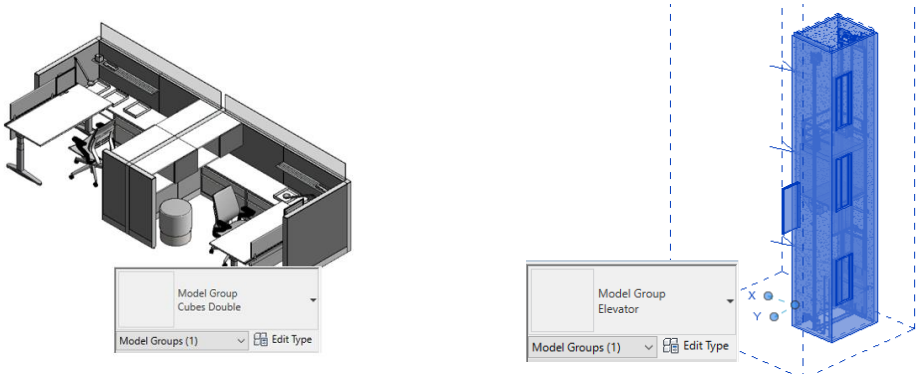
DYNAMO GRAPH TO VERIFY DOOR NAMES

Modeling conventions: Object-oriented, parametric

Groups of Elements:

Simplifies modification process during design when same group of elements are used multiple times in a project or are repeatedly used in various projects. Three types of groups are available in Revit:

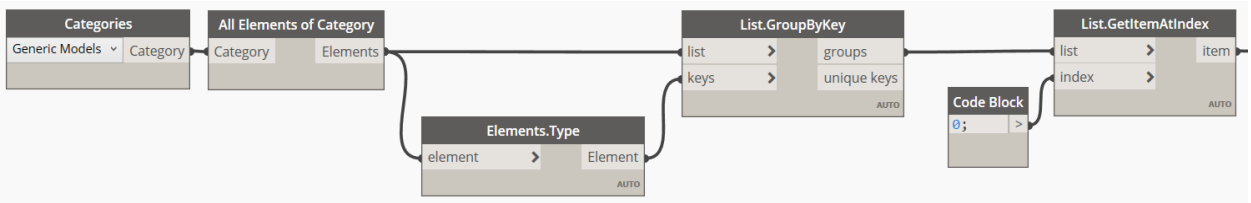
Model groups – contains model elements
Detail groups – contains view-specific elements
Attached detail groups – contains view-specific elements associated with a model group



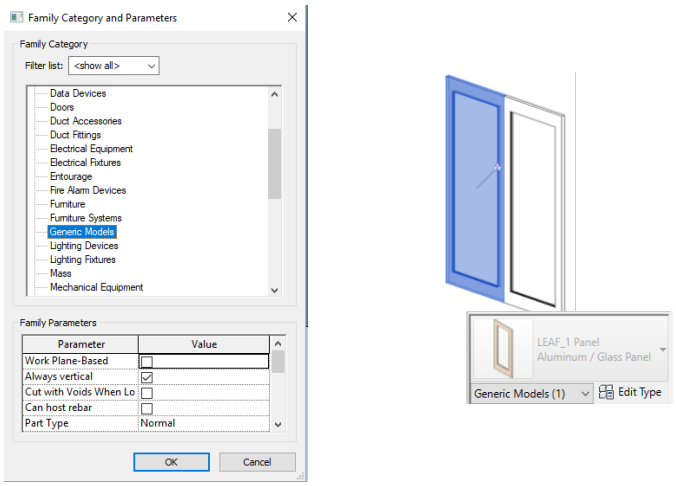
EXAMPLES OF MODEL GROUPS

Limitation: In most cases model groups are not appropriate to contain meaningful information on building elements. In the above example, various elements are included in the model group – partitions, furniture, light fixture. Not only these objects belong to different disciplines, but are of different properties – dimension, material, manufacturer, cost, maintenance requirements, etc. All (or some) of these pieces of information on each element are required for downstream model use.

Generic models:

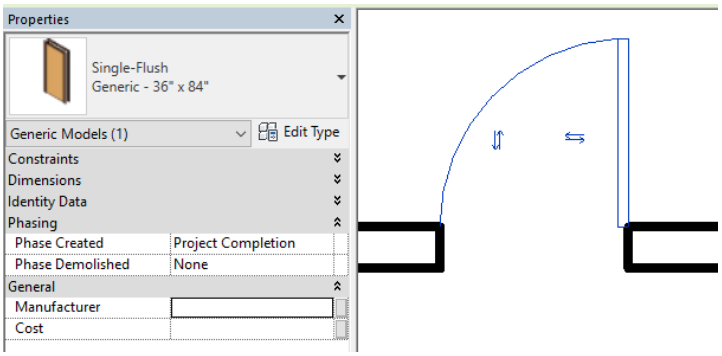


RETRIEVE GENERIC MODELS OF DIFFERENT TYPES IN REVIT



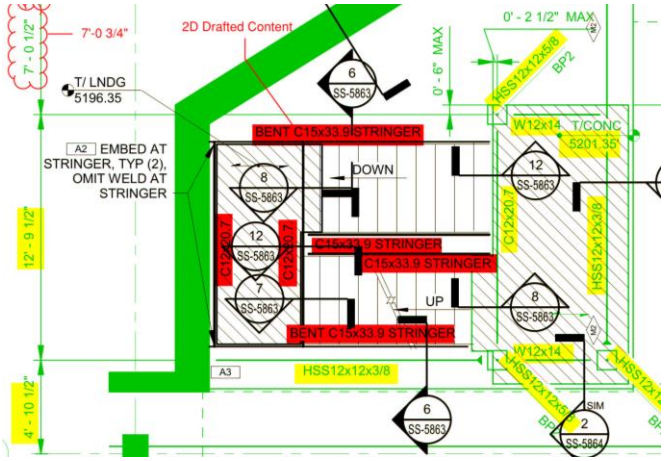
WINDOWS OF GENERIC FAMILY CATEGORY

Although, generic models can contain semantics. Following is two project parameters "Manufacturer" and "Cost" assigned to a generic door.



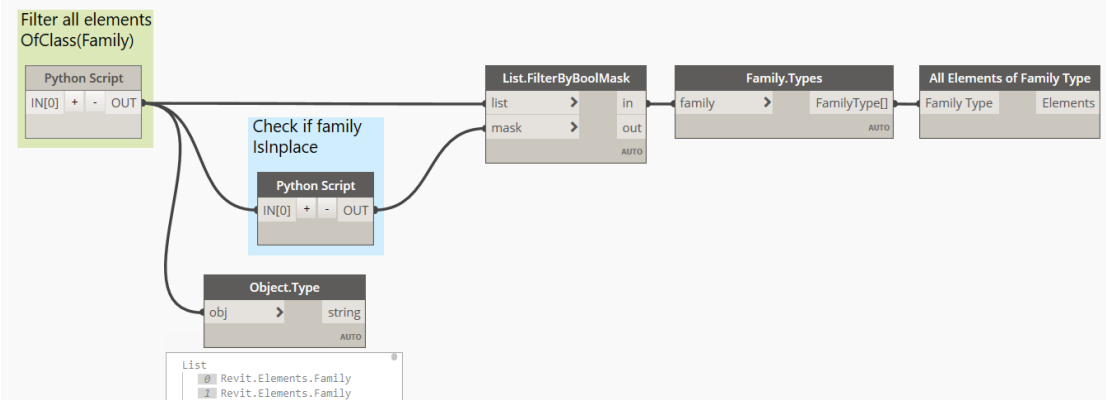
PROPERTIES ASSIGNED TO A GENERIC DOOR

Detail items:



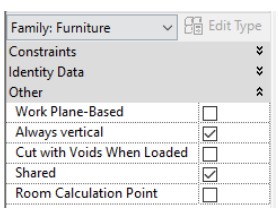
DETAIL VIEWS IN REVIT

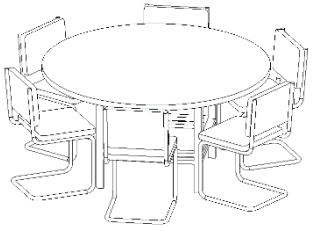
In-place families:
They don't fit into the defined hierarchy



DYNAMO GRAPH TO RETRIEVE ALL IN-PLACE OBJECTS

Nested families
Whether the family is shared or not affects quantity takeoffs



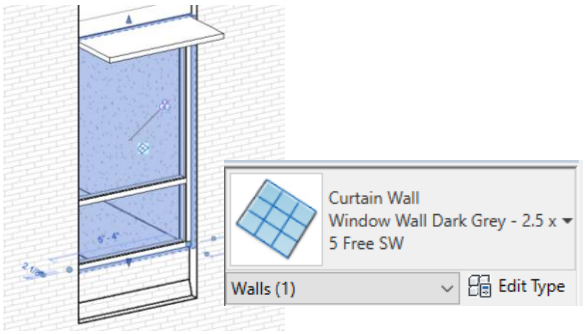


A	B	C
Family and Type	Level	Count
Table-Dining Round w Chairs: 36" Diameter	Level 1	1
Chair-Breuer: Chair-Breuer	Level 1	1
Chair-Breuer: Chair-Breuer	Level 1	1
Chair-Breuer: Chair-Breuer	Level 1	1
Chair-Breuer: Chair-Breuer	Level 1	1

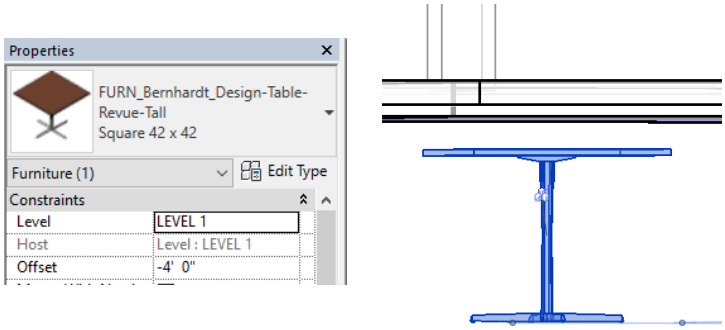
REVIT SCHEDULES - NESTED FAMILIES

Model reliability

Windows modeled as curtain walls

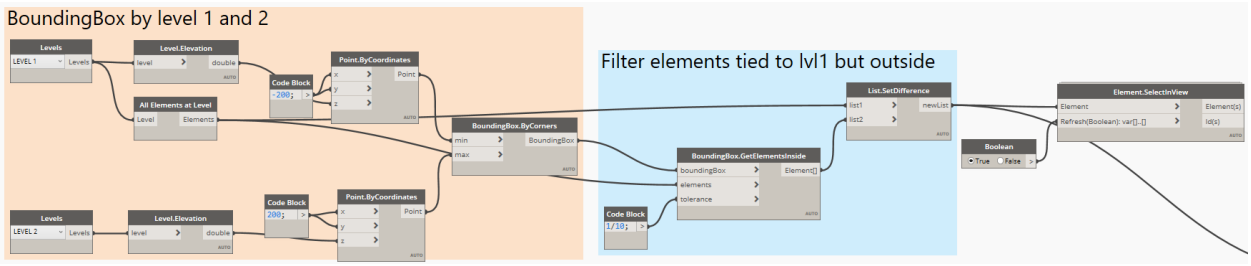


Floating objects



FLOATING OBJECTS

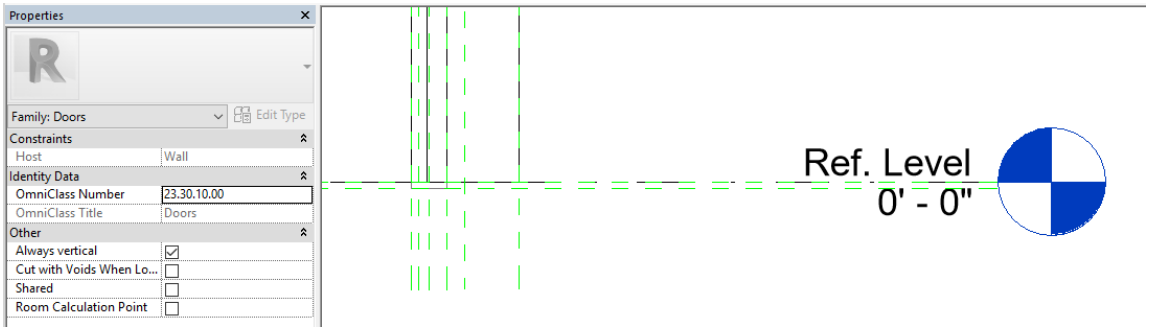
Element level:



IDENTIFY ELEMENTS LOCATED AT THE WRONG LEVEL

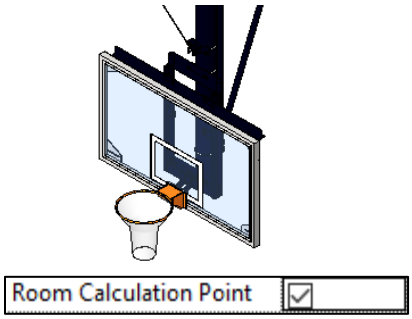
Family origin point

Running the previous graph, Dynamo graph catches this door too because of the way the family is set.



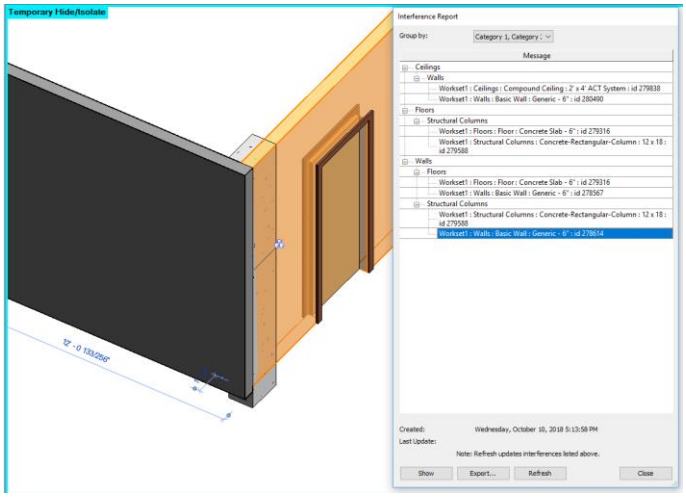
FAMILY ORIGIN POINT

Room-aware families



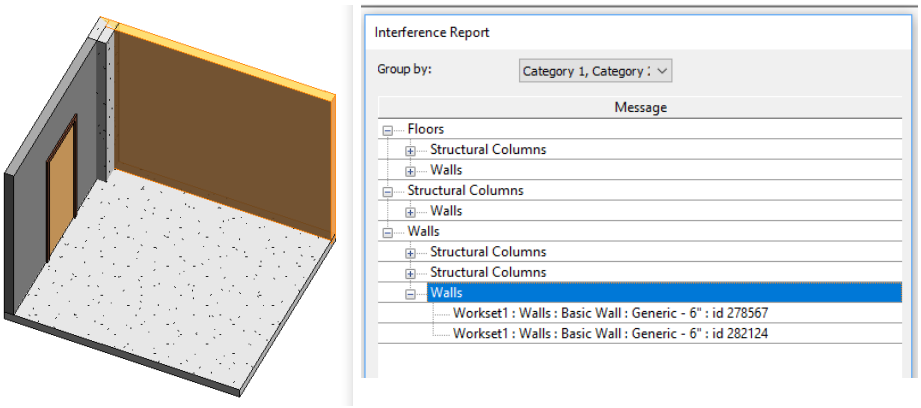
Model Accuracy – Geometry: Overlapping elements

Walls - Columns



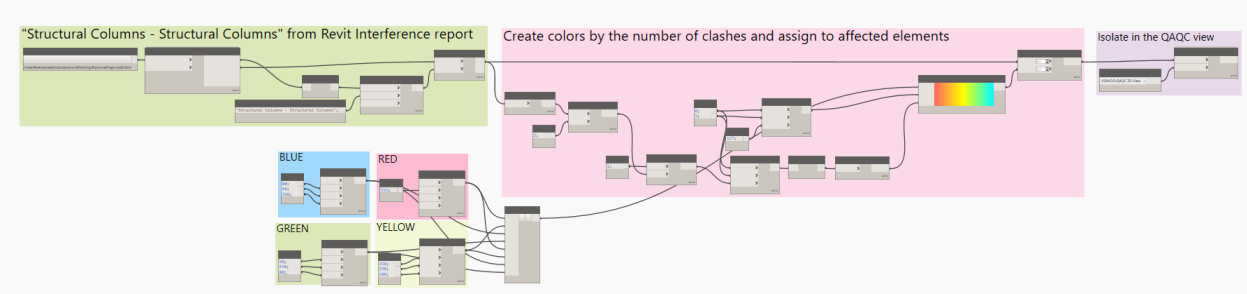
REVIT INTERFERENCE CHECK: STRUCTURAL COLUMNS-WALLS

Duplicate elements – walls



REVIT INTERFERENCE CHECK: DUPLICATE WALLS

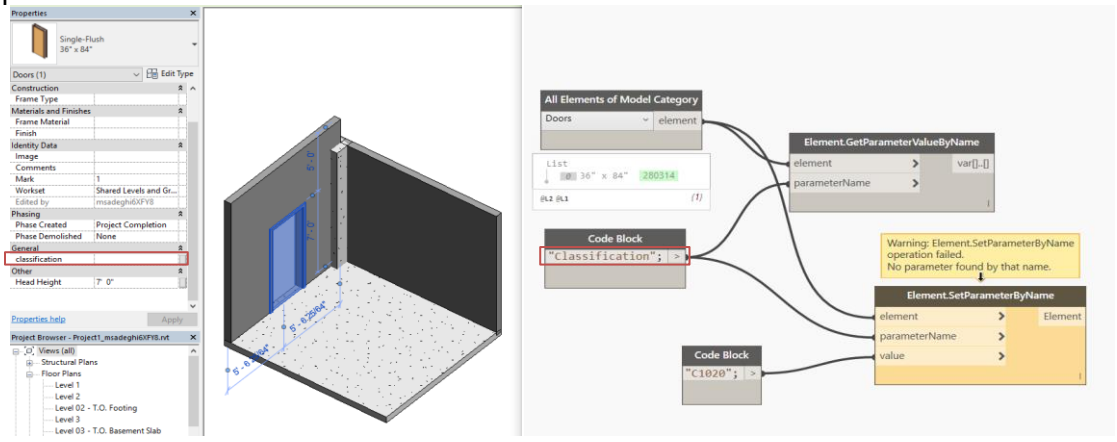
Dynamo to make Revit Interference check more powerful:



STRUCTURAL COLUMNS - STRUCTURAL COLUMNS INTERFERENCE

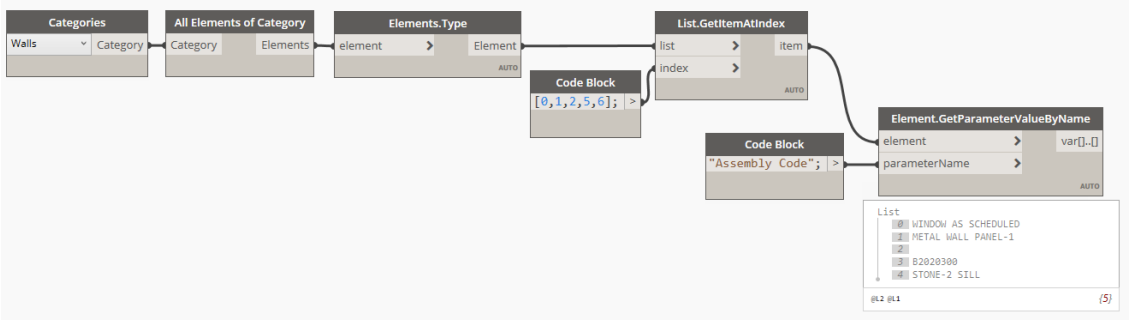
Conditioning – Classification

Typical classification standards: MasterFormat – UniFormat – OmniClass



CONDITIONING – SET PARAMETER VALUE

Watch standard parameters with overridden value:

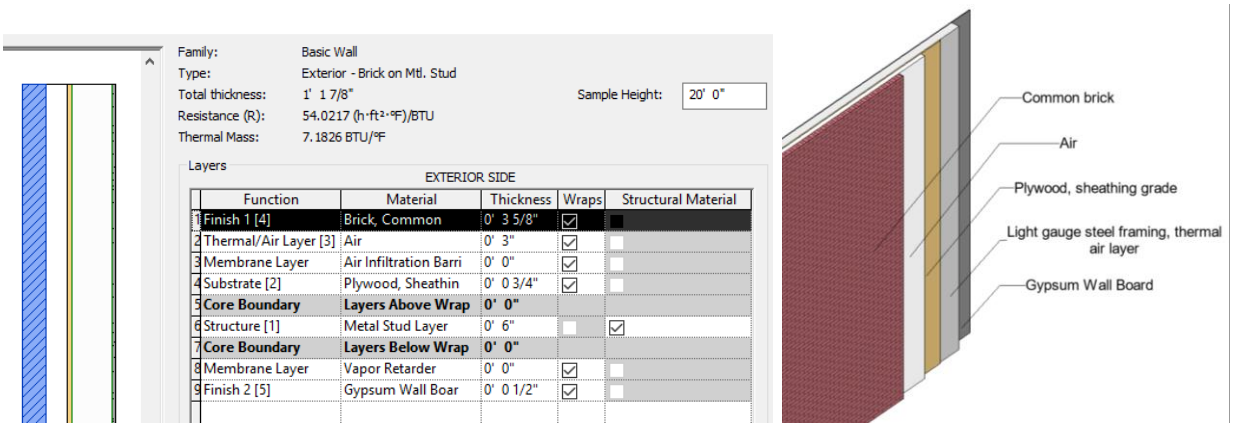


REVIT ASSEMBLY CODE: A TYPE PARAMETER

Design-intent BIM: Modification for downstream implementation

Parts - Properties

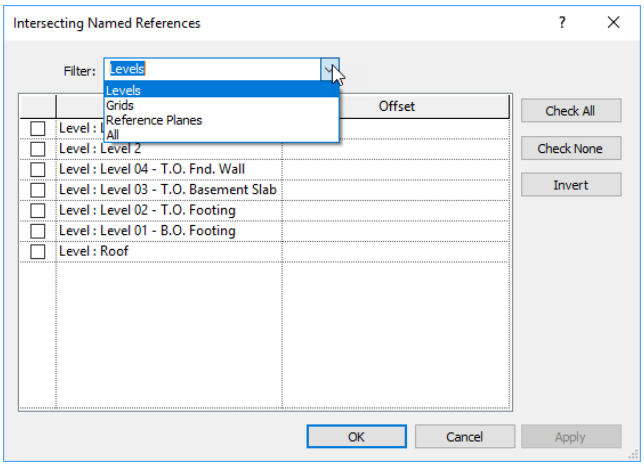
- Create Parts from certain elements (local or linked):
  Elements with layered structure: Walls (not curtain walls), Floors



REVIT PARTS - ONE PART FOR EACH LAYER IN WALL FAMILY

- Loaded or in-place families: structural framing, columns, structural columns (multi-solid made families)
  For these elements: one Part from each solid geometry in the family
  Void won't create a Part
  Model lines will never create parts
  Imported geometry in the family will create Parts (SAT works fine)
  Generic models: will create parts

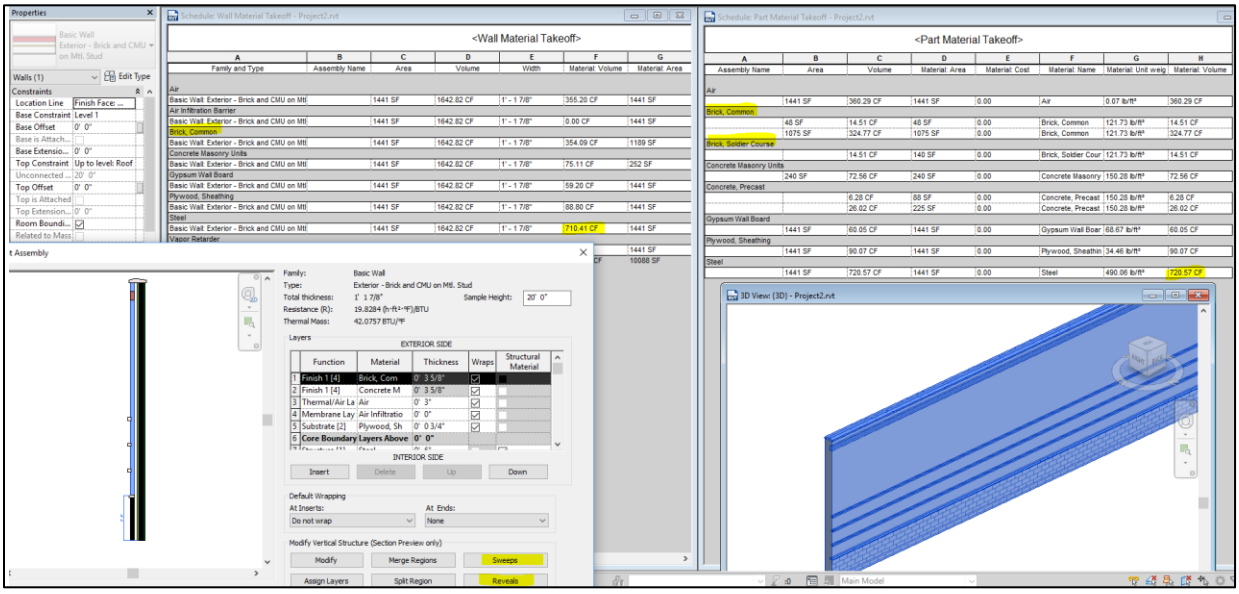
- Divide Parts by sketch/levels/grids/reference planes
  (Reference Planes should be named first)



OPTIONS FOR DIVIDING PARTS

- Delete/modify Parts with no effect on the original element
- Parts automatically update to reflect any changes to the original element (only geometry)
- Schedule, tag, filter, or export Parts independently
- Parts properties
  - Phase-aware
  - Design-option aware
  - Material (by original element)

Parts - Model-based quantity takeoff



REVIT PARTS: MATERIAL TAKEOFF