

AS322918

Using Revit in Early Design Phases

Brian DeYoung, AIA
Skidmore, Owings & Merrill, LLP (SOM)

Learning Objectives

- Develop elements and family strategies to create designs quickly and efficiently
- Use Dynamo to set-up a Revit project with levels, views and other key information
- Use groups and linked files to transfer information from one Revit project file to another
- Develop methods and best practices for working on multiple schemes, options and test-fits

Description

A common approach for many offices is to start a project in CAD and at some point, switch to Revit. This approach is highly inefficient requiring a tremendous amount of work to be duplicated. The most efficient and effective workflow is to implement Revit at the very start of projects – in concept design. However, delaying the implementation of Revit continues to be the status quo for many architects, leading to many missed opportunities, highly inefficient workflows and much longer hours. In this class, we will eliminate all the excuses for not implementing Revit at the start of projects. We will lay out a series of techniques designed specifically for Revit use during early design phases. We will demonstrate lightning fast project set-ups, core study techniques, test-fits and how to manage rapidly changing multiple schemes. With these methods and a little bit of planning we can make using Revit in early design phases fast, easy, highly efficient and desirable.

Speaker

Brian DeYoung, AIA is an architect and the Senior Digital Design Associate at Skidmore, Owings & Merrill (SOM) in Chicago, Illinois. He has been using Revit for 16 years filling the role of BIM Manager in a range of architectural firm types and sizes throughout that time. Brian is a Revit Certified Professional and leads the Revit training program in the SOM Chicago office. He has been a speaker at AIA Chicago and local Revit User Group organizations covering a variety of BIM-related topics. He is also an author for blackspectacles.com where he has contributed a video tutorial series covering a variety of Revit topics and has led interactive online tutoring sessions for the site with participants from across the United States and Europe. He enjoys teaching and helping fellow architects discover the power of Revit. Brian received his Bachelor of Architecture degree from University of Kentucky and his Master of Architecture degree from University of California Los Angeles.

brian.deyoung@som.com

Revit & Early Design Phases

Everyone taking this class has probably heard all the excuses for why teams can't use Revit in early design phases, or maybe even said one or two of them.

The Excuses

- Revit is only for documentation
- Revit demands too much detail
- The design changes too much
- There are too many schemes
- Project schedule is too tight
- Takes too long to produce drawings
- Revit is too complicated
- CAD is easier
- Geometry is too complex
- I don't know Revit very well

This class will attempt to repudiate these excuses and demonstrate that Revit can be used effectively during early design phases and in an efficient manner. The class will show that using Revit during early design phases can lead higher productivity and better BIM. (Okay, so maybe this class can't solve that last excuse, but it can be a start...)

*"What you're supposed to do when you don't like a thing is change it.
If you can't change it, change the way you think about it."*

- Maya Angelou

This quote is something of a thesis statement for this class. Revit works in a very specific way, and that way is often favors later design phase work and traditional construction documentation. In early design phases, Revit may not natively do things in the way that is needed or desired. Except for a few add-ins or Dynamo scripts, there is often not much a Revit user can do about that. Only Autodesk can change Revit. Even the best add-ins can only work with the system. So, what can be done when a thing can't be changed? The intention of this class is to change how Revit users think about Revit and the work that they typically ask Revit to perform in the early design phases. Once this change of thinking occurs, a whole new world of possibilities is born.

Class Agenda

This class will cover a variety of techniques, strategies and workflows for using Revit in early design phases in six different areas.

- 1. Have a Plan: The Big Picture**
How will Revit be used on projects, and how will it relate to the other software used?
- 2. *Keep It Simple: Designing in Revit***
How can schematic level model be built quickly and be easily editable?
- 3. *Mobilize Quickly: Setting Up New Files***
How can new project file be set-up in minutes and all files be consistent across a project?
- 4. *Don't Reinvent the Wheel: Templates***
How can teams get a head start by having all the information they need at their fingertips?
- 5. *Control the Chaos: Project Management***
How can BIM Managers and team member prevent the project's Revit files from descending into chaos and disorganization, as the design rapidly changes?
- 6. *Watch for Curves Balls: Options & Schemes***
How can teams create and manage multiple options and schemes with minimal extra work?

Have a Plan: The Big Picture

Any plan to use Revit in early design phases must begin with the idea of starting early.

Early Start

As any fan of Disney theme parks knows, the best way to experience the most popular attractions in the park without the hassle of long lines and large crowds is to get an early start. Be in the park for “rope drop,” otherwise known as the opening of the park. That one decision to arrive early, can have a lasting positive impact on the rest of day. Guests who arrive later in the day, face a park with already long lines and crowded pathways, putting them in a situation in which they can never catch up to the early risers. Likewise, starting to use Revit on day one or at least within the first few weeks of the design process can have a long lasting positive impact on the project that can often be felt all the way through construction documents and into construction administration. Projects that start using Revit later in the design process will have chronic issues throughout the life of the project and this can further contribute to the idea that Revit cannot be used for design.

Projects that start employing Revit at the end of Schematic Design or in Design Development will be constantly playing catch-up. At these later stages of a project there has already been a substantial amount of time and energy put into another software such as AutoCAD for documentation, and perhaps Rhino or SketchUp for design studies. A lot of momentum has been generated and it is difficult to stop the train and switch tracks.

To one day flip a switch and say the project is moving to Revit is unrealistic and costly. The effort to bring a Revit model up to the level of documentation found in the previous software is huge, and the project schedule rarely accommodates this effort. In fact, the project schedule continues, putting the team in the difficult position of needing to produce drawings, while simultaneously building a new Revit model from scratch.

This often results in teams with one foot in AutoCAD and one foot in Revit, and the transition never seems to be made fully. This leads to work being duplicated in AutoCAD and Revit, or the Revit model being heavily subsidized by AutoCAD links, which makes the Revit model less useful.

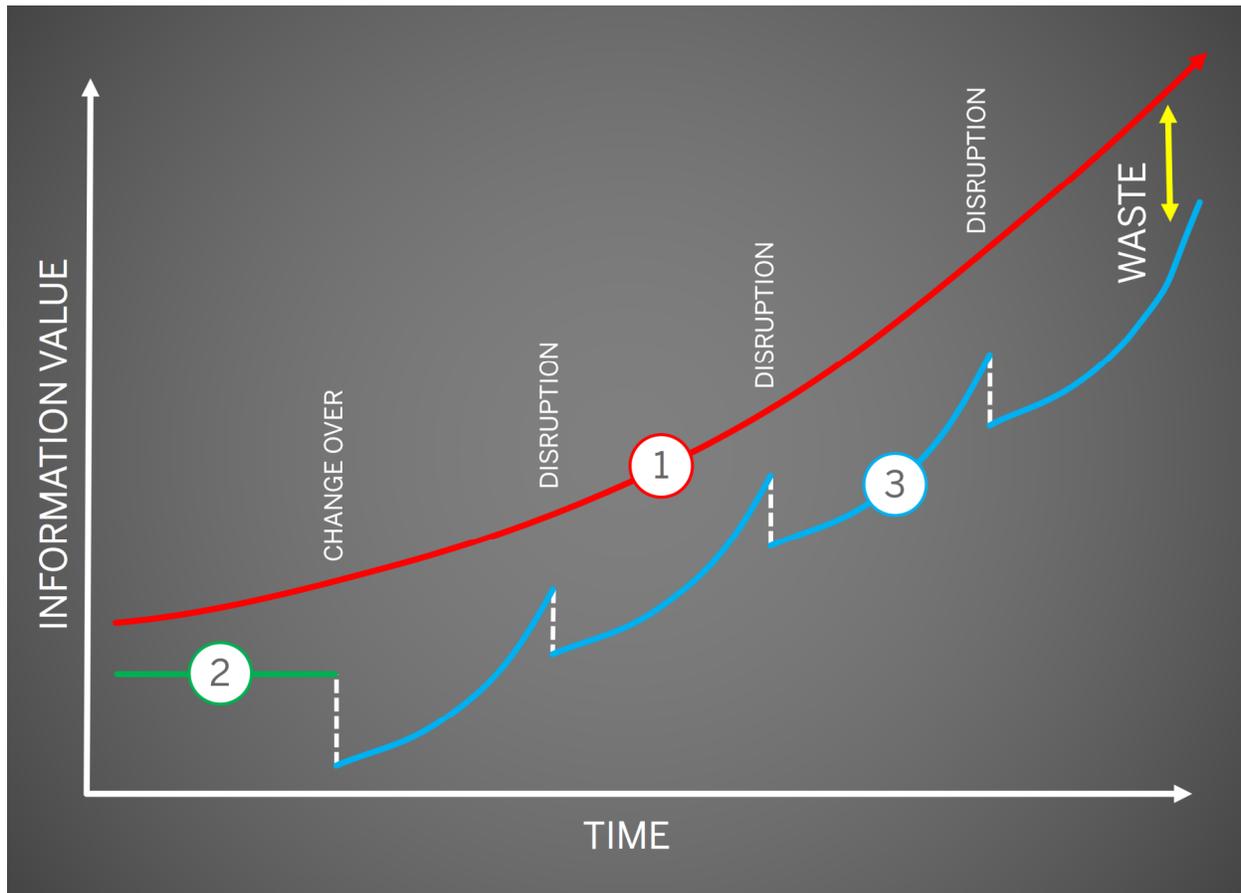


Figure 1: BIM information value over time

In Figure 1, The number 1 represents a single federated BIM developed over the course of a design process. The model begins with the program and the client’s desires. It slowly grows as the design moves forward and more becomes known. All of the work is produced from the Revit model. There is no switching back and forth between software platforms. Of course, there may be setbacks and diversions along that way, but it is a consistent process with few distractions. The number 2 represents a project that started in AutoCAD or another software. The number three 3 represents the same project that started in CAD, but is now transitioning to Revit. What results is a discontinuous BIM process that is constantly plagued by disruptions. Disruptions such as the client demanding information that is not yet modeled in Revit, so the team must go back to the previous software to produce a drawing. Disruptions such as the need to take time out of the tight project schedule simply to build up the Revit model to the level of the previous software. Disruptions to go back to another software to produce renderings because the Revit model is still not caught up to where the original visualization model is. With each disruption, either the Revit model, or the project schedule is setback, time is wasted and work is duplicated.

The practice of starting a project in one platform and transitioning to another is not a sustainable practice. To avoid these problems, teams must start using Revit for key deliverables as early in the design process as possible. However, no software is good at everything. Teams need to find a way to work with multiple platforms in an integrated way, especially in early design phases.

Have a Plan

In any discussion of Revit's use in early design phases, it must be made clear that Revit is not being proposed to replace all other software. Revit is just one piece of the design puzzle. However, in most cases it should play a central role and take on more responsibilities than other platforms. Revit, as with any software, should be used for its strengths.

In early design phases Revit should be used for massing studies, documenting floor plans, color floor plans and diagrams, room and area calculations, scheduling quantities and potentially other data related exercises. It should be the primary source for documentation from the beginning.

In a typical scenario, Rhino or SketchUp may be used for design explorations, Revit for documentation, 3D Studio for visualization and any number of other software for a variety of purposes. The mistake that many make is not developing and maintaining workflows to avoid the inevitable duplication of work between platforms. This duplication leads to lost time (and profit) on projects.

There should be a single source of truth for all pieces of a project. In other words, if the exterior wall is designed in Rhino, then rather than recreating that exterior wall from scratch in Revit, the team should develop a method of translating the critical exterior wall information into Revit and use that information to generate the exterior wall in Revit. There are many methods for doing this such as Dynamo scripts developed by the team, Rhynamo, Geometry Gym, Proving Ground's Conveyor or Speckle just to name a few. If the design changes, the change is made in Rhino and then updated in Revit through one of these methods. Likewise, something like a core may originate in Revit and need to be translated into other software. It is crucial that teams establish and manage tools for geometry and data replication across platforms in multiple directions rather than manually re-creating geometry.

Keep it Simple: Designing in Revit

A common reason for not using Revit in early design phases is “I don’t have enough information to build a Revit model.” Because Revit is so closely associated with documentation and construction documents many assume that all materials, assemblies and details must be resolved before a model can be started, but that is simply not true. This section is focused on a series of relatively straightforward modeling tips intended specifically for early design phases. There are not many complex workflows here. Instead the goal is to establish very simple and fast methods for modeling in Revit.

Massing

The massing tools in Revit can be used for a variety of tasks, including form finding which is a key part of early design phase work. However, in this case the focus is on using massing as a tool for spatial planning and programming.

The first step in this process is to create a basic mass cube. It should have instance parameters for the mass width, length & height, as well as a type parameter for the mass material.

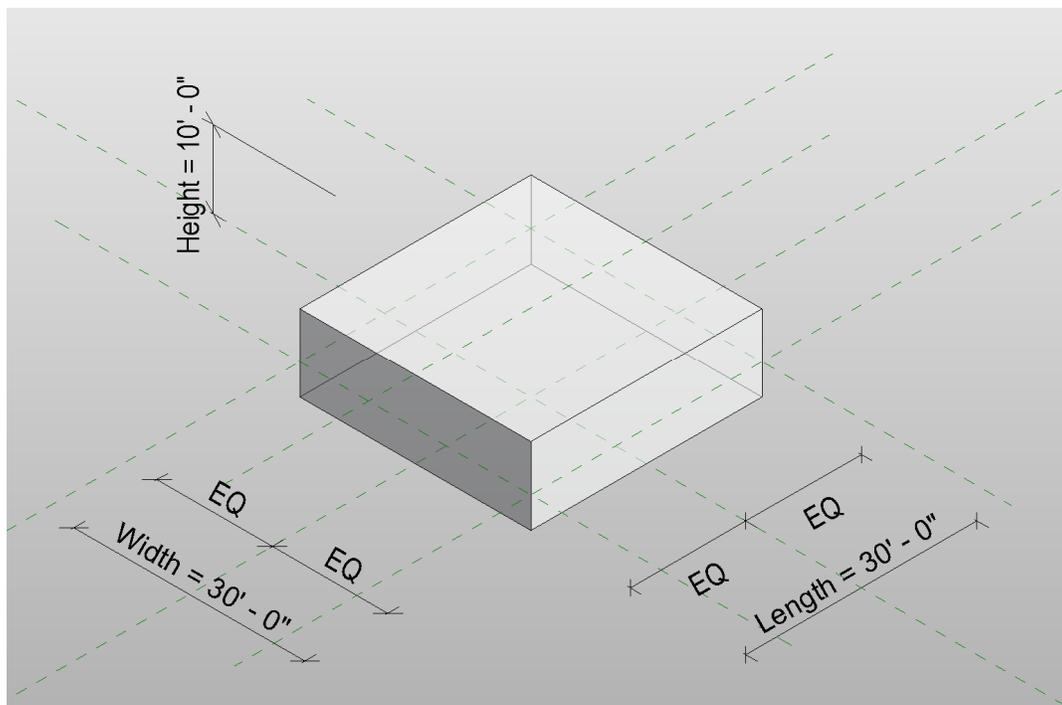


Figure 2: Mass cube family

Then establish types for each of the program elements to be defined, such as Residential, Office, Hotel, Retail, etc. Assign a unique material for each program element. This material will provide the desired color to represent each program element.

Once the mass cube family has been created with all the necessary types, it can be loaded into a project file and placed in the model. From a floor plan view the mass cubes can be stretched to fit the size and shape desired. In many cases, it may be advantageous to create an in-place mass within the project file that defines the overall extents of the proposed design. This mass can be transparent, yet provide a guide and boundaries for the mass cubes. Having an overall building mass in the project file also allows the designer to change the form of the building quickly and easily, without the need to move or reshape many smaller masses. Once the overall mass revisions are finished the smaller mass cube can be adjusted to fit the new building shape.

Mass cubes can represent entire floors or large areas that will eventually contain many different rooms. In special cases, mass cubes can be used to represent individual spaces. In the case of a hotel, one mass cube can be used to represent a single hotel guest room or suite.

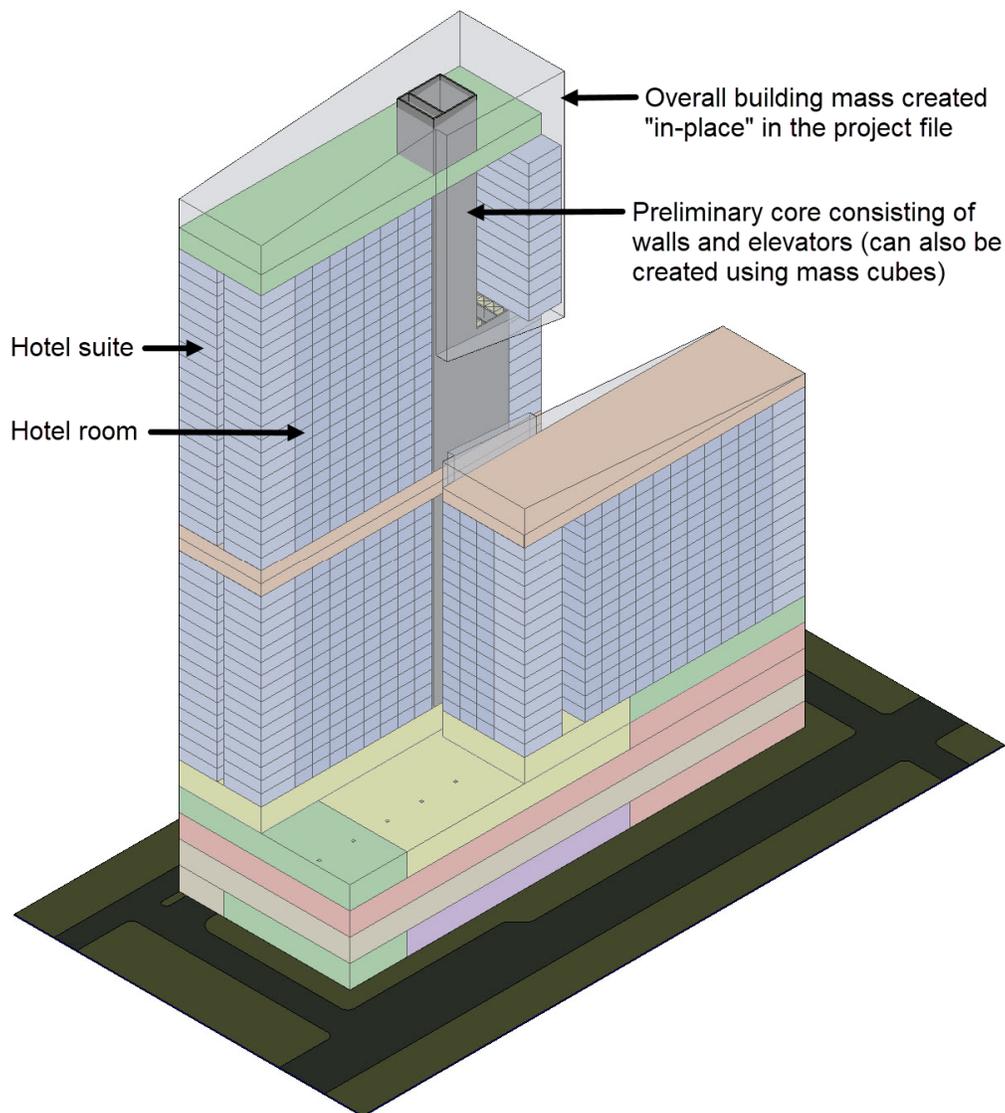


Figure 3: Mass model for a hotel

When all the mass cubes have been placed, they can be used to calculate areas, and if they are used for individual spaces such as hotel guest rooms, they can be used to count how many guest rooms and suites are in the hotel design.

However, the process to calculate the masses is not as straight forward as it might appear. Revit does not allow the level of a mass to be scheduled, so another type of schedule must be used.

Using a Mass Floor schedule, each mass can be located by level, areas can be calculated, and counts can be generated. However, the first step is to generate mass floors for each of the small mass cubes. The overall building mass must be ignored.

Gathering Mass Data

1. In a 3D view, select all of the mass cubes in the project, making sure not to select the overall building mass.
2. Click on “Mass Floors,” select all the levels that contain mass cubes and click OK. A mass floor for each individual mass cube will be generated.
3. Start a Mass Floor schedule. Add the following Fields to the schedule:

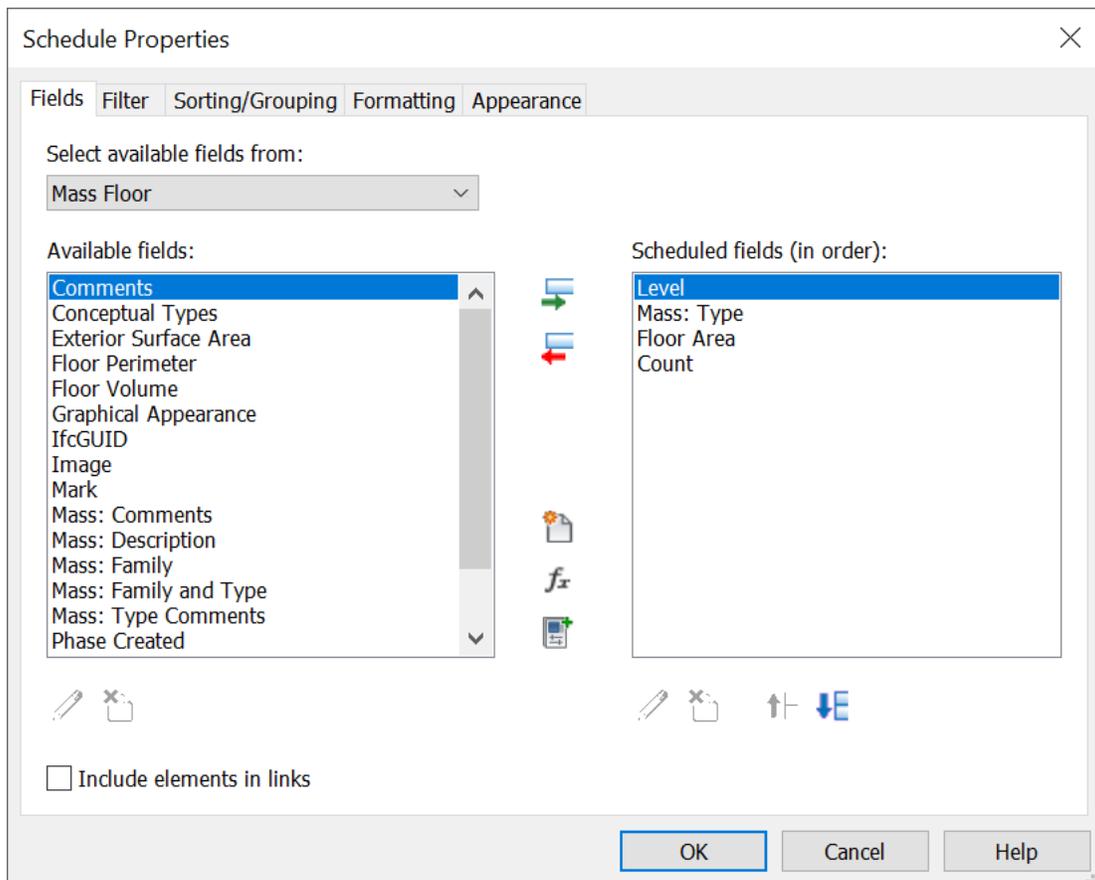


Figure 4: Creating a mass schedule

“Mass:Type” will identify each mass by type. If each mass type is a program element, such as Office, Retail, Residential etc., the total areas for those mass types can be generated and sorted by level.

If the mass types are used for individual areas, such as hotel rooms, not only can the area of each room be generated, but the “Count” field will count the number of hotel room masses in the model by level and in the building as a whole.

Standardizing Color Schemes & Masses

Massing exercises are typically one of the first things a team will do in a file. As part of the exercise, plans and perhaps 3D views are typically presented using the chosen colors for each program element. When the team moves from massing to building walls and floors, the colors from the massing diagrams should carry over to the color floor plans that will be generated, for the sake of consistency. The masses, colors and color schemes can all be easily integrated and standardized by including them in an organization’s project templates.

When developing a project template for an organization or office, load a mass cube family into the template. This mass cube family will contain all the standard program elements and colors that should be used for any project a team creates.

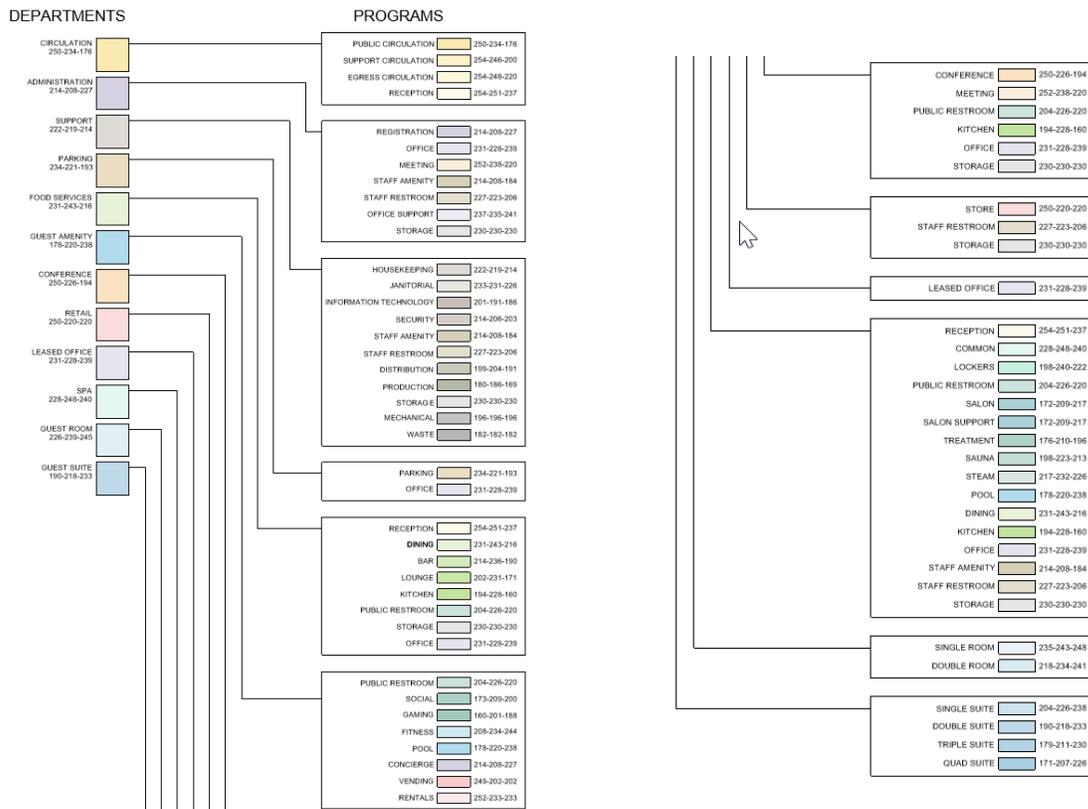


Figure 5: Example of colors to be included in a mass cube family

In addition, the same program elements and colors should be used to pre-populate color schemes in the project template. Then when a team needs to create color floor plans, all they need to do is select the appropriate color scheme from a pre-populated list - Department, Occupancy, a custom parameter, and the floor plans will color with all the same colors used in the massing diagrams.

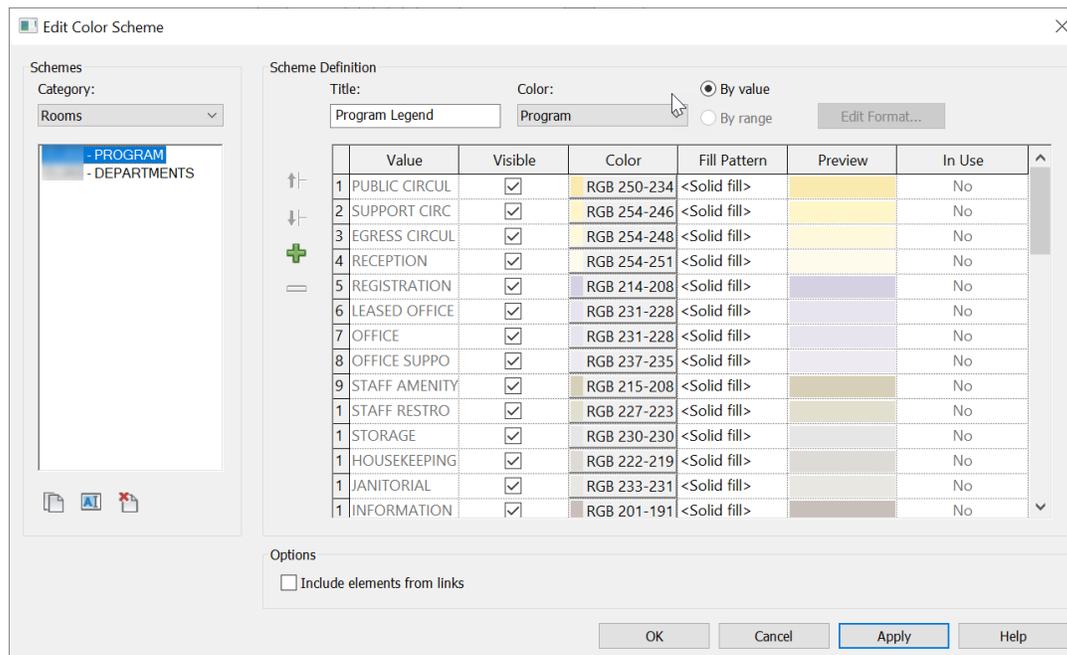
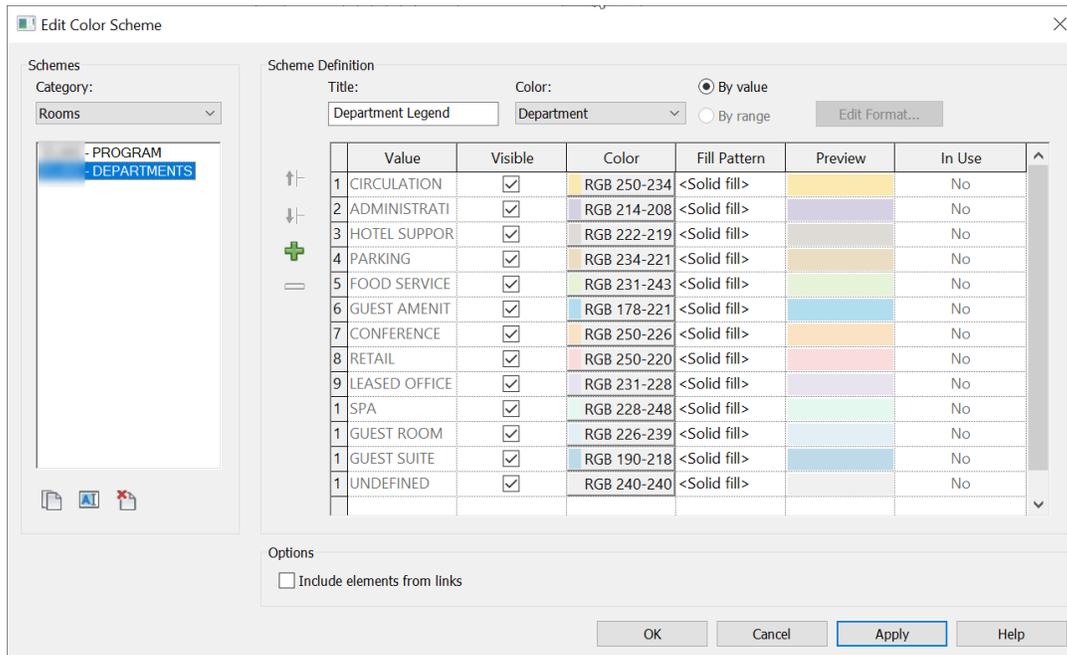


Figure 6: Color schemes using the same colors as the mass cube family

If project teams accept and use the colors pre-assigned in the project templates they will save a lot of time creating color floor plans and sections, as opposed to reinventing the wheel on every project by developing new schemes. Of course, agreeing upon a common color scheme for all projects across an entire organization can be a difficult task, as many designers have strong preferences. However, in the long run it is well worth the effort to develop a single or a single set of uniform color schemes for an organization.

Dynamo +

The use of mass cubes along with standardized materials and color schemes goes a long way to creating an efficient early design phase workflow. However, if Dynamo is added to the mix, teams can get even more use out of the model. At the most basic level, Dynamo can be used to add or edit data within the mass cubes. On a more sophisticated level, Dynamo could be used to place the mass cubes, based on any number of parametric constraints. With tools such as Project Refinery, the power of generative design could be used to develop the layout of a project.

Model Only What is Known

Focus on modeling what is known at the time. In concept design this may be massing, or perhaps where solid walls and glass is located. If that is all that is known, stick to that and don't overcomplicate the model, or include detail that is not resolved. Work with generic walls and floors, create walls and floors of the appropriate thicknesses, but without any real materials.

Level of Detail

Do not overcomplicate the model with information, geometry and details that have not yet been determined or is likely to change. Model with the appropriate level of detail. Of course, this must not be confused with the concept of "Level of Development." Level of detail is about the amount of graphical detail and information added to a model and displayed in the deliverables. Level of Development is a measure of the quality and accuracy of the information and geometry within the model. In the early design phases, the primary concern is conveying the design intent, therefore the general representation of the building is most important. Concerns of the quality and accuracy of the information typically becomes more important in later phases.

The bottom line is don't get caught up in details that will change ten times before the end of the design phase. Add the appropriate details and information as design decisions are made and the model's level of detail will naturally increase as the project progresses.

Concept Design Floor Plan

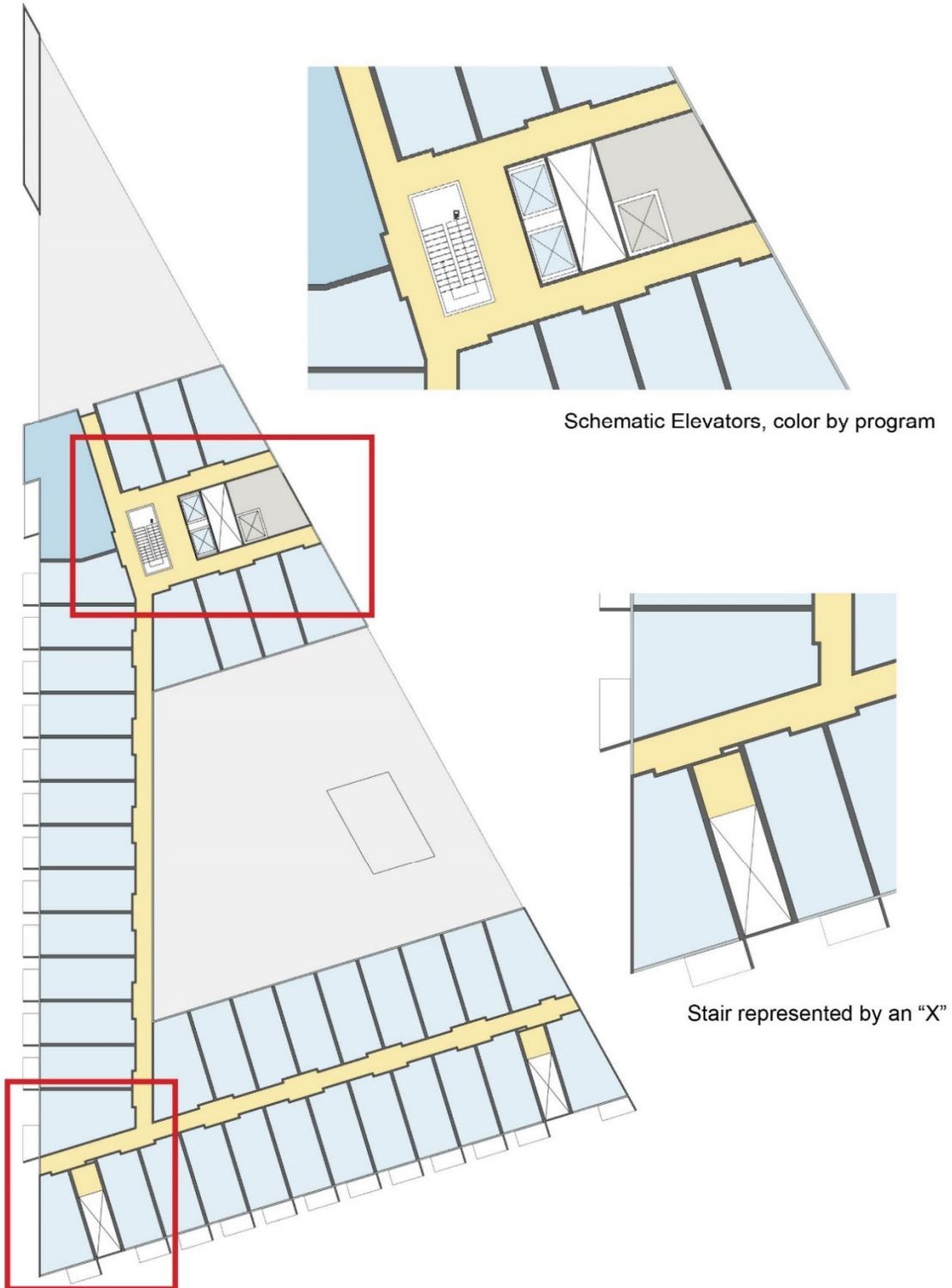


Figure 7: Concept design floor plan

Schematic Design Floor Plan



Figure 8: Schematic design floor plan

Model Only What is Necessary

In addition, to modeling only what is known, an equally important concept is to model only what is necessary. This concept is most applicable to large projects with a lot of repetition in the design, such as a high-rise tower.

If a team is working on a high-rise mixed use tower, they should focus only on modeling the levels that are being documented. For example, the team may focus their attention on modeling the ground floor, a basement level or two, a typical office floor, a typical residential floor and a few hotel floors. All the other levels of the building, even though they may be a repeat of the modeled floors, are not modeled at all.

Yes, it's easy to create groups and include the repeated floors anyway, but the team is in the early design phases and the building design is changing rapidly. Every effort to streamline the modeling process and eliminate extra clicks is important. Don't bother grouping levels and copying them to other levels. That process can make modeling more complicated and also cause the model to run slower if there are a large number of groups.

Of course, this can create a rather strange looking model with a lot of empty space, but if the team has a complete exterior wall, all those missing levels will not be visible. In the early stages of a project when the design may change frequently, don't waste time updating or maintaining parts of the project that no one will see. Later in the process when building sections are shown or data is needed from all levels of the project, a group strategy can be deployed and every level can be modeled.

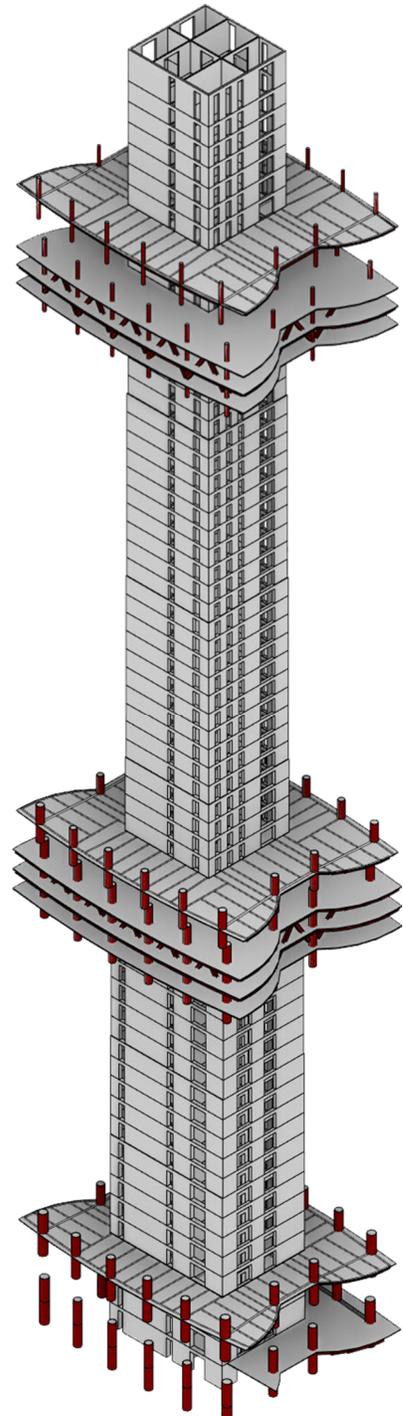


Figure 9: Structural model with only select levels modeled

Walls

A model typically begins with walls, but at the very beginning of the design process very little is known about walls, or any other element. Therefore, if only the location and thickness of a wall is known, use a generic wall (also known as a schematic wall) of that thickness and ignore the makeup of the wall and its materials. Because those materials may change many times in the early design phases. A series of wall types matching the approximate size of standard walls should be a part of every project template as they are a part of the default Autodesk templates. It's also often a good idea to make the placeholder walls slightly thicker than what they eventually will be so the designers have some wiggle room built into the design at this early stage.

For example, a 4" steel stud wall with gypsum board on both sides has a total thickness of 5 ¼", but it may vary due to other finishes or the thickness of the gypsum board used. Therefore, it may be wise to use a 6" wall as a placeholder for a wall that is assumed to be a 4" stud wall.

This idea of using generic walls is nothing new and a very common practice across the industry, but if this idea is taken to the next level, it can be much more useful. In addition to the solid wall types, consider creating schematic walls made entirely of glass. A series of walls, the same thicknesses as the solid walls described above, but with glass as the material gives designers the ability to play with opacity and transparency, solid and void without all the details of mullions, window frames or the concern of curtain wall grids. Those are decisions that can be made later in the process. The use of these glass walls also greatly simplifies building the model. All the walls are basic walls. They join to other walls and clean up at corners, allowing for ultimate flexible and speed in modeling the solid and glass portions of a building.

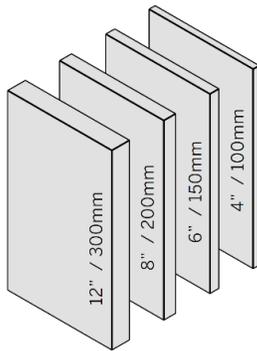


Figure 11: Solid Schematic Walls

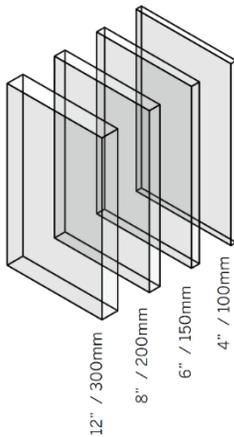


Figure 12: Glass Schematic Walls

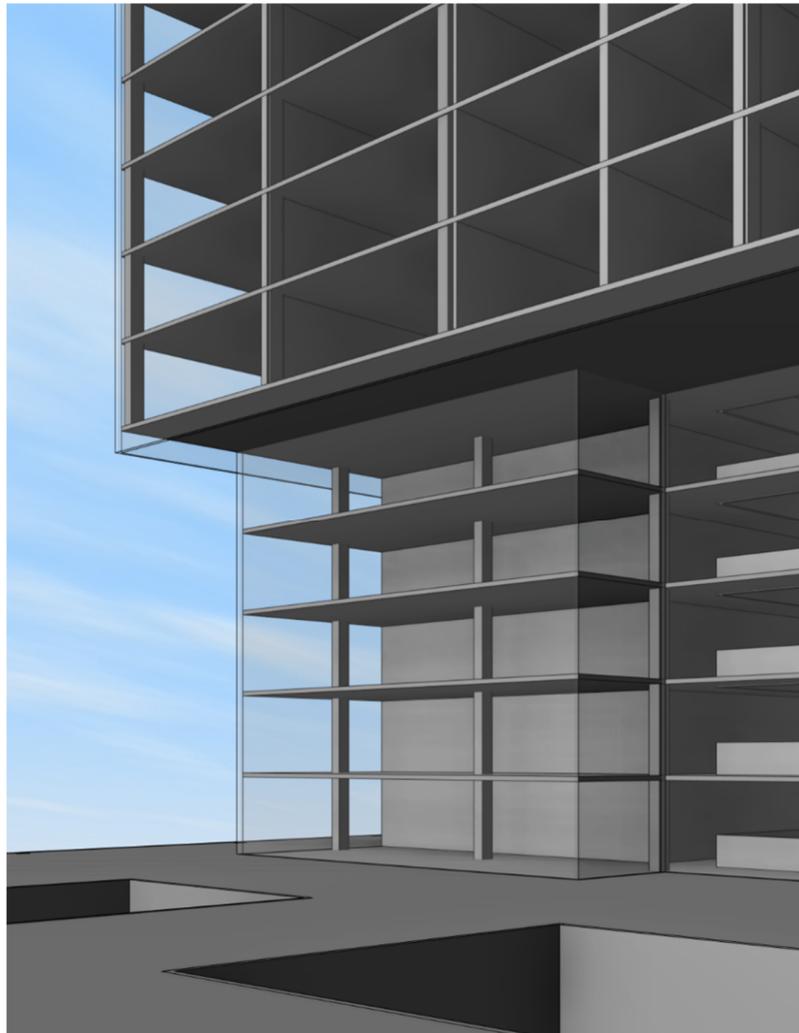


Figure 10: Glass walls

An alternative to using glass walls is to use a “glass-only” curtain wall. This is a curtain wall made up of only glass system panels and a curtain grid. No mullions are applied. The result is very similar to a glass wall with a pattern applied, but the curtain wall provides greater flexibility in terms of the grid size and position.

When actual mullions need to be taken into consideration the schematic glass walls can be swapped out for curtain walls with all the appropriate mullions. The glass-only curtain walls can either be upgraded to include mullions or be swapped out for a different curtain wall with full panels or mullions and the required curtain grid spacing.

Of course, when working with solid and glass walls there is one potential detail that designers will need to manage. How do glass walls join, or not, to the solid walls and how do the two different wall types meet in certain conditions?

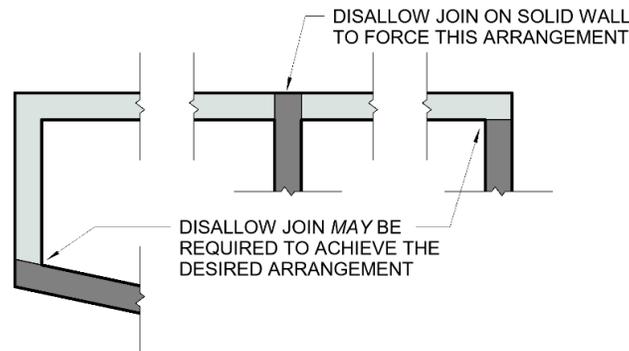


Figure 13: How to control glass and solid walls where they meet

But what if the team is not ready to jump to a full curtain wall? The team wants to stick with the glass walls, but needs to start showing some type of grid? In that scenario, a pattern can be applied to the glass walls. This is done by applying a surface pattern (model pattern) to the material that makes up the wall. When establishing the pattern, how the glass walls are modeled should be considered.

For example, if the glass walls extend multiple floors, that may require one type of pattern. If the glass walls extend only one level in height and are grouped, a different type of pattern may be required. It all depends on the exact spacing of the pattern and how the walls are modeled.

Regardless, creating a wall pattern through the material of a wall is a quick and easy way to display a grid in elevation, while maintaining a clean, mullion free floor plan.

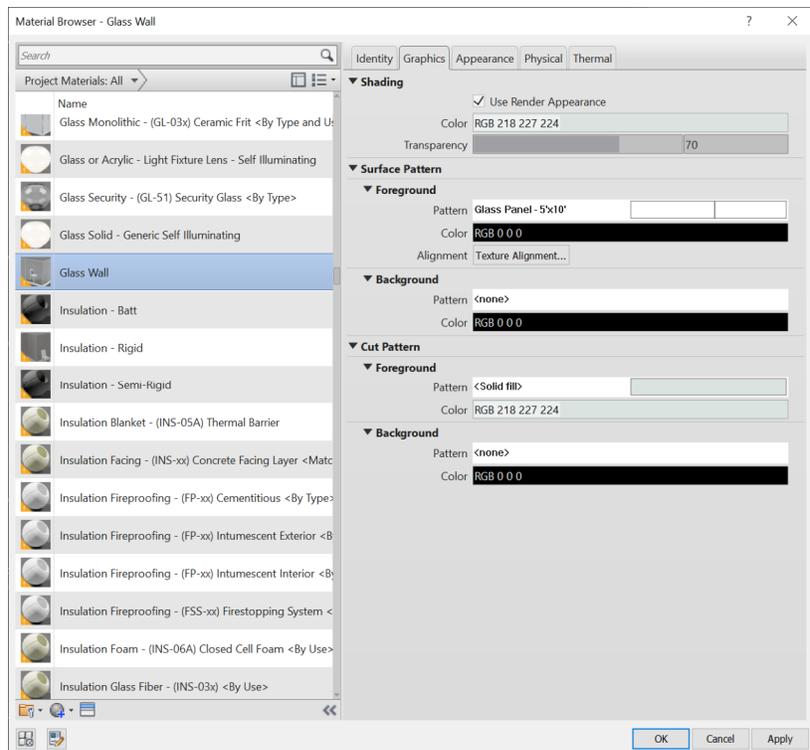


Figure 14: Applying a pattern to a glass wall

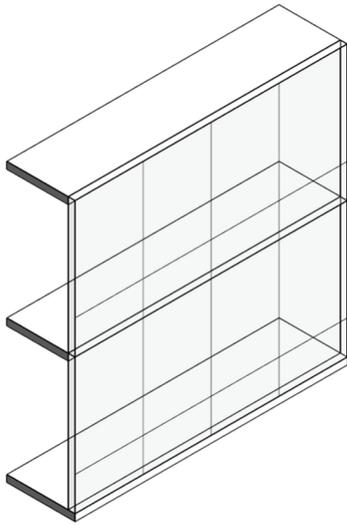


Figure 15: Schematic Glass Walls with Patterns

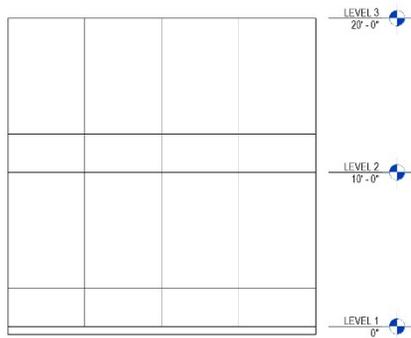


Figure 17: Schematic Glass Wall in Elevation



Figure 16: Patterns on glass walls

BONUS: Location Lines

(not included in live presentation)

Users should always pay attention to the location line of the walls they are drawing. However, almost everyone occasionally slips and forgets to change the location when needed or use the wrong location line. In the long run this often does not matter as long as the design does not change, but in early design phases the design is constantly changing, making a wall's location line very important.

Location lines are critical when swapping out a schematic wall or curtain wall for a more specific wall using actual materials with real thicknesses. For example, if a schematic wall has been built with the interior side facing the exterior of the building, when it is swapped for a stone veneer wall, the gypsum board will be on the outside of the building and the stone on the inside. Flipping the wall is an obvious solution, but this can often shift the edge of the floor that lies beneath the wall, or interfere with other joined walls.

In early design phases, the model often needs to be built quickly, but don't build the model so quick as to neglect using proper location lines. If the appropriate location lines are used, exchanging schematic walls and glass walls for more accurate finish walls will go smoothly with minimal conflicts.

BONUS: Railings

(not included in live presentation)

Railings are probably the most complex (and frustrating) elements in a Revit model. It is often very difficult to control the placement and spacing of elements such as balusters, or the size of glass panels. This results in railings that seldom look the way the designer wants them to look. Therefore, in early design phases avoid using railings if possible. Instead use walls or curtain walls. In fact, the use of walls or curtain walls in place of railings is one practice that can often be used throughout the life of a project, even beyond the early design phases.

There are many different methods for building railings without use of the railing tool. Glass railings can be made from a Glass-Only curtain wall as described in the section on walls. These same walls can be accented with a railing that consists of only the actual handrail, no balusters or panels. In other cases, mullions can be applied to glass curtain walls to stand in for balusters. Lastly, a curtain panel family can be created to act as a baluster. When applied to a curtain wall it can take on the appearance of a railing. This same concept can be used to create fencing.

All of these techniques work well for railings at balconies or anywhere else a flat railing is required. When a railing is attached to a stair, all of the options are still available except for the use of curtain panels. When a curtain wall is used on a stair the profile of the wall is edited to match the slope of the stair. Because the shape of the individual curtain panels that make up the wall are no longer rectangles, only system panels will work, but if an empty system panel is applied along with mullions, the result is still a picket railing. An actual railing can be used for the handrail.

Examples of all these conditions are pictured in figure 18.

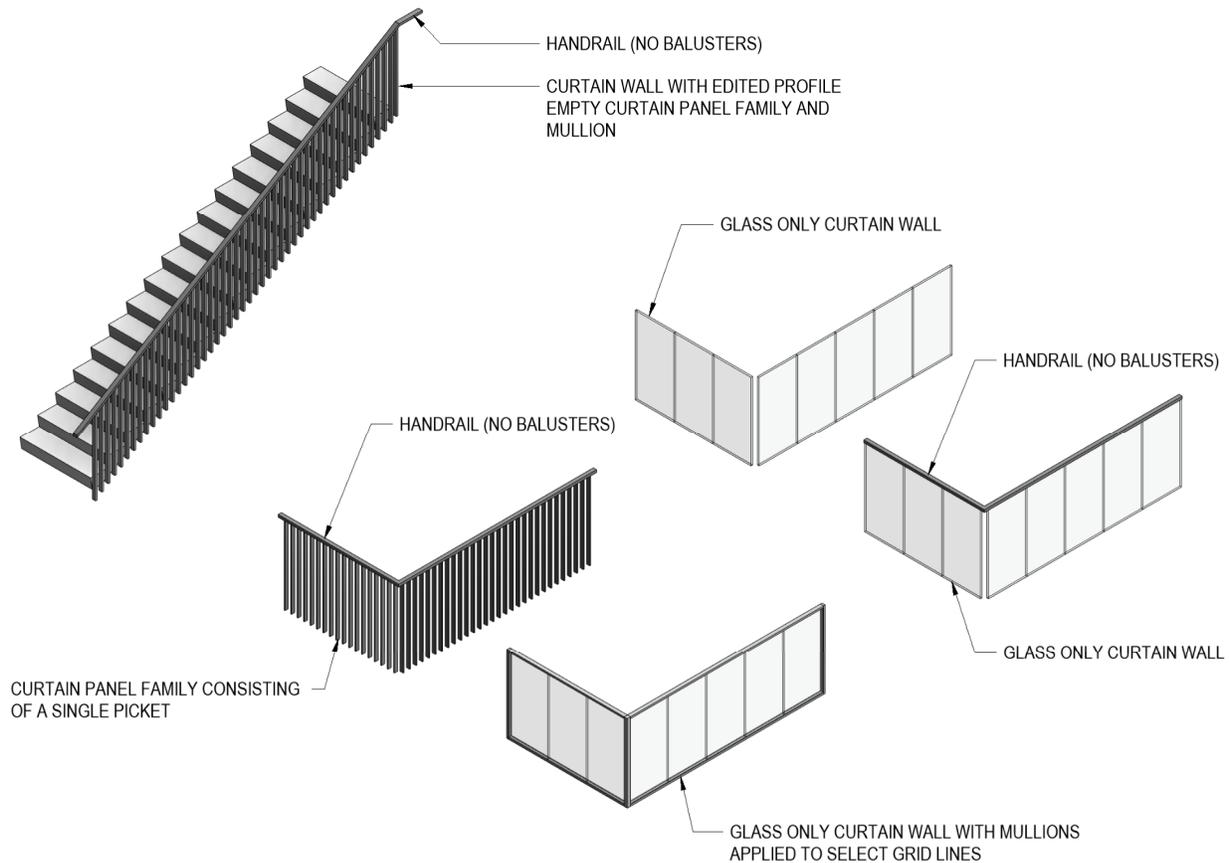


Figure 18: Curtain walls as railings

Building Sections

The concept of using schematic walls can also carry over to floors and ceilings. In early phase design, the exact details of the floor/ceiling assembly may not be important but the floor to floor and floor to ceiling dimensions probably are critical. Therefore, rather than modeling the floor as a concrete slab and separate ceilings, create a schematic floor that represents the full floor/ceiling assembly.

Using this technique has the bonus of producing building sections that are much more diagrammatic and more appropriate for early phase design, while also expressing the design intent much more clearly. However, the true advantage to this approach is the amount of time it will take to build the model. The team only has to deal with floors and walls. There are no ceilings to create and no walls to connect two or more ceilings of different heights. When a floor is edited the ceiling is also edited.

Overall, there may be more floor sketches, if there are many different ceiling heights, they will be much easier to model and manage, as opposed to having separate ceilings and floors.

If the architectural team is lucky enough to have an MEP team working in Revit at this early stage, the MEP team can set all the floors to be slightly transparent in their model and then they will be able to see all the ducts, cable trays and other equipment running through the floor/ceiling sandwich, yet still be able to see the outline of the floor/ceiling sandwich to make sure everything is contained within it.

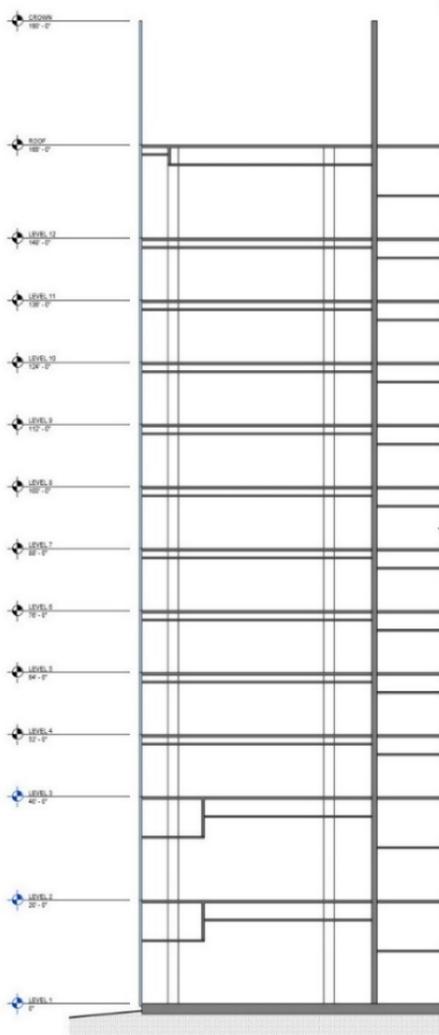


Figure 20: Traditional Section



Figure 19: Schematic Section

The image below shows how multiple floors can be used to create a simple diagrammatic building section and a simpler model in general. When the floors are joined, the lines between the floors will disappear if all the floors are made up of the same material.

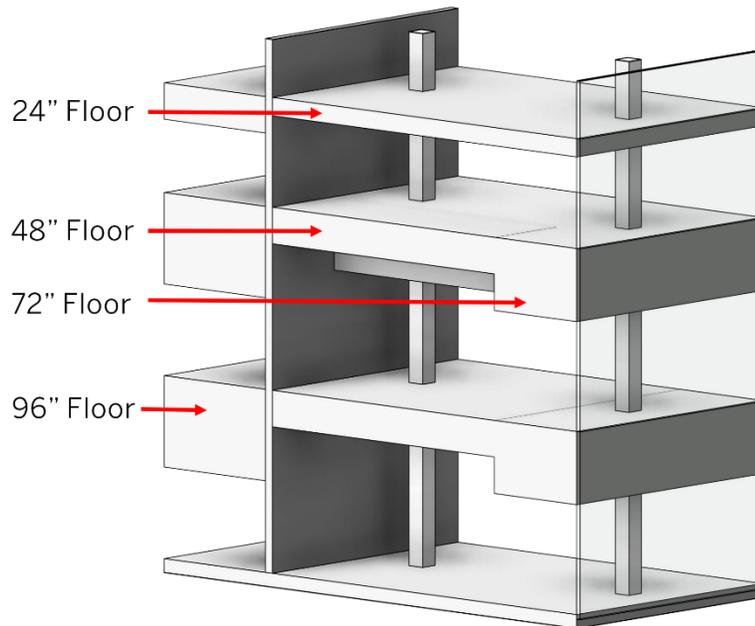


Figure 21: Schematic Floors

Adding Details to a Floor/Ceiling Assembly

Adding detail to a floor/ceiling assembly usually takes the form of adding light coves, trim or other decorative accents that tend to be line-based. To create these items, use a line-based family, an in-place floor sweep/reveal, or a slab edge. There are potentially many disadvantages to using a line-based family, chief among them is that if the floors change, the family will not update with the floors. Therefore, the two best options are in-place floor sweep/reveal or a slab edge.

In both cases the edges of the existing ceiling (floor) can be used to generate a sweep/reveal or slab edge. If the shape of the original floor/ceiling changes, the sweeps will update, if the changes are just a matter of size or position and are not adding or subtracting edges.

The advantage of using in-place sweeps is that they are completely flexible and each one can be unique. If the same shape needs to be swept in many different places, a profile family can be created and used to generate the shape. If the shape changes, update the profile and the sweeps will update. The largest disadvantage to in-place families is that team members are not always focused on keeping them organized and consistent. The result of this are fragments of in-place families on multiple levels or overlapping with other in-place families. If this technique is used for its flexibility, the project BIM Manager should establish the rules for how the families should be created and used.

The advantage to using slab edges is that they are generated from profiles and therefore, the best option for sweeps of the same shape used in multiple locations throughout a project. The tool is more rigid than in-place families and therefore tends to be easier to control and manage. The only real problem some people have with using slab edges for this purpose is that it is a tool being used in a way that it was not intended, and this can cause some users anxiety.

Not all floor/ceiling details are line-based. For example, a sloped ceiling can be added as an in-place floor extrusion. The concept is the same, except instead of using a sweep and tracing a path, an extrusion is created, usually from a vertical plane. The only thing that can complicate the use of extrusions is if the sloped ceiling, for example, is not a perfect rectangle in plan. If it is an L-shape, or any other non-rectangular shape, a void must be created to cut the extrusion into the required shape.

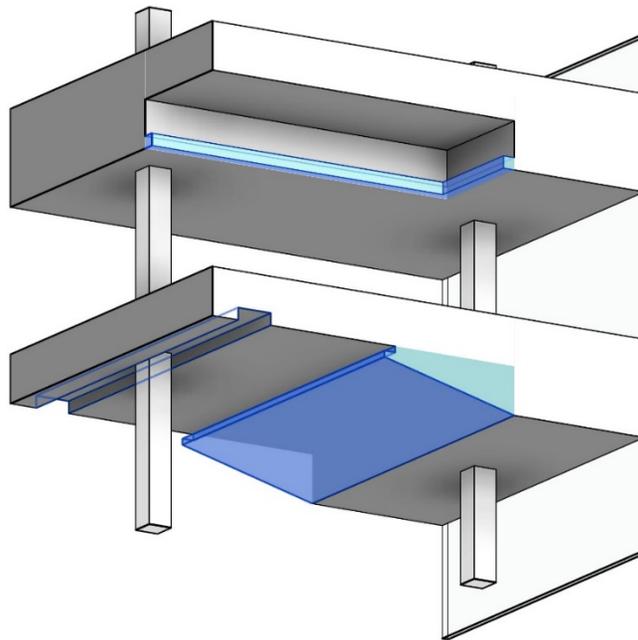


Figure 22: An example of an in-place floor sweep and floor extrusion.

Schematic Families

It is inevitable that if a highly-detailed family is placed into a rather schematic model, the client's eye will be drawn to the details and the questions will start. How many cabinets can fit? Are those side or rear counterweights on the elevators? What size are those mullions? What brand of toilet is that? Of course, none of those questions can be answered because the design has not yet progressed to that point. So to avoid those questions and a complete derailment of the conversation in a client meeting, avoid using the typical families used in design development or construction documents. Instead use generic or schematic families. These families only convey that something exists in the space or in a specific location, but nothing more.

In keeping with the idea of modeling only what is known and at an appropriate level of detail, it is a good idea to have a collection of these schematic families loaded into office project templates, so that they are at the fingertips of team members and can be used on any project in the early design phases.

These families contain very little, if any, detail and are often diagrammatic in nature. Some may only be 2D. Nevertheless, they all are a placeholder for something to be developed later and replaced with more detailed families. Some of the most common examples of this are casework, risers, elevators, plumbing fixtures and curtain panels, but there are many more.

Casework

In early phase design, the location of casework may be known, but the exact design is not. The use of schematic casework families that contain instance parameters so they can be stretched to fit, provide the appropriate level of detail and are very quick and easy to add to a model.

However, casework comes in many different configurations, and if there are families for all the most common configurations - base cabinets, upper cabinets, full height cabinets - they can be loaded into an organization's project templates and deployed on any project. Additionally, more specific and less frequently used configurations such as L-shaped casework, islands or u-shaped casework can also be included. However, some users may not like having a lot of casework families in their file especially if only a few are frequently used.

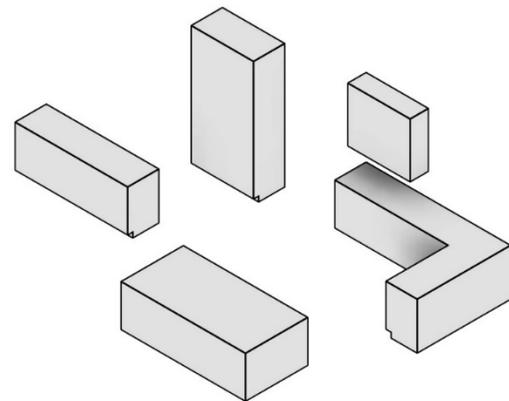


Figure 23: Casework

As an alternative to loading up a project template with every conceivable casework configuration, consider creating a special Schematic family library on a local drive, or use a cloud service such as Unifi and create a saved search for schematic families. In either case, all schematic families, including casework, are located in one place and are easily accessible for team members.

Risers

At the beginning of the design process there are many things that the design team must incorporate, but may not know exactly where or how large the items need to be. This becomes even more complicated when items must be tracked through multiple levels of the building. That is why it is a good idea to model placeholders for the mechanical, plumbing and electrical risers yet to be designed.

Not only does including these placeholder families remind the designer that riser space needs to be included, but they can help make sure that the space is aligned and clear on all levels. Of course, shafts can also be used for this purpose, but a major drawback to using shafts for risers is that shafts cannot be tagged. Many times, in early design phases risers need to be identified and the easiest way to do that is with a tag.

The riser families work like a column family. A base and a top constraint is identified and the families stretch vertically to fit. The width and length are instance parameters so the families can stretch to fit. Each riser type has a different color making them easy to identify in shaded or consistent color floor plans or sections, but remain black and white in hidden line views.

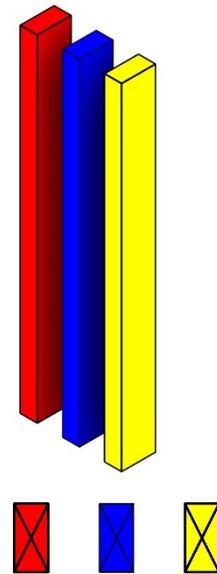


Figure 24: Risers

Elevators

The process of designing a building begins abstractly. The building slowly evolves and gains detail as the design process progresses. If highly detailed elements are used at the early stages it can be distracting. With elevators, questions about counterweights, cab measurements and door operations can start to be asked too early and divert from more important basic concerns, such as which floors an elevator services.

This is why it is often a good idea to begin with a greatly simplified version of an elevator. One that merely represents the space of the elevator. In addition, on high rise projects that contain many elevators, there are usually several groups of elevators that serve different parts of the project. By using color-coded elevator families, it is very easy to see if an elevator services a residential floor, office floor, hotel floor, etc. The colors can make it easier to make design decisions, or simply to visualize the levels that any given elevator services.

The same top and base parameters exist in the schematic elevator families as in the more detailed elevator families, so swapping out the schematic elevators for the more detailed elevators can be done without losing information. However, the detailed elevators have many additional parameters that must be addressed.

An integrated color scheme

If colors are used on elements such as elevators, those colors should be part of an overall integrated color scheme that extends to other families, the area, room and spaces color schemes in a file and the filled regions.

The idea is when an elevator is assigned to “Office” it is blue as in figure 25.

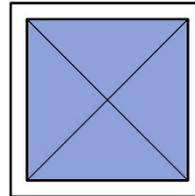


Figure 25: Schematic elevator in plan

This same shade of blue should be used in the project’s color scheme to represent “Office” in all color floor plans. All of this can be included in a project template, and more information on this topic can be found in the “Don’t Reinvent the Wheel: Project Templates” section later in this handout.

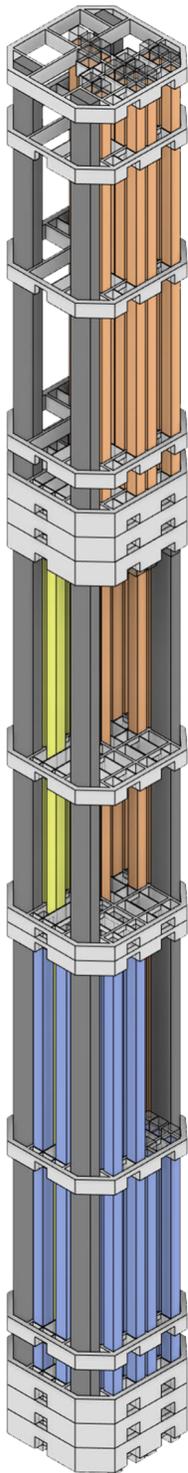


Figure 26: A schematic core using color scheme elevators. Only selected levels are fully modeled.

Accessibility Clearance (ADA)

It is very important for designers to keep minimum space requirements in mind when initially developing a floor plan. In many cases accessibility requirements (ADA requirements in the United States) can drive how much space is required for a room especially in terms of bathrooms and egress. Therefore, designers should have a way to verify that the appropriate amount of space has been provided.

This can be done with a series of schematic families that in plan display as dashed lines representing the minimal clear space that must be provided. These families can be dropped into plans and positioned to verify that the building code has been met.

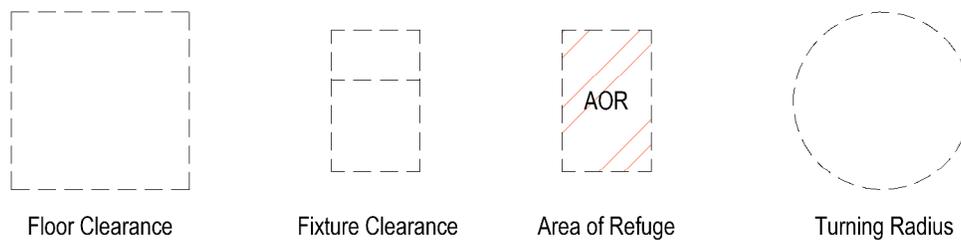


Figure 27: Accessibility Clearance Families

In plan these families seem extremely simple, and some may question the need for them to be families. Detail lines could just as easily be drawn into a plan. However, these families are secretly 3D. The 3D component has been set to only display when a view is set to a “Fine” level of detail. Therefore, most of the time it is invisible, and does not cause problems with rendering programs such as Enscape.

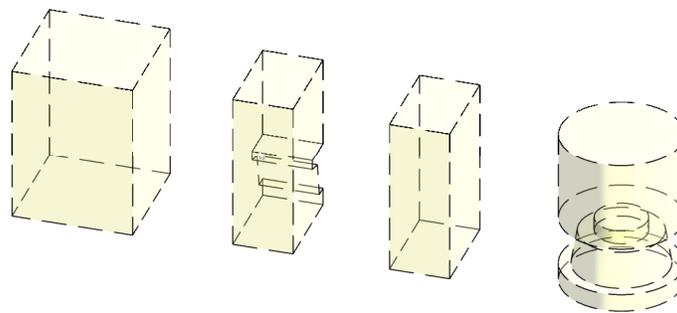


Figure 28: Accessibility Clearance Families in 3D

In later design phases these families can be clashed with other items such as walls, doors and casework to not only determine if the correct amount of space has been accounted for in plan, but also in three dimensions.

In-Place Families

In-place families can be controversial, some people avoid them if possible, others may have rules against them. If used correctly and in an organized methodical way, In-Place families can be a very valuable and useful tool especially during early design phases. Of course, in-Place families can be used in any number of different situations, but when distilled down, there are three basic uses of In-Place Families:

- For design and visualization aides
- For items that are difficult to model using normal system families
- For custom “fit to space” elements

Design and Visualization Aides

In-Place families used to aide in the design process or as a visualization tool are often very large, many times spanning multiple floors and stretching long distances through a project. They also tend to be very simple and abstract. This is due to them being utilized in diagrams. One of the most common examples of this is means of egress.

Buildings often have complex egress paths. There may be one or more levels in which one stairway transfers over to another stairway in a different location. This can make it difficult to understand the full path, understand how the stairways connect or exactly where the stair shafts exist on different levels of the building. To illustrate this, and verify that the designers have provided the required space to accommodate the egress path, an in-place family can be created that abstractly represents the extents of the path. This consists of extrusions representing the space of all full stair shafts and extrusions representing the horizontal paths of travel connecting the stairs as well as the connecting the point of discharge.

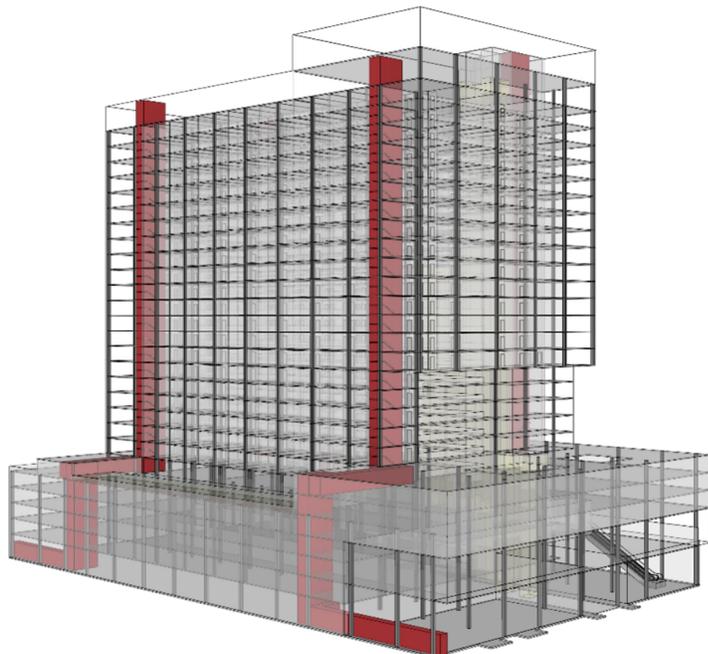


Figure 29: Means of egress diagrammed

In figure 29, the means of egress is represented by the red in-place families. In this image, the walls and floors have been overridden to be slightly transparent, allowing the generic model in-place family to be clearly seen. The red color is used simply to make the egress path stand out from the rest of the building model.

This is an example of how a concept or schematic design level model can be used as part of a diagram. However, while this is great for presentations, it can be even more useful as a design aide. Once the egress path is built, it is a consistent reminder of where the stairs and horizontal paths need to be and allows the designers to design the

rest of the building while always providing the appropriate space for the egress paths.

*BONUS: Difficult to Model Items
(not included in live presentation)*

Occasionally there are spaces or elements that are difficult to model with floors, walls and ceilings alone. At other times, the designer may be testing ideas and needs a quick and easy way to model the space free of the restrictions of floors and walls. In these scenarios in-place families can be the answer.

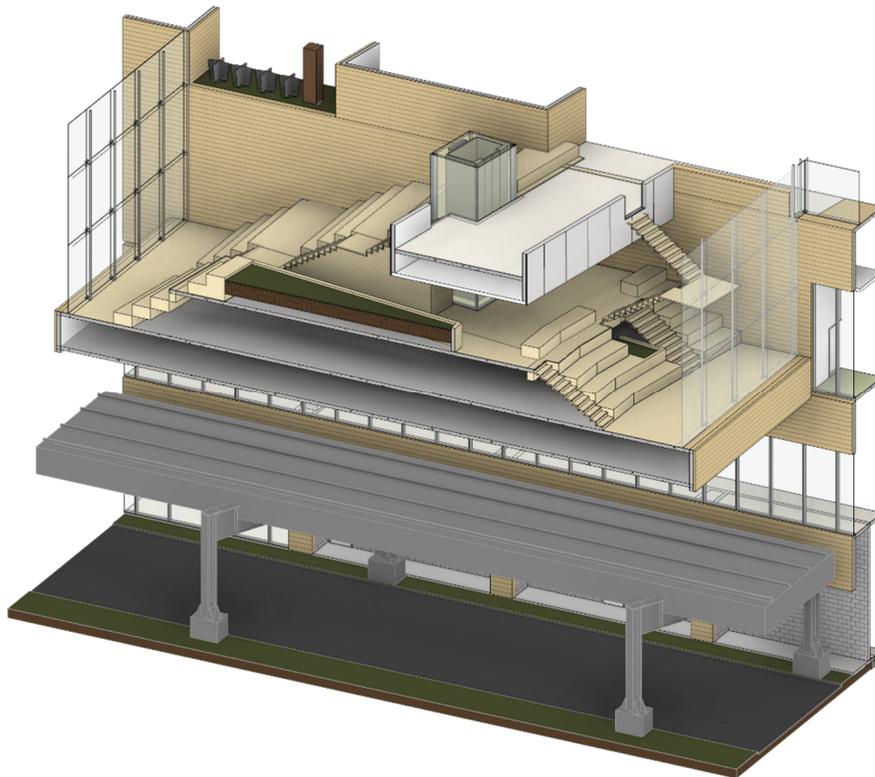


Figure 30: In-place model used to help design a difficult and complex space

In figure 30, the design called for a very complex space that consisted of multiple levels, four different stairways, an elevator, several planters, bleachers and a small grab n' go café right in the middle of it all. To develop and explore different design concepts the designer used several in-place families to model the various elements as blocks. Once the abstract blocks were arranged to the satisfaction of the designer, some of those in-place blocks became more detailed, other elements were replaced with real floors and stairs. In the end, the flexibility and freeform nature of in-place families helped the designer arrive at the desired result faster than if traditional floors, walls and loadable families had been used.

Fit to Space Elements

In most designs, there are at least a few items that are highly custom pieces, some may fit into and fill a space precisely. The reception desk depicted in figure 31 is one such case. It was

designed specifically for that space. It fits exactly to the wall and shape of the adjacent window. Of course, this could have been created as a loadable family from the start, but the designer would need to measure the Revit model and design the desk out of context. By creating it as an in-place family the designer can not only design it in the context of the Revit model, but need not go back and forth between the family and the building model.

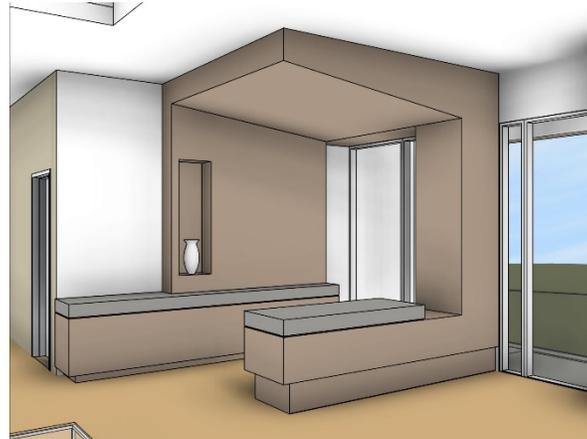


Figure 31: Custom reception desk designed specifically for this space

However, in many cases it may not be a good idea to have in-place families remain in a Revit model long term. For example, the above reception desk may need to be broken down into different pieces for scheduling or quantity take offs or maybe some of the elements of the design should be converted to walls. Whatever the reason, there are many cases where in-place families need to be converted to a loadable family. However, is it possible?

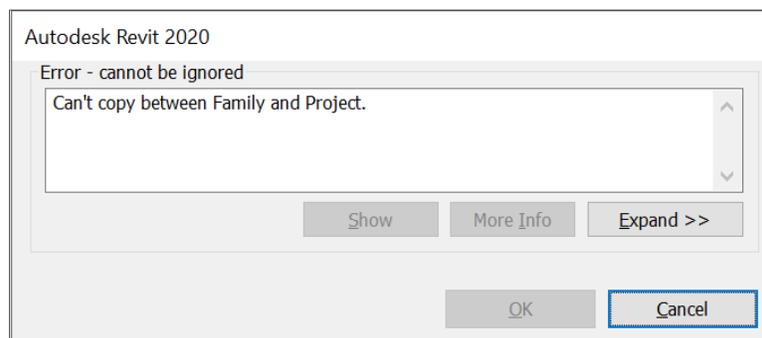


Figure 32: Can't copy between family and project?

The secret is not to copy the in-place family from the project to a family file, but copy it family to family. To do this open the in-place family for editing in the project file. Once in the edit mode for the family, select all the geometry, model lines, reference planes, anything that needs to be copied, but avoid selecting anything 2D and copy those items to the clipboard (Ctrl-C). Then, open a family file using the appropriate category and template (while keeping the in-place family open for editing in the project file.) In the family file, go to the floor plan view and paste in the content from the clipboard (Ctrl-V).

In the process, some of the geometry may lose its association with reference planes, geometry that was locked to planes or other geometry may become unlocked, and lastly sometime voids will stop cutting some solids. So, there may be some clean-up required, but it will be minimal compared to the prospect of needing to remodel the entire in-place family from scratch as a new loadable family.

In fact, using this method of transferring information from an in-place family to a loadable family, designers can use an in place family to simply measure a space in the Revit model into which an element needs to fit, and then use those measurements to start building a loadable family. In this case the designer would open a new in-place family within a project. Then draw model lines (Not detail lines, everything must be 3D) on the floor along the walls or other datums to create an outline for the family to be created. If heights are critical, simple forms can be created to represent the height or slope of the to-be design family. Once all the boundaries and restrictions are laid out in model lines or in forms, these items can all be copied to the clipboard and pasted into a loadable family. Once pasted into a loadable family the in-place items essentially become a template upon which to build the loadable family.

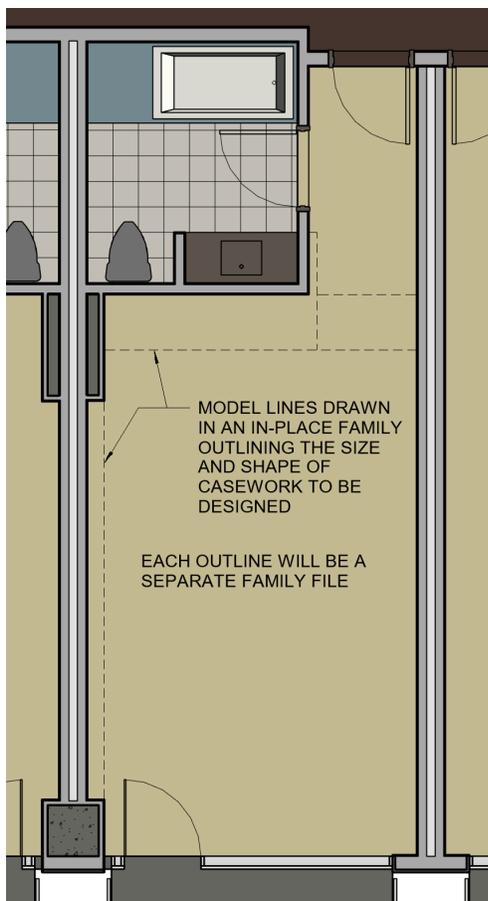


Figure 34: Creating an in-place “template” family to build a loadable family

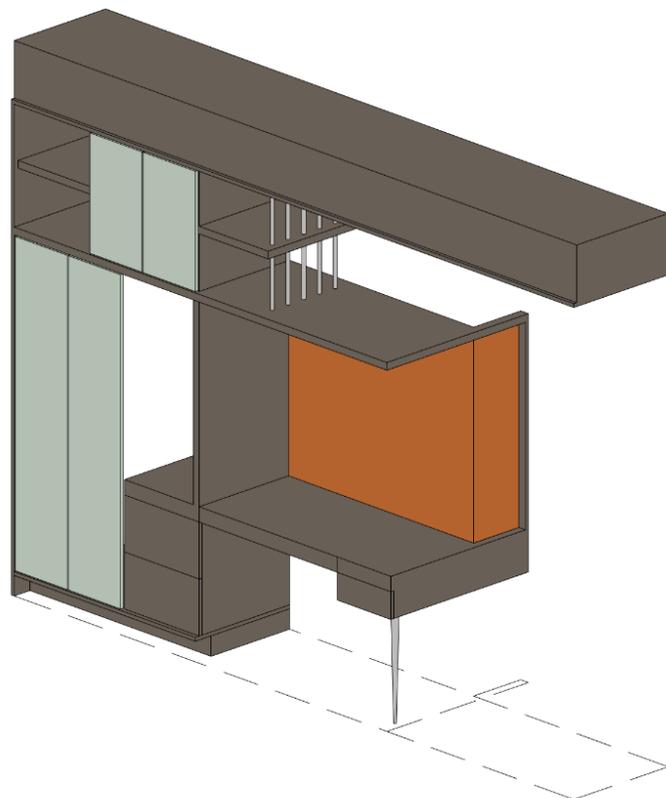


Figure 33: Loadable family from an in-place “template”

Now in the loadable family, the designer can see what the space constraints are in the Revit model and can build a loadable family that will fit perfectly into the Revit model (assuming the Revit model does not change in the meantime.) The in-place family can remain in the project if the space is likely to change, or can be deleted from the project now that the loadable family has the required layout.

In-Place Family Tip:

In the project browser it can be difficult to distinguish between in-place and loadable families. Because in-place families have different properties and capabilities from loadable families it is important to differentiate them in the project browser. This can also make them easier to find when they need to be edited. Therefore, teams should come up with a naming strategy for in-place families. For example, all In-Place families start with “IP” or “In-Place” and then the rest of the name conforms to office standards. In the long term this will help teams better manage all of the families in their files.

Materials

Materials are very important in early design phases, especially with the advent of real-time render programs such as Enscape, TwinMotion and others. There is more pressure than ever to show real materials in projects earlier and earlier in the design.

The problem is that in the beginning stages of a project, the design is changing rapidly, and material choices are potentially up in the air. Teams need a quick and easy way to display materials in their projects without the need to create highly detailed walls containing different combinations of materials. One solution can be to create single material walls.

These are walls that are made up of one single material, such as stone, metal panels, gypsum board etc. The idea is that it is easy to duplicate a wall and change its single material to something else rather than get bogged down in highly accurate multi-material walls that take into account every possible combination of materials. These single materials wall types work best in exterior wall applications and when looking at the model from the exterior in a program like Enscape. If the designer is focused on the exterior, it does not matter if stone is showing on the interior.

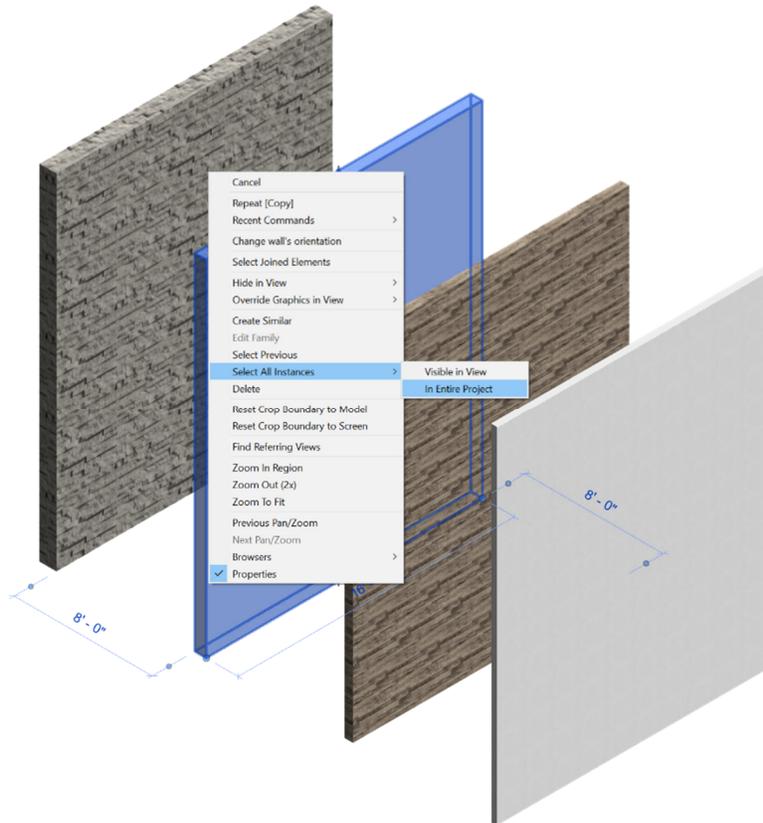


Figure 35: Single material walls

The best part about this workflow is that if all of these single material walls have the same thickness they can be swapped out very easily with little to no clean up. Don't like the stone? Then right click on one of the stone walls and "Select all instances"... "in the entire project" and swap all the walls out for metal panel walls. Obviously, once the final materials have been settled, these single material walls should be exchanged for more accurate walls with all of the proper materials.

Painting Materials

Painting material can be controversial, and many users do not like to use the tool, but in early design phases it can come in handy in rapidly changing designs. It can also be an alternative to using single material walls. Painting materials is an easy and fast way to add materials to a project. However, it can lead to problems down the road because when a material is painted, it is a zero thickness. If a material has a significant thickness such as stone veneer, it may cause unforeseen problems when the paint is replaced with the actual materials and their true thicknesses.

Therefore, painting materials is only recommended for testing materials and what they will look like with other materials. Rather than spending the time to create walls, floors and ceilings with all the specific materials required, paint the materials on the surfaces. Then view the model in

Enscape of another real-time renderer. The materials can be changed on the fly by painting the surfaces and the results can be instantly viewed in Enscape.

Painted materials should not remain in the model long term. In the later design phases, walls, floors and ceilings that include the actual materials and their correct thicknesses should be introduced.

One of the biggest complaints about paint is that it is not always obvious which surfaces are painted and which ones are not. To solve the problem, a Dynamo script can be developed to remove all paint from a Revit model. It requires a little bit of python code, but it can be done. Then after all the paint is removed, the team can start painting again, or allow the team to move on to the next phase with a clean, paint free model.

Mobilize Quickly: Setting Up New Files

In early design phases, team members need to create new project files frequently for a variety of reasons. Whether files are being created for a design exploration or to develop an alternate scheme, when many different people are creating files, the files can often become disorganized and inconsistent.

To remedy this problem a process must be put into place that addresses three key concerns. First, it must be a simple and easy process that any team member can perform. This takes the burden off the project's BIM Manager to create every file. Second, the process must be fast. Teams cannot afford for one team member to spend all day creating new files. Third, and probably most importantly, the process must be able to create consistent files. All the files should have the same worksets, matching levels, the same content, materials and settings loaded.

Dynamo to the Rescue

With these goals in mind, Dynamo was immediately chosen as the tool for this task, along with a heavy reliance on Excel as a place to compile the required project information. With this in mind a workflow was beginning to take shape, but some goals had to be defined.

Workflow goals:

- No manual editing of project files required
- Re-use existing project information
- Compatible with Dynamo Player
- Easy
- Fast

With these goals in mind, a workflow consisting of three easy steps was developed.

Step 1: Excel

The first step is to gather basic information about the project. It is assumed that most, if not all, of this information was generated either in the pre-design phase or can be gleaned from the project's program requirements. This step is also performed just once, just before the first file is created. After that, it only needs to be revisited if or when a project's information changes. However, the associated set-up scripts should only be used to set-up new project files. If, for example, the levels change on a project, the Excel file can be updated, but to update any existing Revit models, a different script would need to be used to update the existing levels. New files created after the level change could be created using this Excel file and the set-up scripts.

A copy of the Excel file to be used (the template) is saved on a protected network drive, so the original cannot be altered, but can be copied. The team creates a copy of the template in their project directory and begins to add information. In the Excel template, there are 6 tabs. The file opens to the first tab, where the required Dynamo packages are listed, as well as basic instructions. The rest of the tabs require the team to fill-in the required project information as follows:

Workset Tab

At SOM there are six architectural file types that are used. Each file type is listed with all of the worksets that the file should have. The team needs to complete this list by adding the workset names for any linked files that will be added to the files.

Project Information Tab

Here the team must enter all of the basic project information. This is most of the information listed under “Project Information” inside Revit.

Levels & Working Floor Plans Tab

On this tab, the team lists all of the levels in the project along with their elevation above “0”. At SOM, the levels and their elevations are typically determined as part of the pre-design process and another Excel file known as an “Area Chart” contains the level information. All the team needs to do is open the area chart and copy/paste the level information into this tab. If there is no area chart the level information, or what is known regarding levels, must be entered manually. The information on this tab is not only used to create the levels for the project, but also generate a “working” floor plan for each level.

Layout Views Tab

At SOM we have two major view types – working views and layout views. Working views are used for modeling and are not placed on sheets. There are typically working views for every level. Layout views are views that are placed on sheets. (They are called Layout Views to avoid the confusion between the terms “sheet views” and “sheets.”) For high rise projects not every level is documented. Therefore, not every level has a layout view. On this tab, the layout views are listed by what the view title should be on the sheet and the view’s associated level. As with the levels, a rough outline of the documentation set is created as part of the pre-design phase, and another Excel file called “Sheet List” has typically been created. If this file exists, the list of sheets can help in creating the list of required layout views.

Sheets Tab

On this tab, the team lists all of the sheets to be created. As stated previously, there is often a pre-existing “Sheet List” in which the sheet information can be copied from that Excel file and pasted into this tab.

The following is a view of the Excel sheets described above:

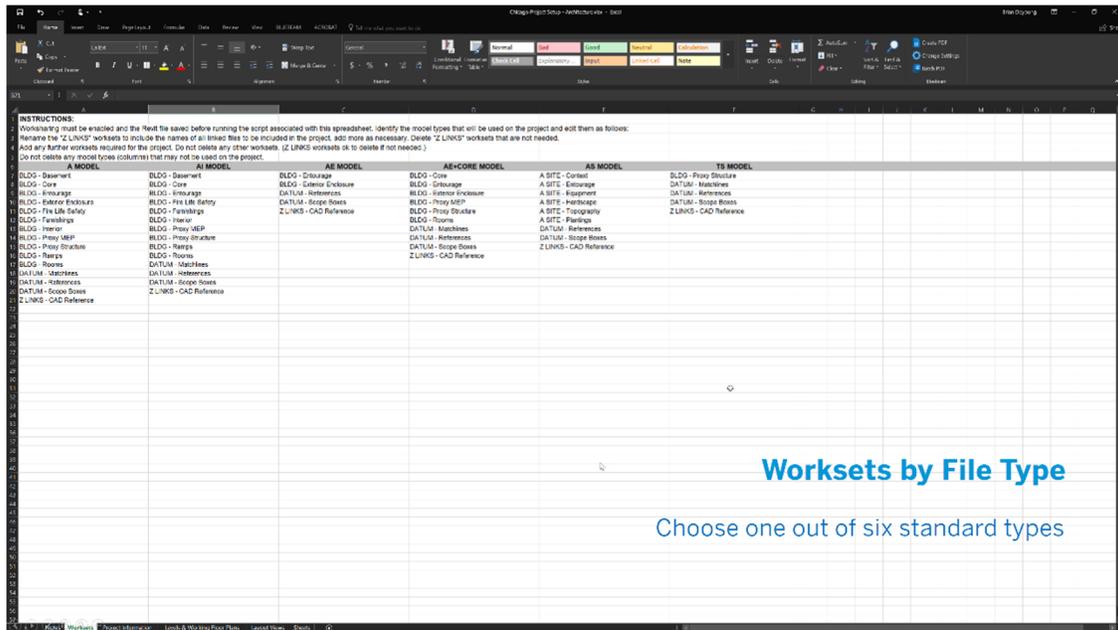


Figure 36: Worksets by type

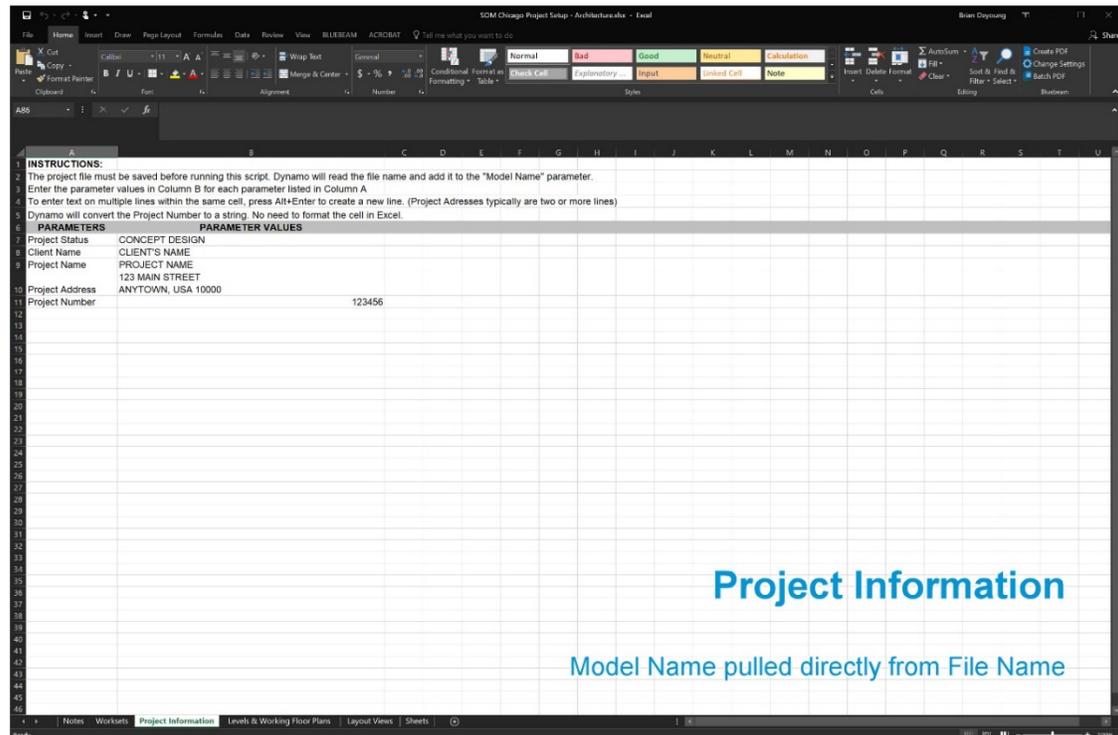


Figure 37: Project Information

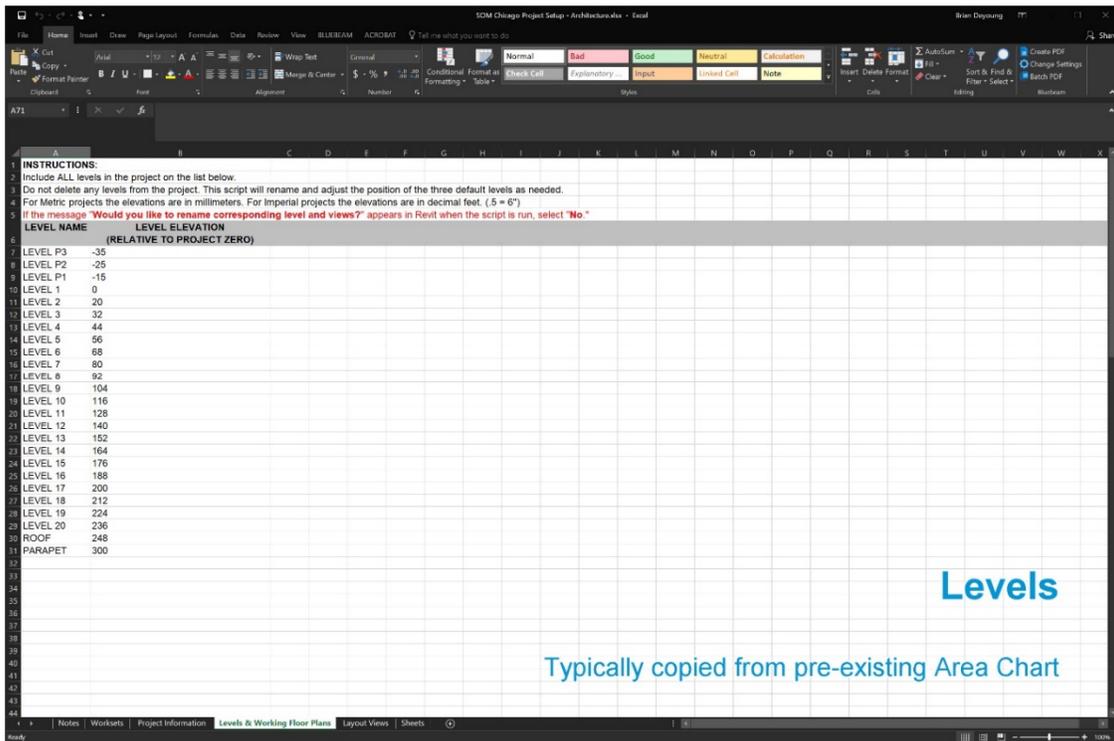


Figure 38: Levels

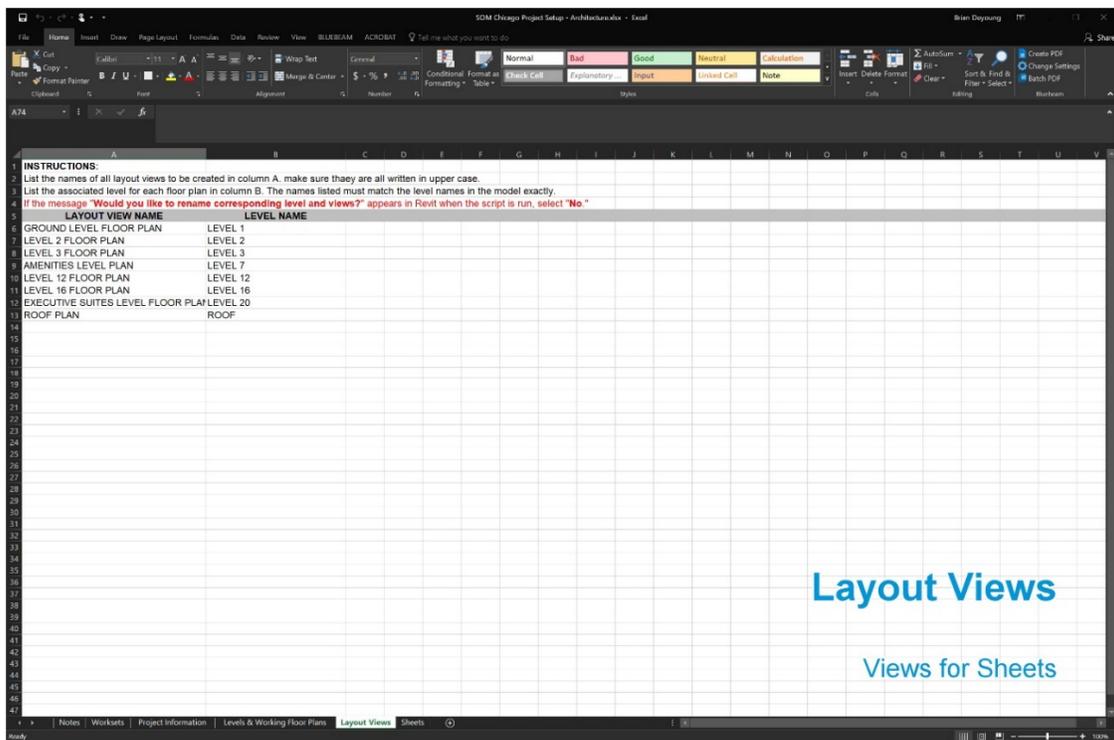


Figure 39: Layout Views

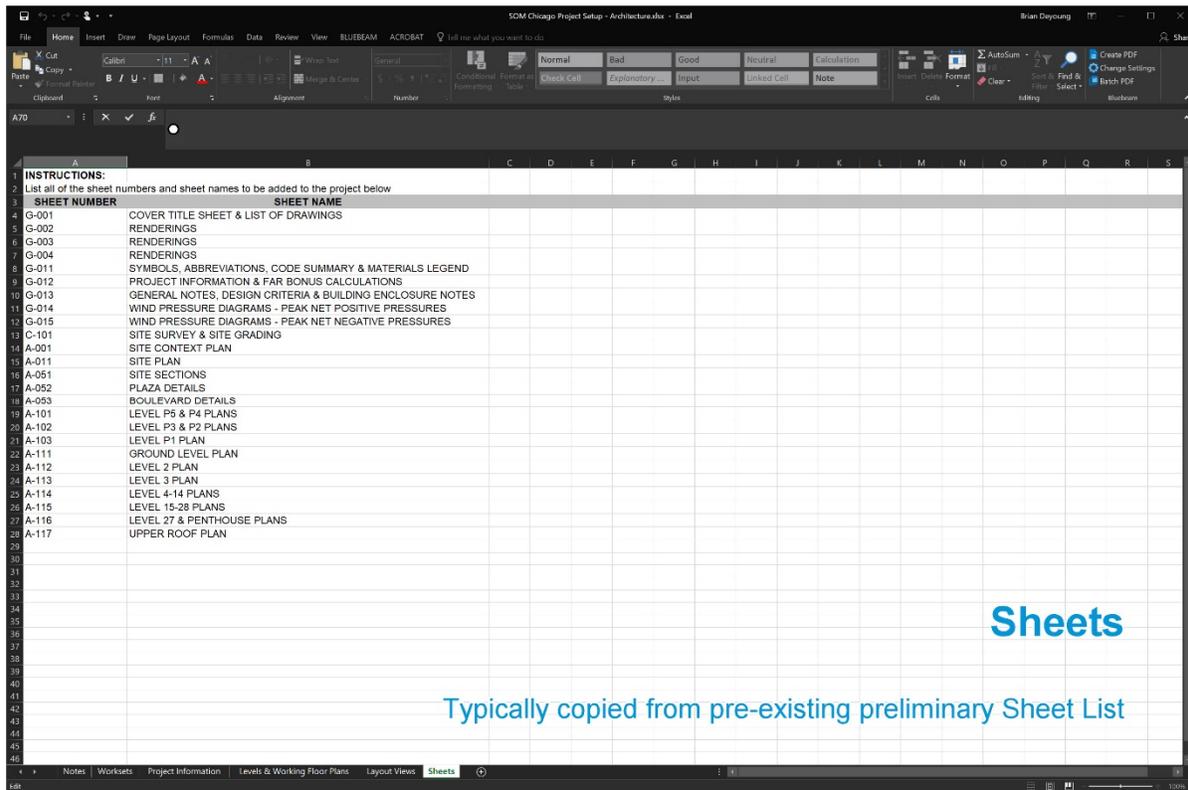


Figure 40: Sheets

Step 2: Start a New Project File & Save

After the Excel file has been updated with all of the appropriate project information, the team may move on to step 2, which is to open a new project file in Revit and save the file with the appropriate file name according to the firm's file naming standards.

In the development of this process, there was some experimentation with using Dynamo to generate and name new files. There was some success in this area, but it was a rather complex procedure and prone to failure. As a result, it was determined that for the sake of simplicity the files would be created and saved manually. This is the only manual step teams must perform after the initial Excel set-up.

Step 3: Run Scripts in Sequence Using Dynamo Player

The third and final step is where nearly all the work is performed. It consists of eight Dynamo scripts (and growing) that are saved in a special folder on a protected network drive. The team must copy the folder containing these files to their project directory and run the scripts from the project directory. The scripts are run from the project directory because there are several inputs that must be set within the scripts. As a result, the Dynamo scripts are edited as part of the process. To avoid conflicts between projects, each project runs the scripts from their own copy.

The Scripts

There are eight default scripts that make up the project file set-up routine. Some use the information from the Excel file edited in Step 1, others do not. If a project has unique requirements, or additional items to set-up individual teams are free to create additional scripts to help set-up files. Likewise, not all of the default eight scripts are run on every file. In fact, the eighth and final script is only run once per project and does not involve any Revit files.

The following is a brief description of what each of the scripts does:

1. Enable Worksharing

This enabled worksharing on the Revit file that is currently open. It consists of a single custom node from the “Synthetic” package.

2. Create Worksets

The inputs for this script are the file path of the Excel file and the model type that is being created. With this information, the script creates all of the desired worksets and also renames the two default worksets to match SOM standards.

3. Create 3D Workset Views

There are no inputs for this script. It takes the list of worksets and creates a 3D view for each workset. In each view, all the worksets are hidden except for the workset the view is named after. These views are used as an aide in keeping elements on the correct workset.

4. Edit Project Information

This script reads the project information from the Excel file and fills in the project information in the Revit file. It also reads the file name, and Revit version and fills in the that information on the project’s “Start Sheet” along with all the other project information.

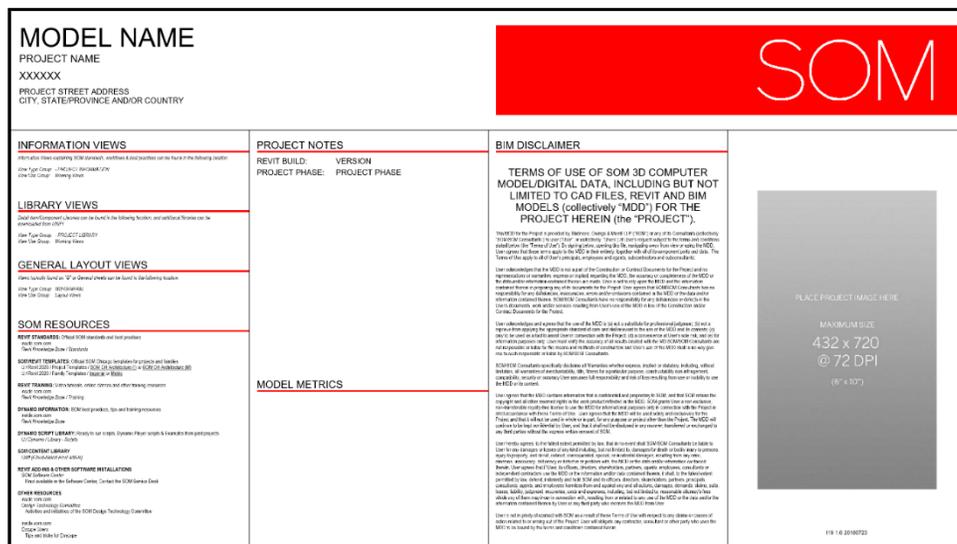


Figure 41: SOM’s start sheet for Revit 2020 (We change the appearance of the sheet with every release of Revit)

5. Create Levels & Working Floor Plan Views

This script does several things. It adds all the levels included in the Excel file at their stated heights, it also integrates the default levels that are in the office template file into that list. As part of that process the site views that are in the template file have the potential to be voided, so the script deletes the original site views and re-creates them from the view located at “0”. Lastly, it sorts all the views in the project browser.

6. Create Layout Views

This script reads the layout views listed in the Excel file, creates a Revit floor plan view for each one and sorts them in the project browser.

7. Create Sheets

This script is very straightforward. It takes the list of sheet numbers and names from the Excel file and creates sheets. It does not place any views on sheets. Of course, this script is only run on the file that contains sheets.

8. Create Keynote & Shared Parameter File

The script performs one of the tasks that is often forgotten when setting up a project. That is to copy the office default keynote and shared parameter files from the network drive to the project directory. This script is run only once per project.

There are currently two additional scripts that SOM uses as part of the project set-up process. These scripts rely heavily on custom python nodes/scripts and thus due to the propriety information contained within them are not able to be shown or shared as part of this class. However, one of the scripts establishes the shared coordinate system for a project, or adopts a coordinate system from another file. It also provides an opportunity to link in other Revit files by shared coordinates and place the links on their own dedicated workset that is named after each linked file. The other script provides the ability to link in 3D files from other software platforms such as Rhino and place them inside an in-place family to make these files easier to manage and control in Revit.

More scripts are being considered or in development to make the project set-up procedure even more comprehensive, including a script that loads additional content and settings, including shared parameters based on criteria selected by the user. Ultimately, some day in the future all of these scripts and procedures may be wrapped up into an custom Revit Add-In that will perhaps make setting up new files even easier.

Dynamo Player

These scripts may be run by opening them in Dynamo, setting any inputs and pressing “Run,” but they have all been designed to be run from Dynamo Player. All the user needs to do is set Dynamo Player to the folder that contains the scripts and they will be listed in Player in the sequence that they should be run, as shown in figure 42.

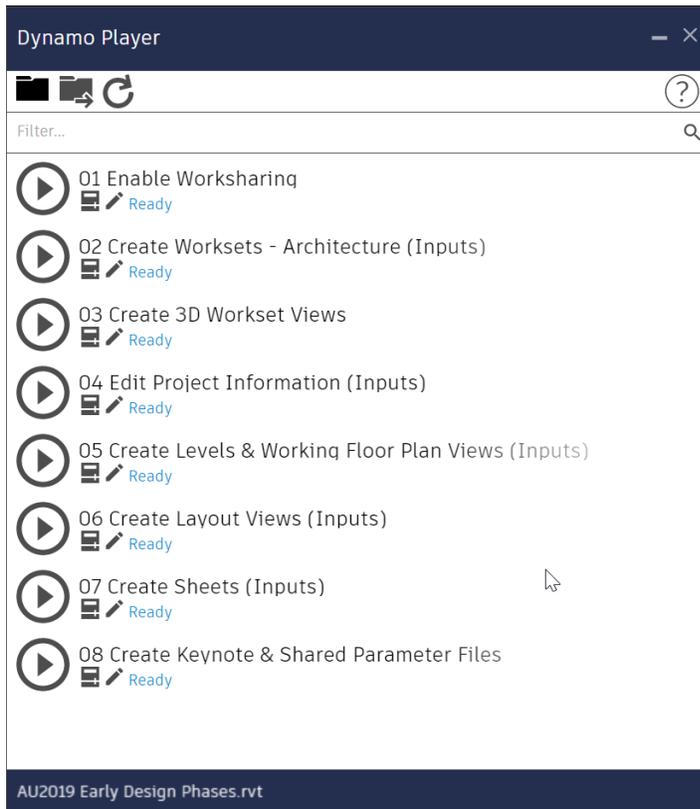


Figure 42: Dynamo Player

However, one major deficiency of Dynamo Player is that if a user presses the play button for a script that contains editable inputs, rather than prompting the user to verify the input settings, the script runs with whatever the saved settings are, and there is no guarantee that the inputs will be correct.

Therefore, in an effort to prompt users to set inputs before running the scripts, any of the scripts that contain inputs, have the word “(inputs)” at the end of the file name. When a user encounters one of these scripts, they click on the rectangular symbol next to the play button and this brings up all the inputs within the file and allows them to be set properly before the play button is pressed.

A script that does not have “(inputs)” at the end of the name does not have any editable inputs and the play button can be immediately pressed for those scripts.

Dynamo Player Tips

Dynamo Player is a simple and easy tool to use, but there are a few tips and tricks that can make it even more user friendly.

Display Options, Notes or Warnings through Player

For a node to be an input in Dynamo Player, the user must right click on the node and select “Is Input” in Dynamo. Then when the script is run through Dynamo Player that node and its contents will show up as an input in Dynamo Player. However, not all inputs necessarily need to be actual inputs. When a node is specified as an input in Dynamo, all that means is that it will display in Dynamo Player, sure the user can edit it to change the input, but it can also be used to simply pass along information. A good example of this is illustrated in figures 43 and 44.

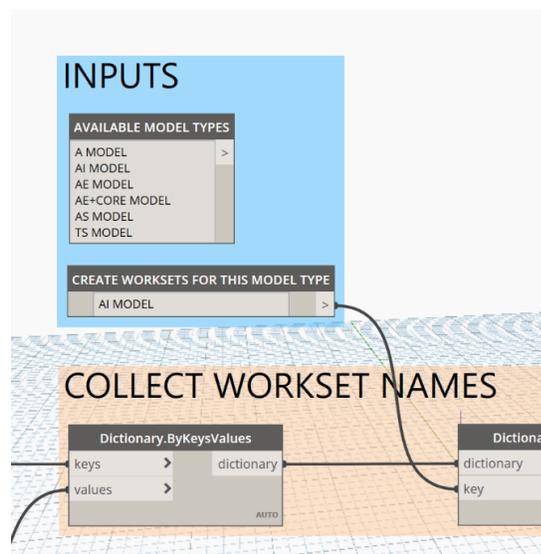


Figure 43: Dynamo Inputs

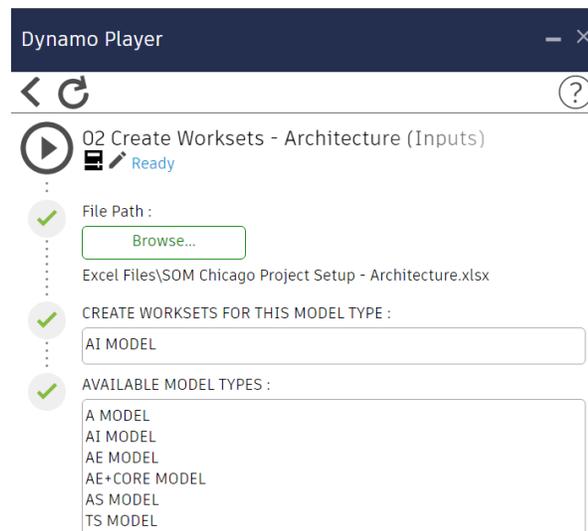


Figure 44: Inputs in Dynamo Player

Figure 43 is the Dynamo script. The node called “Create Worksets for this Model Type” is an actual input. The user must type in the model type they wish to create. However, the node called “Available Model Types.” Is not an real input. It is not even connected to the rest of the script, but because it is identified as an input it displays in Dynamo Player as seen in figure 44. This node is merely providing the user with all of the options available, so that the user does not need to guess at what the actual input should be.

This technique is used in many of the project set-up scripts to convey information, warn the end user to do something or not to do something, or to identify options as above. In this regard there are a few other things to keep in mind. All traditional input nodes in Dynamo such as sliders can be inputs. A string node can be an input. However, a code block cannot be used as an input. Lastly, as shown above, nodes do not need to be connected to the larger script in order to be used as an input and show up in Dynamo Player.

Specify All Drop-Down Menus as Inputs

If any nodes containing a drop-down menu in which the user must make a selection is used in a script, it is a good idea to make these nodes inputs, even if these nodes are meant to only display one choice and never change. What has happened in the past, is scripts were saved with a drop-down menu displaying one selection and then when the script is opened in Dynamo the node displays a different selection. If this happens in Dynamo Player, the entire script will fail when a node is displays the wrong selection. Therefore, make all drop down menu nodes an input, so that the user running the script through Dynamo Player at least has an opportunity to verify that the correct selection is displayed. If it is not, they have an opportunity to change it before running the script.

Create Outputs in Dynamo Player

If a user wants to see the output or result of a script displayed in Dynamo Player, a Watch node can be used for this purpose. Just like inputs, right click on the watch node in Dynamo and select "Is Output." Then when the script is run in Dynamo Player and finishes, the contents of the watch node will appear in Dynamo Player. Multiple Watch nodes can be set at outputs in the single script.

Don't Reinvent the Wheel: Project Templates

The more information and content a team has at their fingertips, the less likely they are to go out on the Internet or rummage through past projects looking for what they need. BIM Managers are in a constant struggle to prevent bad content and habits from finding their way into clean project files. The primary defense against this is having project templates that have all the information that a team needs, and providing easy access to any additional information or content needed by the team.

Purpose of Templates

The reason for project templates is simple. They should be used so that every project within an organization has the same set of graphic standards, project organization and basic content. In Revit, there are two type of templates. Project templates which are widely used and customized in every office, and family templates which are often not customized for office use (but should be).

Family Templates

There is much less to customize in a family template. However, there are the usual graphic standards. These will be overridden by the project, but it is still a good idea to set up these standards in family templates so that the family displays in the family file as it will in the project. Beyond that, there are frequently used parameters that can be added to family templates, additional reference planes, etc. However, one of the most important and time saving things that can be added to a family template are materials. If the same default materials are added to family templates as exist in project templates, then materials can be easily assigned in the family file. Again, the materials will be overridden by the project if the material in a family file matches the material name in the project, but the materials are added to family templates for ease of use.

One thing to keep in mind regarding materials and family template files, is that by adding materials to a family template, the file size will increase, sometimes dramatically. However, do not worry about this. When a family is loaded into a project all the extra materials that are unused in the family are essentially purged and the file size of the actual finished family within the project file will be much less than the original template file size. SOM has tested and benchmarked this process proving that loading materials into family templates does not adversely impact the performance of project or family files despite the larger size of the templates.

Office Project Templates

There are many ways to provide templates to an organization. Some organizations may have one template for each of the major building types that they design such as a hospital template, office template, school template, etc. Other organizations may have a template for concept design and another template for design development or construction documents. In either case, there are many different files to manage and upgrade when the time comes each year. For BIM Managers, this can be a nightmare and is time consuming. The small benefit of using multiple project templates is far weighed by the effort that must go into maintaining them. Especially when there are many other much more effective ways of achieving the same result without the use of multiple templates.

It is recommended that each organization develop and maintain just one single project template for all project types and project phases. Of course, if an organization has imperial and metric

Creating A Saved Group

1. In a new project file, load all the required families and within the file, place all the family types in the model that should be included in the saved group. Ideally, the families are placed and arranged in a methodical manner so that the user can easily see what is contained in the saved group.
2. Select all the content and create a new group. Remember the name of this group will become the name of a new Revit file, so choose a name appropriate for a Revit file, or follow the applicable office naming convention. If a series of these files is created, consider using a name like "Family Collection - <Category or Project Type Name>."

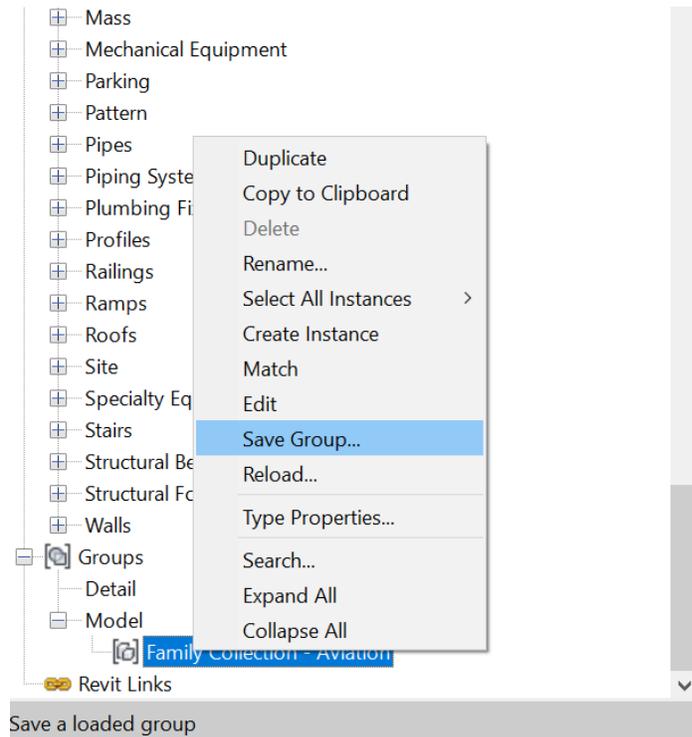


Figure 46: Save Group

3. Go to the project browser, scroll down to the bottom and find the list of groups in the file. Right click on the group to save and select "Save Group..." from the pop-up menu.
4. Save the group in a location accessible by all.

After the group has been saved, it will be a regular project file. However, the file will only contain information needed for the group and be stripped of all other information that was contained in the original file.

The saved group file can be edited, family types added, deleted, etc. However, do not waste any time making changes to the file other than content related adjustments, as any non-content related changes will have no impact on the saved group.

Loading a Saved Group

1. Open the file in which the saved group is to be added.
2. On the insert Tab, select “Load as Group,” and select the desired saved group file.

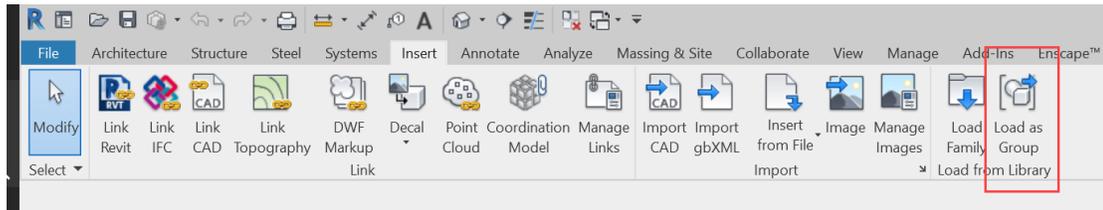


Figure 47

3. The group will be added to the file and the name of the group will display as a loaded group. The content contained in the group is now loaded into the project file. The group can remain in the file or be deleted. If the group is deleted the content will remain in the file.

Tip: Only the items added to the group will be loaded into the destination file. Therefore, if a family has 6 different types, and all 6 types need to be loaded into a project file, then all 6 types must have an instance included in the group. Including a single family type in a saved group, does not automatically include all the types contained in the family.

Schematic Families

If there is just one project template in use. It will need to include schematic level families. These families are small, so even if they are not used they are not a burden on the template. The question is which schematic families should be included? That depends on the type of work being done in an organization. As discussed in the previous, “Designing in Revit” section, these families can consist of casework, elevators, risers, glass walls, simplified curtain walls and/or panels and more.

To make sure that this content is not accidentally used in later design phases, the names of the families should somehow reference the fact that they are schematic in nature. For example, “Cabinet – Base – Schematic.rfa” signals to the end user that this is a schematic family and should not be used in Design Development or later phases.

Color Schemes

Color floor plans and sections are the backbone of documenting most projects in early design phases. However, in many offices, standardized color schemes are either not defined and recreated on every project or the desired color scheme is not included in the project templates. If color schemes were included in an office’s project templates the process of producing color floor plans and sections would go much faster and eliminate all the questions about which colors to use.

Standardizing Color Schemes

It cannot be stressed too strongly that any organization that regularly uses color floor plans and sections should implement a uniform color scheme for all their projects. At SOM for a long time it seemed that every studio or group within the firm used a different set of colors and standards on their projects. These color schemes had to be established from scratch in every file. That was until a few years ago when a uniform color scheme was implemented across all studios and this color scheme was integrated into the SOM project template.

Now when a team needs to produce a color floor plan, all they do is fill in the appropriate room parameter, such as “Department” or a custom parameter called “Program” the standard color is automatically deployed. There is no guessing or set-up necessary. It all happens automatically.

The key to setting up a standard color scheme is using past projects to come up with what the standards should be. Look back over many different project types and identify all the different room/area types that appear in the firm’s projects. Then make note of the colors that are used, and come to an agreement on which colors should be used if there are conflicting colors found.

Once a standard is identified, it should be added to the project template as a color scheme. There can also be multiple color schemes loaded into a template. For example, there could be one color scheme based on “Department,” and another based on “Occupancy” or another custom parameter. Figure 48 is an example of a comprehensive color scheme, that also builds in some flexibility for personal preference or different situations.

COLOR CHART

PROGRAMMING COLORS FOR DIAGRAMS.
USE ROOM / AREA / MASS OBJECTS WHERE APPROPRIATE.

COLORS ARE COORDINATED WITH COLOR SCHEMES AND MATERIAL ASSETS.
ANY CHANGES MUST BE UPDATED IN ALL LOCATIONS.

COLORS SELECTED BY HUE / SATURATION.
SHADE VARIES BY +/-15 LUMINANCE

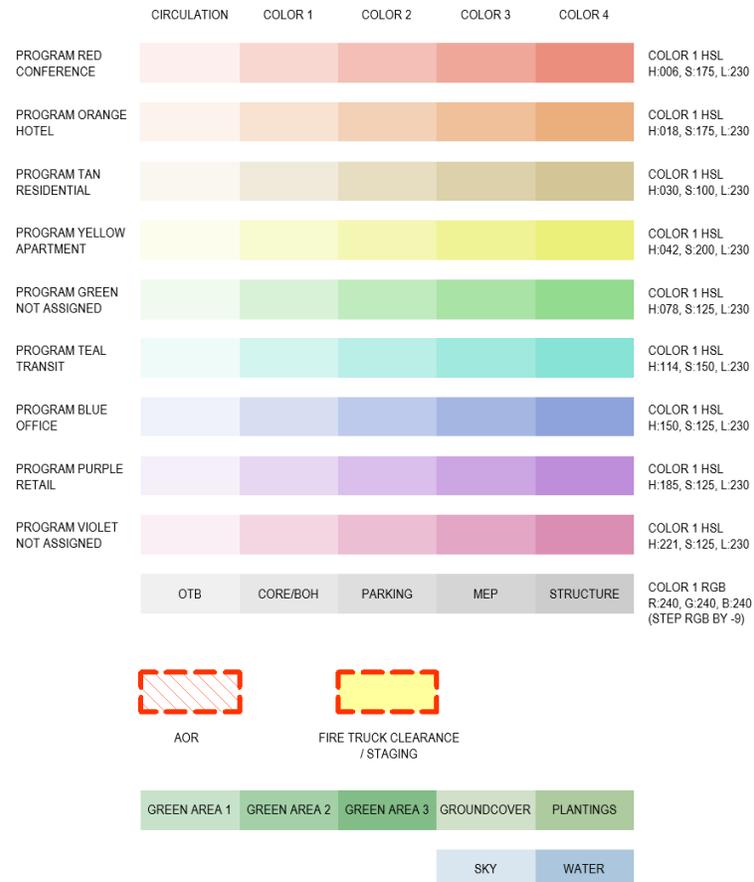


Figure 48: Color Scheme

One of four different colors can be applied for each programmatic area. In Revit the user chooses Conference 1, 2, 3 or 4, Hotel 1, 2 3 or 4 etc. This way the colors can be pale and light or more saturated depending on the final output. However, all of the colors and programmatic designations are pre-determined in the template.

Standards & Best Practice Notes

As stated at the beginning of this section, the more information teams can have at their fingertips, the less likely they will be to go out and seek information from other, potentially faulty sources. One of the areas that is often overlooked in this regard is office standards and protocols. Even though many organizations have a Standards Manual or an internal website that covers all the organization's standards and best practices. These sources are often still too far away for users to reference them regularly. But what if the most important information was right there in the Revit file, right at the user's fingertips?

For example, Drafting views can be established in the template that outline the standard line patterns, line styles, text styles and any other graphic standard. Other drafting views can be used to provide guidance on difficult items in Revit such as stairs and railings. Still other drafting views can explain what the standard sheet order should be or what the naming convention is for views. Figure 49 shows an example of the "Project Information" views that can be included in a template. These are the views included in the SOM project template.

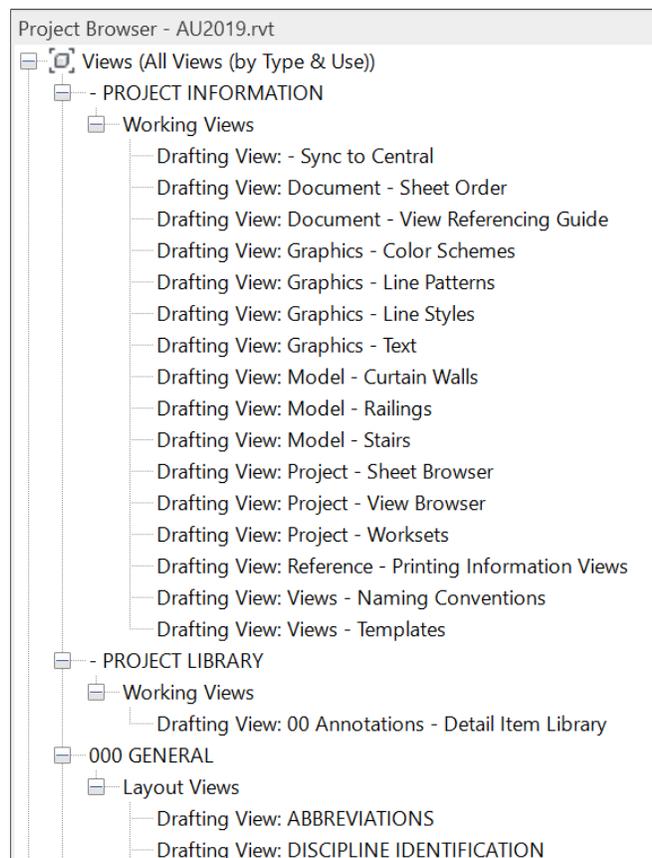


Figure 49

Materials

With the growth and popularity of programs and add-ins such as Enscape, TwinMotion and Lumion, it is more important than ever for Revit models to contain high quality materials, especially in the early design phases. In fact, it is a difficult problem to manage. On the one hand, in all other areas of this handout it is advantageous to have less detail and more abstraction, but render programs demand more detail and accuracy. One way this can sometimes be mitigated is by using high quality materials. Sometimes the materials can show details that the 3D model may be missing.

Render Quality, Render Ready

All materials in a project template should be render quality. This means that the “Appearance” tab for every material is complete and contains suitable bitmaps and textures. This includes all materials used in schematic or generic families. A generic material may render as a light gray with a slight texture, or maybe white. All glass walls should render as glass, but maybe with a tint. The point is that generic or schematic walls and other content will render with the same quality as a defined masonry wall or metal panel. The only difference is that the generic materials are undefined as to what they are. They are placeholders while the other materials are clearly defined. This will ensure that no matter what material is chosen in the file, it will render correctly and have a level of quality appropriate for Enscape or similar programs.

Of course, even though it does not play a role in rendering, the “Graphics” tab for all materials should also be complete and include all required fill patterns. This will ensure that materials not only display properly in renderings by also in Shaded, Consistent Color and Hidden Line views.

Materials Library

The best way to control material quality and consistency is to create a custom Materials Library that can be used by anyone in an organization. However, the term “Materials Library” can be a little misleading. A Revit Material Library does not actually include complete materials, with graphic settings, appearance, physical and thermal assets. In most cases a custom material library consists of appearance assets that can be used to create new materials in a project file.

Having a custom material library is important for organizations that use rendering programs such as Enscape. The custom library can contain reliable high quality appearance assets for materials that are commonly used within the office. The contents of the library have been vetted and approved for use. If the library is large and covers a wide spectrum of materials, team members will be less likely to go out on the internet or find other sources for materials that have not been vetted or approved for use.

The process of creating a library is easy, but it is not exactly user friendly or intuitive. It can be confusing at first, but once a library is established it is easy for team members to use.

Creating a Custom Materials Library

To create a new library, open the Materials dialog box and in the lower left corner is a folder button. Click on the folder button and Select “Create New Library.”

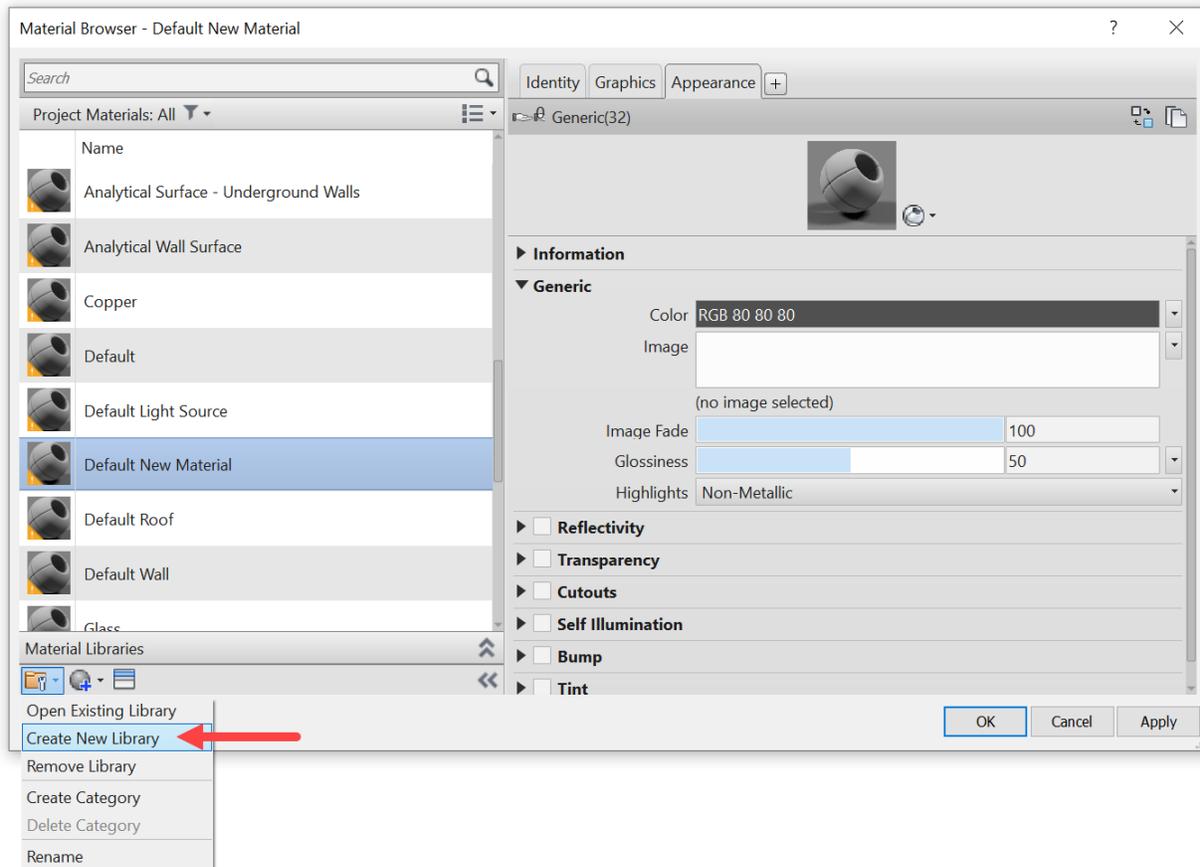


Figure 50: Create a New Materials Library

The user will then be prompted to save a new “.adsklib” file. This file should be saved in a location accessible to everyone within the organization. It is recommended that the file be saved in a write protected location, so that team members do not have the ability to make changes to the library, but can use the materials within as-is.

This will create a new materials library file, but it will be empty. Now the library must be filled with materials. This can be done by right clicking on any material from any library and selecting “Add To..” then select the new library. It is an easy process to build up a custom library, but it is a time-consuming process, because the materials must be loaded one by one. There is currently no way to batch load materials into a library.

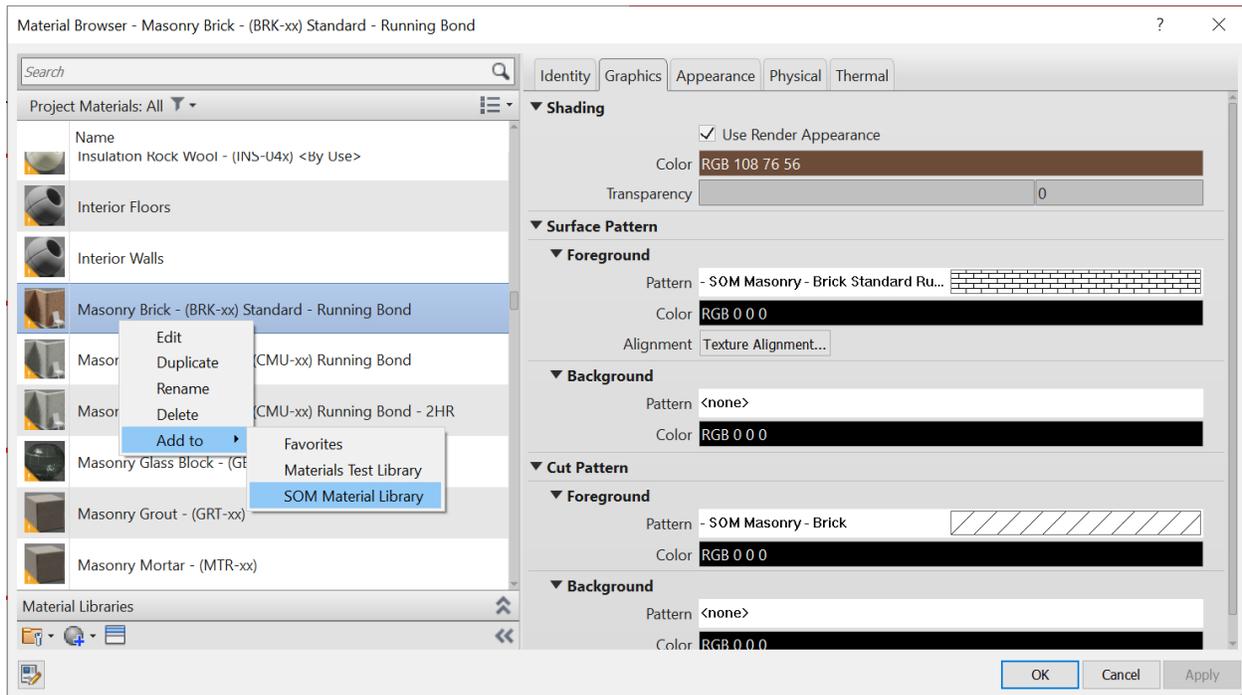


Figure 51: Adding to a Materials Library

When all the desired materials are loaded into the new library, the materials can be put into categories. This is not a necessary step, but can make it easier to find materials. The categories function like folders, so if someone is looking for a fabric, they can click on the fabric category and only see the available fabric choices.

To create a category within a library, go to the asset browser. It can be reached by opening up the material dialog box and then clicking on the button in the lower left that looks a little bit like the asset browser. Once the browser is open select the custom library that needs categories and then right click on the library name and select “Create Category.” Give the new category a name. To add materials to a category, click on a material in the custom library and drag and drop it on the category name. The material will be added to the category. If the material already exists in the category Revit will display a circle with a slash through it symbolizing “No.”

In order for categories to be saved, they must contain at least one material. Revit will automatically delete empty categories when the custom library is closed.

Creating a New Material from a Custom Materials Library

To use a material from a custom library, load the library by clicking the folder button in the lower left corner of the material dialog box and select “Open Existing Library.” Browse to the location of the material library (the .adsklib file) and select it. The library will become available within the asset browser.

1. Open the Materials Dialog box and click on “Create New Material”

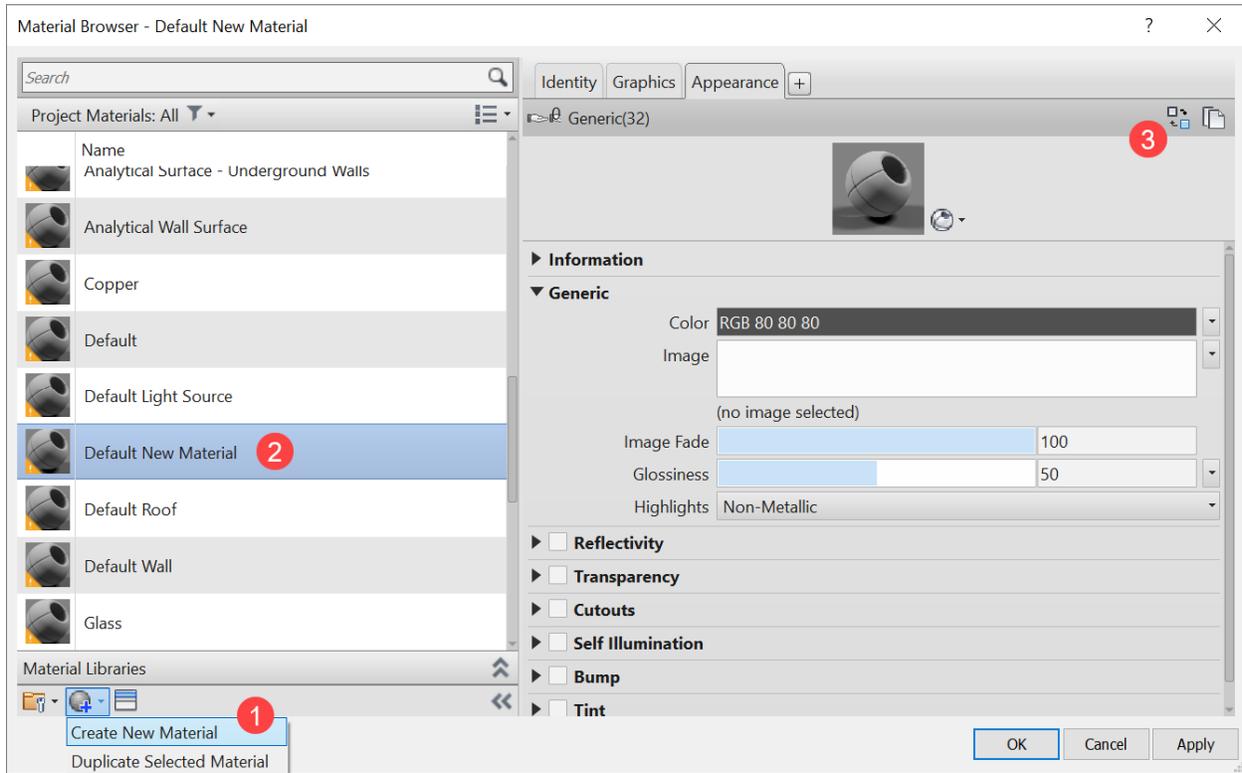


Figure 52: Creating a New Material

2. Click on the Default New Material and give it a new name.
3. On the Appearance tab, click on the icon with the rotating arrows in the top right to select a new appearance asset
4. On the Asset Browser, select the custom library, and the appropriate category if applicable. Then finally select the desired appearance asset. To add the asset to the new material, click on the arrows that appear on the right hand side of the asset.

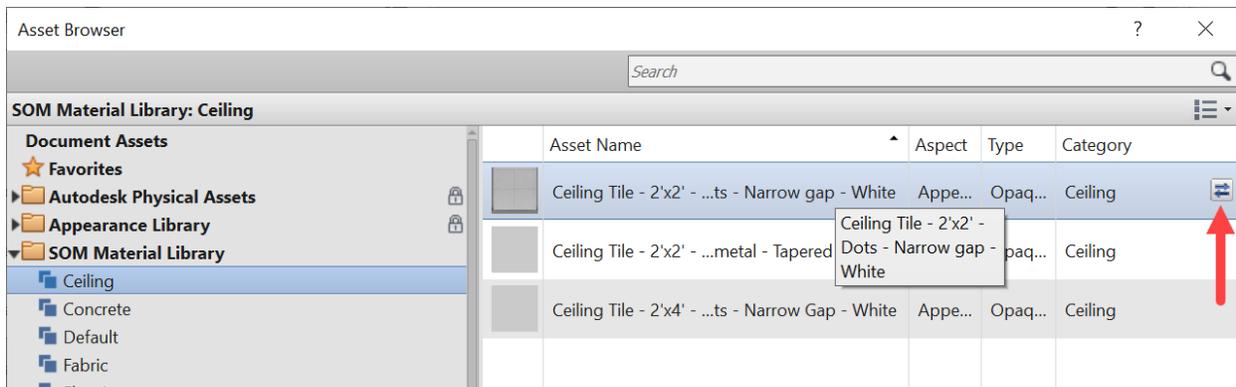


Figure 53

Control the Chaos: Project Management

All BIM Managers and anyone who has taken a project from concept to construction knows that it can be difficult to keep a Revit model clean, organized and conforming to office standards. Particularly during concept and schematic design, Revit models can be extra difficult to control. The design is changing every hour, there are design options and alternate schemes, not to mention multiple documentation sets and different forms of documentation to juggle, all with a very tight schedule and no time to manage the model. At times, it may feel like the models are being held together with tape, rubber bands and a little bit of luck. However, there are a few workflows and techniques that can be implemented that will allow a BIM Manager and a project team to stay on top a Revit model in early design phases and not allow it spin out of control.

Standards & Best Practices

Hopefully, all organizations have office standards and best practices for how to build a Revit model and all team members are trained in those standards. However, many times those standards are created for later design phases, when the design is not changing every day and the primary task is technical document production. In early design phases, there are different demands and sometimes different deliverables. Therefore, it can be more difficult for teams to adhere strictly to the office standards and best practices. Even though it may seem contradictory, during early design phases, BIM managers may need to loosen their grip on standards and be open to alternative best practices for the sake of rapid change and flexibility.

Early design phases demand rapid response, whether it is to explore different designs or produce atypical deliverables such as diagrams or rendered floor plans. The demands placed on a team can be very different from later design phases. Teams should be able to bend the rules a little bit for expediency sake. However, there are a few hard and fast rules that should never change.

- Do not allow deviations from anything covered in the project's BIM Management Plan.
- Do not allow any graphic standards to be altered such as line styles, line weights, line patterns, object styles, filled regions, etc. However, view templates can be used to alter the appearance of views if the underlying graphic standards remain unchanged.
- The best practices that govern how families are to be constructed should always be followed, because some families may be used throughout the life of a project.

Accommodating the Design Process

One example of making a slight change to office standards to make the design process easier happened on an SOM project known as Kunming Poly. The normal workset standard is to have the entire exterior wall on a single workset. However, the design of the project is rather unique in that the building has an exterior wall that faces outward and exterior walls that face inward and opposite each other. This arrange of walls made it more difficult to model because it was often difficult to visually distinguish between the two different curtain wall systems when everything was in a single workset. Therefore, the team decided to bend the rules and break the exterior wall up into multiple worksets, allowing the two different curtain wall systems to be isolated in different views, making it easier to edit the walls.



Figure 54: Kunming Poly

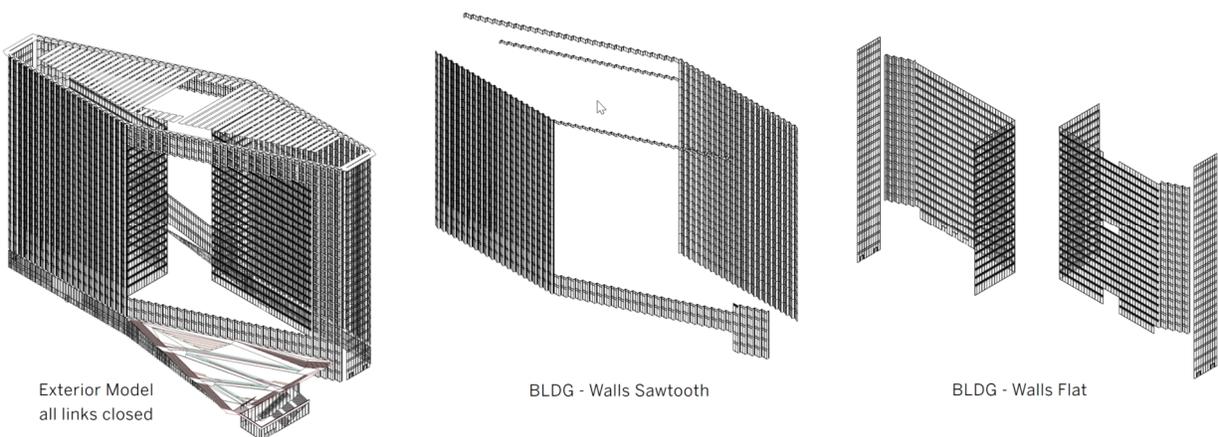


Figure 55: Exterior Wall Worksets

In figure 55, the original workset arrangement is shown on the left. By splitting the two curtain wall systems out into two different worksets the team could isolate those two systems making it easier visualize each system and the extent of each system, as well as making it easier to edit the curtain walls.

Of course, this class and handout are essentially proposing a set of best practices for early phase design. Once an organization has done a few projects in Revit through the early design phases, they may be able to devise their own set of office standards and best practices specifically for the early design phases and integrate those workflows into their overall office standards.

Dynamo

One of the best ways to control a Revit file, or get a disorganized and chaotic Revit file back in line, is by using Dynamo. Dynamo is not only great for design studies and form generation, but it can also be used as a project management tool.

At SOM, there is a Dynamo script library saved on a network drive. The library is accessible to all, but can only be edited by the firm's BIM leadership. The library is a collection of several different types of Dynamo scripts, each with its own purpose.

- *Specialty Scripts*: These scripts were developed for a highly unique purpose on a past project and because they are so unique, they have no realistic application on other projects. However, these scripts are in the library to serve as workflow examples or inspiration for how to achieve a goal. They may not be able to be recycled, but portions of them might help users figure out how to solve a problem or achieve a task.
- *Historical Scripts*: These are scripts from past projects that solved a problem or produced a certain result for that project. They probably can't be reused as-is on new projects, but with a little editing or customizing can be recycled. These scripts are meant to give teams a head start in producing new scripts for their project.
- *Generic Scripts*: These scripts may have been developed for a past project, or they may have been created without any project in mind. They are generic in nature and with very little to no editing can be used on any project. Nearly all the scripts used for project management fall into this category.

Within the SOM Dynamo Library there are a series of subfolders that attempt to group scripts by what they are trying to achieve. One of these folders is called "Project Management."

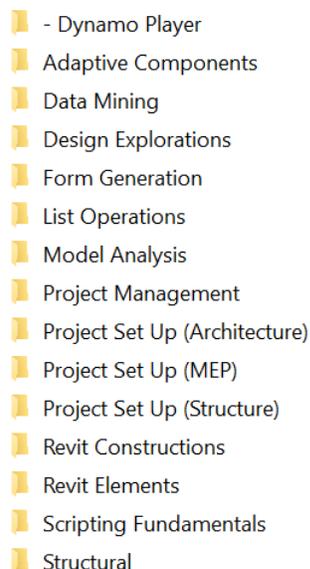


Figure 56: SOM's Dynamo script library

The project management scripts are used to organize a Revit file or get a Revit file that has been neglected back in line. Some of the most frequently used scripts are outlined below:

- *Worksets*: The primary script for worksets moves elements that are not on the correct workset to the correct workset. Of course, for some items it is impossible for a simple Dynamo script to know which workset every single element in a Revit file belongs. Therefore, this script focuses on the elements that have only one correct answer for their workset. These are elements such as rooms, room separation lines, reference plans, levels, grids, scope boxes, CAD links and Revit links.
- *Project Browser*: Sometimes team members put views in the wrong location in the project browser or views stay in “???” There is a series of scripts that move views to their correct location in the project browser and/or rename views to office standards.
- *Views*: A variety of view related scripts are designed to keep views in line with office standards. Some make sure that all Layout View names are written in uppercase and Working View names are written in lower case. Other scripts assign the appropriate view template or remove view templates from views that should not have one. There is also a script that creates a series of views and crops them to a designated scope box.
- *Room Separation Lines*: SOM often designs towers that have doubly curved facades, and therefore must use adaptive component curtain panels for the exterior wall. However, adaptive components cannot bind rooms, so room separation lines must be placed at all the floor edges (or a defined offset from the floor edge.) There is a script that can produce these room separation lines and update them when the floor shape changes. In addition, this script along with some visibility/graphics changes can produce early design phase floor plans that appear to have exterior walls when there is no exterior wall modeled. The room separation lines stand in for walls.
- *Finishes*: There is a script that adds finish floors based on rooms, as well as ceilings.
- *Data Management*: There are several scripts used to collect data, some export the data to Excel. Other scripts add data to elements within a project file, such as number parking stalls in a designated sequence or add room numbers to rooms based on level and location.

These are just a tiny fraction of the scripts that SOM uses to control and manage their project files. These scripts may be run on a file one time to get a file back in shape or run repeatedly throughout the life of a project. For scripts that need to be run regularly, the Revit add-in called Imaginit Clarity can be deployed to run nearly any Dynamo script automatically on a designated schedule, thus eliminating the need to track what otherwise would be a manual task. There is more on Imaginit Clarity, later in this handout.

Project Specific Templates (Project Masters)

In the previous section of this handout, the importance of general purpose project templates was discussed. However, in most cases there is also a strong argument for creating a project template that is specific to an individual project. These project specific templates are referred to as a “Project Master” file. SOM uses this term so the files will not be confused with the Office Project templates. The Project Master files always begin as a copy of the office project template. A Project Master file has two basic functions:

1. Used as a template to create all new project files for a project.
2. Collect and transfer updated settings and content to all project files.

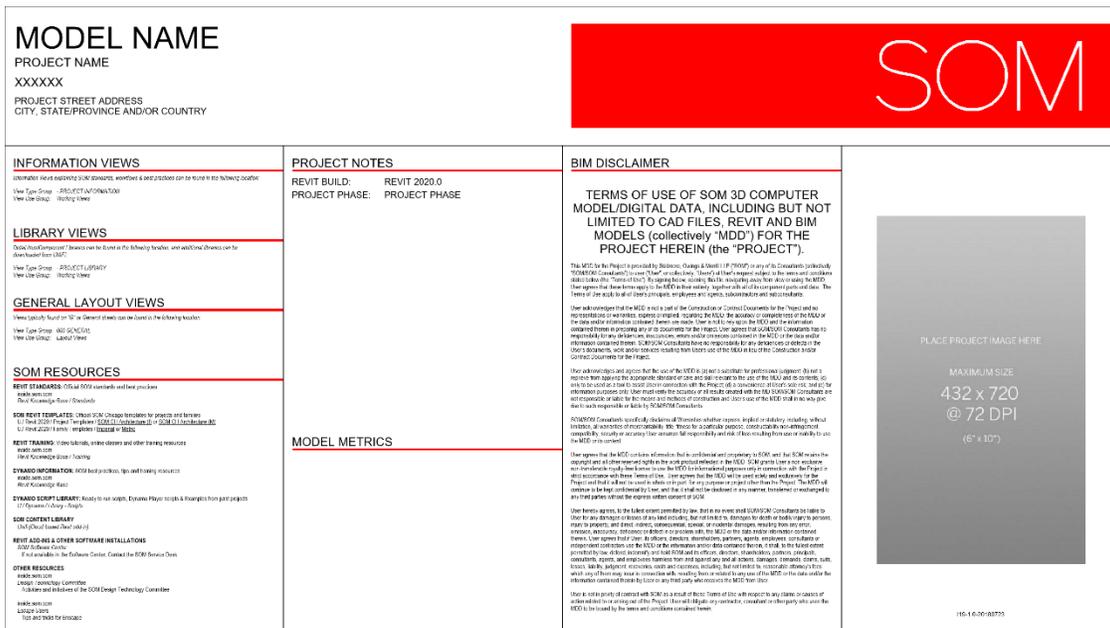


Figure 57: SOM starting view (sheet)

Create New Project Files

Of course, new project files can always be created from the office project template. However, projects can last a long time. A project that starts today, may not be completed for another 5, 7, 10 or more years. Over that time there are software upgrades and many changes, not only to the settings and content within the existing project files, but with the software itself. As a result the office project templates may change over time. The project template used to set up files in year 1 of a project, may not be the same on year 5, resulting in project files with different settings and content.

Therefore, rather than always starting new files from the office template, and then spending time getting the new file up to date with all the other project files, start new project files from the Project Master file and all the new files will be instantly up to date with all the older project files – that is if the Project Master is kept up to date with the current project settings.

Transferring Settings & System Families

As a project progresses, slight alterations to settings and content may occur. These adjustments can be made directly to the Project Master file (if they are known ahead of time) and distributed to all project files via the “Transfer Project Standards” tool, or a Dynamo script. The key here is that one file contains the required information and distributes it out to multiple project files, ensuring that all project files are consistent. For example, if there is a series of custom wall types that need to be added to the project, these wall types can be added to the Project Master file and then can be distributed out to all project files through “Transfer Project Standards” or a Dynamo script.

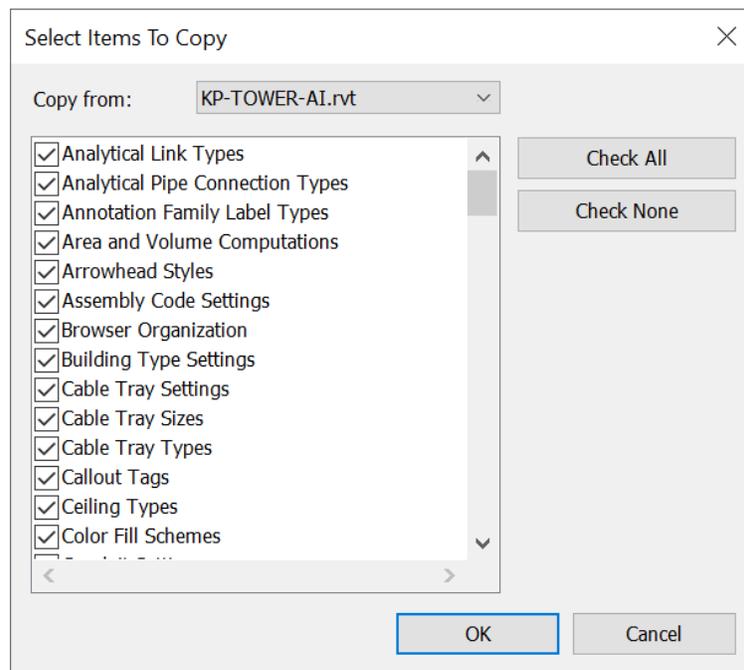


Figure 58: Transfer Project Standards

However, it is more likely that adjustments or additions are made on the fly in one of the project files. Then at some point it is determined that all the other project files need to have the same adjustments made. In this case, the changes should be transferred to the Project Master file first, so that it always contains the most up to date information, then the changes can be distributed from the Project Master file back out to all the other project files. This is all a very simple process if the transfer between files can be achieved through the “Transfer Project Standards” tool, but what if loadable families need to be transferred from one file to another? The “Transfer Project Standards” tool cannot be used for loadable families. This is where Saved Groups comes into play.

Transferring Loadable Families

How to create and work with Saved Groups is discussed as part of the previous section in this handout. The question now is how can Saved Groups be used in the context of a Project Master file? First, saved groups should never exist as a placed group in the Project Master. If groups are placed in the file, when the Project Master is used to create a new file, the new file would then contain groups placed in the model.



Figure 59: Furniture saved group

The key is that Saved Groups do not need to be placed to be part of a file. Just like any other group, they can exist in the project browser listed under groups, but not be placed anywhere in the model. Thus, when a Saved Group is created from a project file, or from another source it can be loaded into the Project Master file without the group showing up in any view. The group is still a part of the Project Master and therefore, if a new file is generated from it, the new file will also contain the group and all of the content within the group.

If a group needs to be edited, for example a few families added to the group. The project Master can be opened, the group can be temporarily placed, edited and then saved out as a group again, overwriting the original file. Once the group is safely saved, the placed instance of the group within the Project Master can be deleted, and the Project Master saved with the updated group.

Likewise, families can be added to a Project Master, put into a brand-new group and then the new group can be saved out as a saved group file. Once it is saved out as a separate file, the placed instance of the group in the Project Master can be deleted.

Lastly, a Dynamo script with some custom python can be used to copy groups from one Revit file to another Revit file. Despite the “Load As Group” command not being exposed to the API, there are ways to copy elements from one file to another, including groups, using Dynamo.

Project Notes

In the previous section, there was a discussion regarding the inclusion of notes on Standards and Best Practices into project templates. These are notes that apply to all projects. However, there is often plenty of other information that is specific to a project that designers may need to reference repeatedly throughout the early design process. This information could be as simple as the number of elevators required and what floors the elevator zones should be. As the design changes and new configurations are tested, the designer could reference the “Vertical Transportation” drafting view to make sure that they are still meeting all the requirements. Other examples are a project’s design criteria or programmatic requirements, parking requirements, even important sections of the building code could be included – such as the code section that covers the number of bathroom fixtures required.

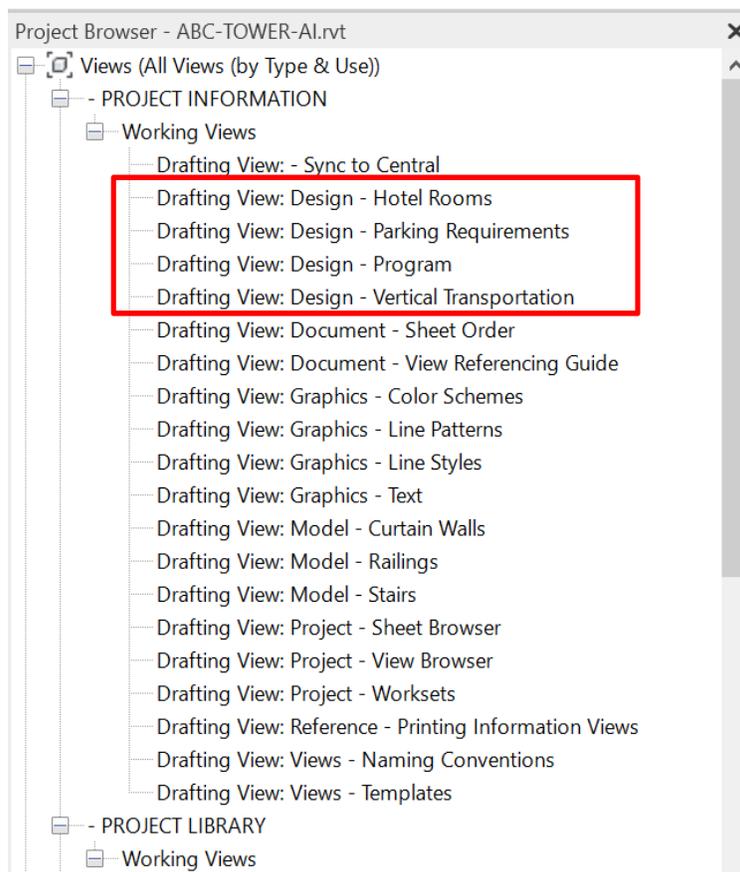


Figure 60: Project Notes

In most cases, project notes of this type are coming from another pre-existing source. Therefore, rather than typing the information into a drafting view, a jpg image can be inserted into a drafting view, and now with Revit 2020, a PDF can be inserted into a drafting view. This eliminates most of the work required to include this information. Most importantly, it puts the most critical information at the designer’s fingertips, right inside the Revit file. Thus, there is no need to search for the information in the project directory, a code book or the Internet.

Content Libraries

Where do team members find schematic content for projects in early design phases? It often gets mixed in with all the other families in a library and can get lost. For organizations that save their family libraries locally in a folder structure, consider creating a special folder called “Concept & Schematic Families” and save all the schematic families there.

For organizations that use a service such as UNIFI, it is much easier. Create a Saved Search called “Concept & Schematic” and filter the saved search with a tag such as “Schematic.”

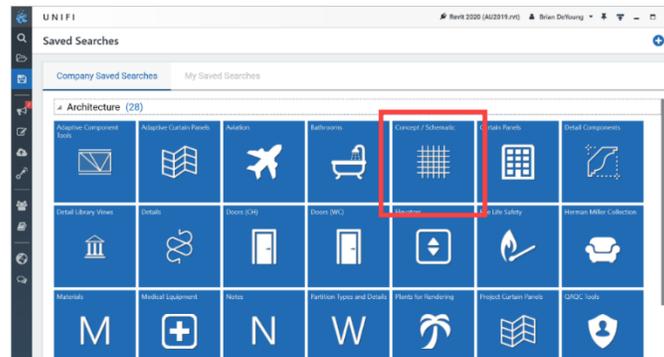


Figure 61: UNIFI Saved Searches

UNIFI

Unifi is a Revit add-in that allows an organization’s content library to be hosted in the cloud. There is no longer a need to have families saved locally in a collection of folders. UNIFI provides a wealth of tools for tagging, sorting and organizing content. It also allows not only families, but drafting views, sheets, schedules, materials and more to be saved in the library, as well as non-Revit content.

The goal is to make it easier for teams to find content appropriate for early design phases, and by grouping families into a “Schematic” saved search or a local folder, it helps team members find the content they need easier and faster.

Structural Coordination

At the beginning of a project and sometimes through most of the schematic design phase there is no structural Revit model, or if there is a Revit model it is not suitable for the architects to use as a link in their architectural models. This leaves the architect in a difficult position. The major structural members need to be represented in some fashion in the architectural plans, but the structural engineers have not produced a Revit model yet.

This happens for a variety of reasons, but primarily because the structural engineers have a different focus in the early stages of a project. They are concerned with structural calculations and coming up with a structural design that mathematically works. There is a heavy reliance on structural analysis software in the beginning and producing a Revit model is often not a top concern.

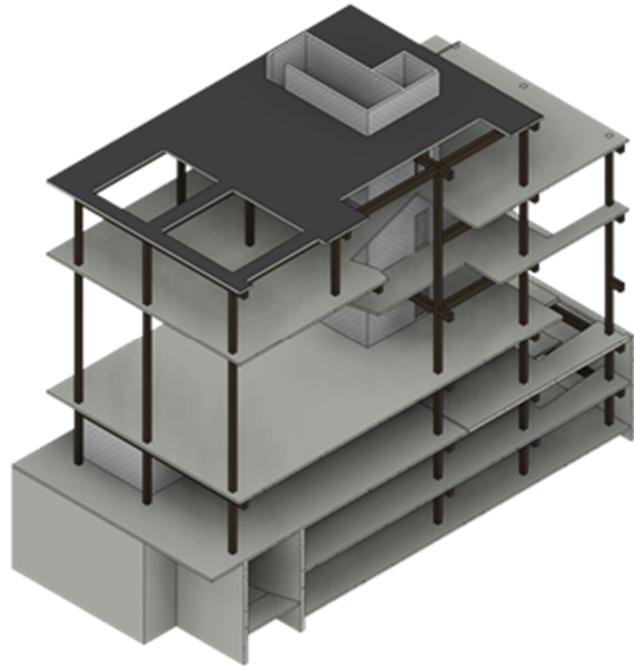


Figure 62: A typical “TS” Model

Whatever the reason, this forces the architect to model key structural elements within the architectural model. These elements are modeled based on preliminary information provided by the structural engineers and are most commonly structural columns but, key beams and trusses are often modeled as well. It is critical that these items be modeled in Revit, both for early coordination and to make the architectural deliverables complete.

However, this workflow eventually leads to another problem. When a structural Revit model does become available, what does the architect do with the structure they have modeled in the architecture model? It is never a good thing to have two versions of the same thing on a project. There should always be a single source of truth, and the single source of truth for a project’s structure should be the “S” model or structural model. The problem is the structural model is linked into the architecture model and there are discrepancies between the architect modeled structure and the engineer’s structural model. Ideally, this is worked out in the traditional coordination process, but what often happens is the architect will turn off the structural model and continue to use what they have modeled, leading to unaddressed coordination issues, or the sense of “false coordination” in which the architectural model appears correct, but it is actually not properly coordinated. How can this be resolved, or at least improved?

TS Model

At SOM, a partial solution to this problem has been for the architects to stop modeling structure in the architectural Revit files and instead create a separate model called a “Temporary Structure” model or “TS” model for short. In the TS model the architects model all the structural elements

that they need to show in their views. This usually includes structural columns and a few beams, maybe a truss of two, but not the entire structure. It should also include the structural floors and roofs. The architect only models what they need to show in their views and any structure that must be accommodated architecturally.

The architect spending time modeling structure is obviously not ideal, so it should be done in a way that is quick and easy. The architect should model only what is necessary for their documentation, but the tools used to build this content should also be simple and easy. Revit's structural tools and families are often confusing for architects. They work differently than the architectural tools. Therefore, structural elements are rarely, if ever, used in the TS model. Likewise, the TS model should never be seen as a legitimate structural model so avoiding real structural elements to build the model reinforces that fact.

Most of the time the “structural” columns in a TS model are just architectural columns. Beams are line-based generic model families that have been designed for ease of use by architects. Trusses are often loadable generic model families or sometimes in-place families. Lastly, the structure is abstracted. For the architect, it is the size of the structure that matters, not the exact steel shape (unless the structure is an architectural expression, then that is a whole other issue), so most beams are modeled as simple rectangular shapes, regardless of whether the beam is steel, concrete, wood or web joists. The color of these “structural” elements is also often a uniform color, which makes the elements easy to identify when the TS model is linked into the architectural models.

An example of a TS model for a small project is shown in Figure 62. The cut-away axon reveals that all columns, but not all beams are modeled and they are all simple rectangular shapes. All structural floors and structural walls are also modeled. The camera view in Figure 63 reveals that only some beams have been modeled (but there are probably more beams modeled in the camera view than is typical for an average TS model). These beams are modeled to verify that the desired ceiling heights can be achieved and that there is space for mechanical ducts and equipment. Of course, all of this is done in schematic design so it is all very abstract and rough estimations.



Figure 63: Perspective View of a “TS” model

Advantages of a TS Model

At SOM, the use of a TS model has made the architects more aware and conscientious about the structure of projects. But the TS model also solves a lot of annoying modeling issues that can arise when the structure and architecture are modeled in the same file. When structure and architecture are modeled together, how walls join can be a constant battle. Finish walls improperly join to structure, or finish walls simply do not clean up at corners or at openings because of conflicts with structural walls. When the two different wall systems are in two different models, these problems disappear along with the architect’s headaches.

The extent of rooms and the resulting color schemes can also be problematic when structure and finish walls are in the same model. The color floor plans can develop white stripes between structure and finish walls due to two room bounding walls being back to back and not allowing the rooms on either side to touch in doorways.

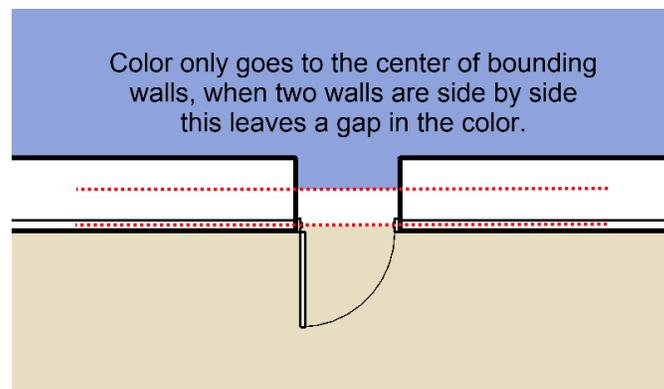


Figure 64: Color gap resulting from structure and architecture modeled in the same file.

There are fixes for this, and some unfortunate hacks, but regardless this causes extra work. However, the single biggest advantage of using a TS model is that it makes the transition to a real structural model extremely easy.

Structural Integration

When a complete and accurate structural Revit model becomes available, all the architect needs to do is open the *Manage Links* dialog box, click on the TS model and change the file path to point at the new “S” or structural model. Assuming shared coordinates have been properly established, the old TS model will be seamlessly swapped out for the new S model. No extra work required, except for the normal structural coordination that will be an ongoing process.

Likewise, if there are significant differences between the TS and S models that need to be negotiated between the architect and structural engineer, the TS model can always be relinked into the architecture model for comparison, or linked directly into the S model to visualize and help the structural engineers resolve the differences.

At SOM, the transition to a TS model workflow has been a little rocky, and some architects occasionally complain about having an additional model to manage, but the results have been

dramatic. The graphic problems that result from rooms and walls improperly joining or not joining are gone, and the architects are more aware of the building structure. But the most sweeping result of using a TS model is it has actually improved structural coordination, and made the transition to the use of a real structural model smoother and nearly seamless.

BONUS: Imaginit Clarity
(not in live presentation)

Clarity is a Revit add-in, but not a typical add-in. There is no button or menu inside Revit for Clarity. Instead, Clarity runs alongside Revit and users access it through an internal website. Clarity is a task automation and scheduling tool. It can be used to perform a wide variety of tasks on Revit models, saving teams hours of work. However, Clarity requires a substantial IT investment because it runs on a dedicated server or set of servers. The servers will open Revit, run tasks and close Revit on a schedule on by demand. It's like having a Revit robot added to a project team.

Clarity can take control of mundane time-consuming tasks such as printing PDFs or exporting DWGs, allowing the team to focus more on the design.

For example, many teams at SOM have Clarity publish PDFs of the entire document set each night, so that the team always has a fresh, up to date set of sheets every day. Teams also run many other tasks on an automated daily, weekly or monthly schedule.

Clarity can also help manage the Revit models and help control the chaos. Clarity can audit, compact and resave the central file, which keeps the file running smoothly. It can also collect model metrics on files. These are things like how many warnings are in a file, how many families are loaded, how many links are in use, and many more. All this information can be presented in graphs and charts that can help BIM Managers identify when files are getting into chaotic territory.

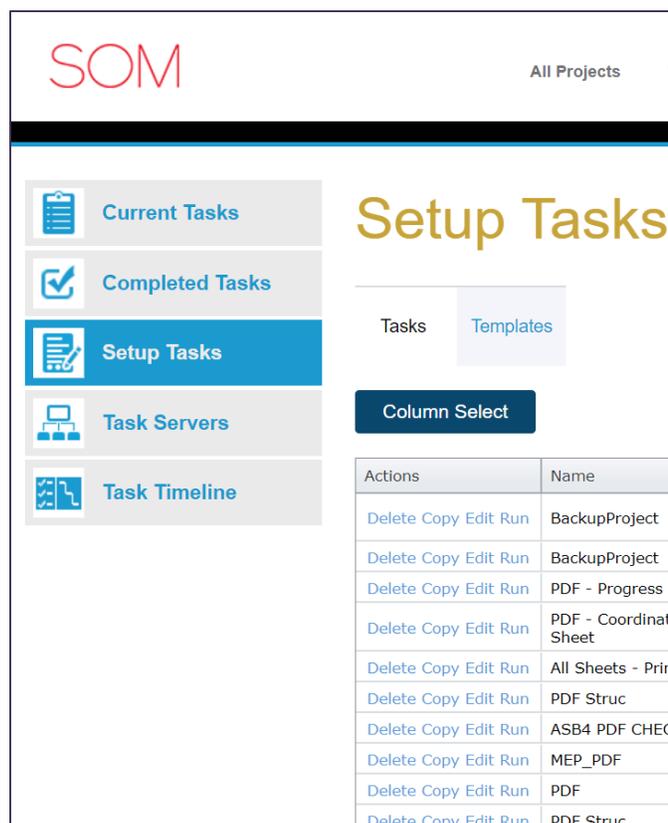


Figure 65: Imaginit Clarity Interface

Clarity also offers a task called “Performance Advisor” which will warn teams when model metrics are starting to impact the performance of a file and allow teams to correct the problem before it starts to cause files to crash or cause file corruptions.

Clarity can also run Dynamo scripts on Revit files. Therefore, the only limits on what Clarity can do are the limits of Dynamo. If an organization can support an IT investment like Clarity, it can save project teams hundreds of work hours over the life a project (no exaggeration), and at least some of the file management can be automated.

Lookout for Curve Balls: Options & Alternate Schemes

In the early design phases it is very common for an architect to explore more than one design scheme for the project, or explore multiple options for a certain portion of a project. In some cases Revit's built-in Design Options tool can be used for this. However, the Design Options tool works best for small isolated options. For very large design options or for full project schemes it breaks down and causes file sizes to become unsustainable. Therefore, a different strategy needs to be developed for large scale design options, which for the purposes of this discussion are referred to as "alternate schemes."

Alternate Schemes

At SOM, there are four major types of alternate schemes encountered on projects:

- **Core:** Quite often teams must develop multiple core schemes for high rise projects. Since the core is a massive part of any tower, using Revit's design option tool for core studies is simply not practical.
- **Exterior Wall:** There are often several exterior wall schemes in development at any given time. They may be as simple as exploring different curtain panel options, or as complex as different grid spacings or entirely different curtain wall systems.
- **Interior:** The term "interior" covers a lot of scenarios. Test-Fits can be considered a type of alternate scheme, as well as different office layouts or hotel room designs. These can vary from small and isolated which Revit's design option tool could handle all the way up to large and complex alternate schemes that the Design Option tool would not be able to handle.
- **Everything:** Occasionally at the very start of a project, a team may explore different concepts for the entire project. Concept A may be several low-rise buildings. Concept B may be one single high rise, while Concept C may be some combination of the two. Regardless, some alternate schemes may encompass the entire project.

Creating Alternate Schemes

If Revit's design option tool is not appropriate for the situations described above, how does a team create and manage multiple alternate schemes? They must develop a "DIY" Design Option solution. (DIY = do it yourself)

Three options for creating DIY Design Options

- Create new file – create an alternate scheme in an entirely new file
- Detach from Central and edit the detached copy to create and alternate scheme
- Transfer the elements that make up the alternate scheme from the existing file to another file and edit the new file

Each of these options have their own "best use" scenario, so all three could be deployed on a single project, given the right circumstances.

Alternate Scheme Option 1: Create A New File

This is the simplest way to create an alternate scheme. A team member opens a new Revit file, from the project's template file, runs the appropriate Dynamo set-up scripts and starts modeling the new scheme.

Best Use: Use this technique to create a new scheme that does not yet exist in any form and the new scheme is radically different from any version of the project that currently exists.

Alternate Scheme Option 2: Detach from Central and edit detached copy

In this case, a team member opens the original Revit file as a detached copy, saves the file with a new name and transforms the original design into the new alternate scheme.

Best Use: Create a scheme that is similar to and easily created by editing the original version of the project.

Alternate Scheme Option 3: Transfer the alternate version elements into another file

This is the most complicated method to create an alternate scheme, but may be the most common. It should be used when only a portion of a building requires an alternate scheme such as the core while the rest of the building does not have alternate schemes. This is done by converting a group to a linked file.

Best Use: The alternate scheme to be created is a clearly distinct portion of the design, e.g., the building core, and the existing portion of the design will be one of the alternate schemes.

How to Convert a Group to Linked File

Option 3 outlines above, relies on converting a group to a linked file. This is a built-in Revit tool, but it is often overlooked.

1. Create a group that includes all the elements that need to be a part of the alternate scheme. This includes, walls, doors, floors, openings, stairs, railings, furniture, and anything else that is 3D. Do not include any 2D content such as annotations. Levels and grids should not be included. (Levels will be regenerated automatically)
2. Once the group has been created, verify that all required elements are included. If the group is large and complex, this can be a difficult task. One technique that can make it easier, is to create a new workset specifically for the group. Before the group has even been created move the elements that will eventually be part of the group to this new workset. Set up a new 3D view in which all worksets are hidden except for the workset that contains the future group. What is visible is the current state of the to-be group and what will end up in the new file. If things are missing, create another 3D view and this time, hide only the workset that the future group is on. Now any items that remain in the view, but should be part of the group can be moved to the group's workset. When both 3D views

look correct, the group is ready to be created. In the 3D view in which the group is isolated select all items and create the group.

3. Convert the group to a linked file by selecting the group from any view and then click on “Link.”



Figure 66: Link

The group name will become the name of the new file, so name the group as if it was an independent Revit file, according to office standards. The new file can remain in the current file as a link, or the link can be removed.

4. To edit the new scheme, open the new file as normal. The file will not be workshared. If the file needs to be workshared, run the first couple of project set-up Dynamo scripts to enable worksharing and establish the correct worksets.
5. Update the file with the correct project/office standards. When a group is converted to a linked file, the new file will only contain the elements and data contained in the group. Revit will use Autodesk defaults for everything else. This means that the new file will be missing all office standards and custom project settings. To update the new file and get it in line with all other project files, open a new file using the Project Master template or an Office Template and then use the “Transfer Project Standards” tool to pull in all the desired project settings from one of those templates.

Managing Alternate Schemes

Now that all alternate scheme files have been created, how does a team manage them and integrate everything into an organized project structure? This can be the most challenging part of the process, but it is also where the idea of the DIY Design Option really comes into play.

The DIY Design Option

At its core, Revit’s built-in design option tool does two things. It controls what can be seen in one view versus another based on a set of properties, and it also allows more than one element to occupy the same space and time without conflict (or at least mostly without conflict). When coming up with a DIY solution for design options, it must also do these two fundamental things.

There are two key ingredients to DIY Design Options.

- **Linked Files** – Rather than elements being put into options, linked files function as the options. Linked files can occupy the same space and time without conflict (At least without conflict as far as Revit is concerned.)

- Worksets & Visibility Controls – Rather than controlling the visibility of elements through options, the visibility of the linked files is controlled by worksets.

Setting Up DIY Design Options (Managing Alternate Schemes)

In this scenario the documentation for all schemes comes out of one file. This is almost always the original file from which all the option files came. In this case, each of the option files is linked into the original file and each one is placed on its own dedicated workset.

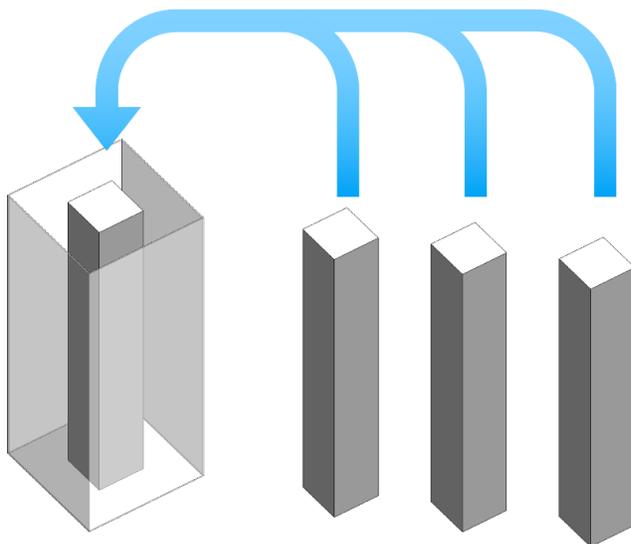


Figure 67: Original File

Core Schemes

Core Schemes

In the diagram in Figure 67, there are three different core schemes. Each one is linked into the original file. The original file does not include a core. The original core has been removed from file by putting the entire core into a group and converting it to a linked file. Thus, the original core becomes one of the core options. It is important that all the options are linked files. If one option is native to the original file and the rest are links, this process will not work as well or as intended.

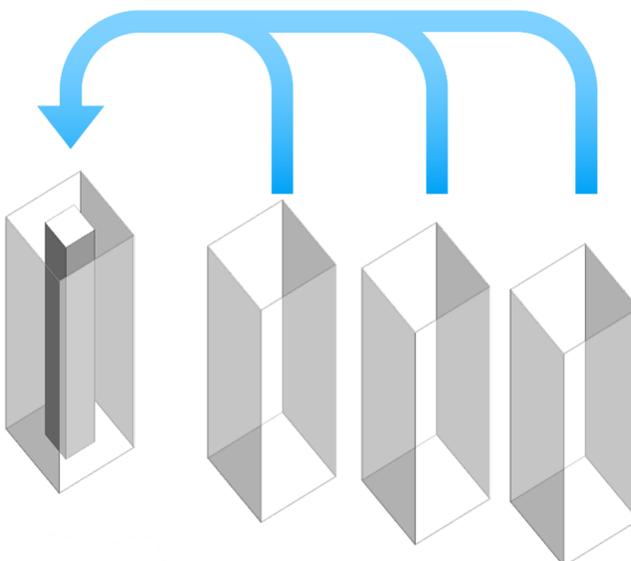


Figure 68: Document File

Exterior Wall Schemes

Exterior Wall Schemes

In the diagram in Figure 68, there are three different exterior wall schemes. On very large projects the exterior wall is often modeled in a separate file and is not a part of the file where the documentation is taking place. If this is the case, the exterior wall options are most likely created from detached copies of the original exterior wall file and the copies are the additional schemes. These files are therefore not linked back into the original file, but into the file where the project documentation is taking place.

Once all the alternate schemes are loaded into the original/documentation file and each one placed on its own dedicated workset it is time to sort out the visibility settings. There are a few different ways that the visibility of the links can be controlled, but the best way is to control their visibility across all views at once. Of course, visibility/graphics are view specific and only impact the current view. View templates are normally the solution to this problem, but in this case view templates the primary method of view control because separate view templates would need to be established for each combination of schemes to be shown. Therefore, teams should turn to the only visibility setting that is global and not view specific.

The Workset setting called “Visible in all views.”

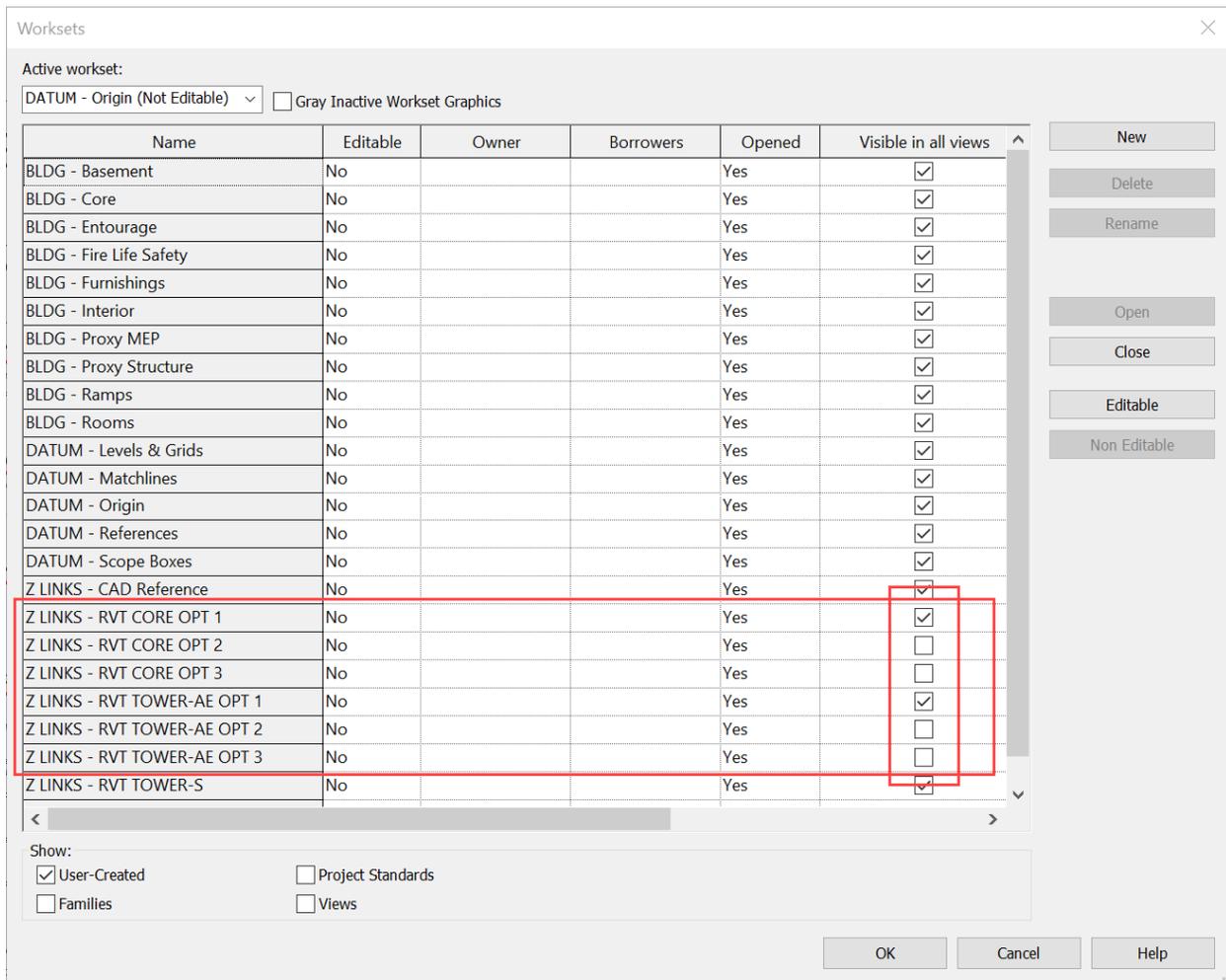


Figure 69: Workset Visibility

Notice that in figure 69, Option 1 for the Core and Exterior Walls have “Visible in all views” checked, but the other two options are unchecked. This set-up establishes the Core and Exterior Wall Option 1 as the “primary” option. Those options will now be visible by default in all views, and all other options will be turned off by default.

If a floor plan or elevation needs to display Option 2, The “primary” option is overridden by changing the workset visibility for that individual view as shown in figure 70.

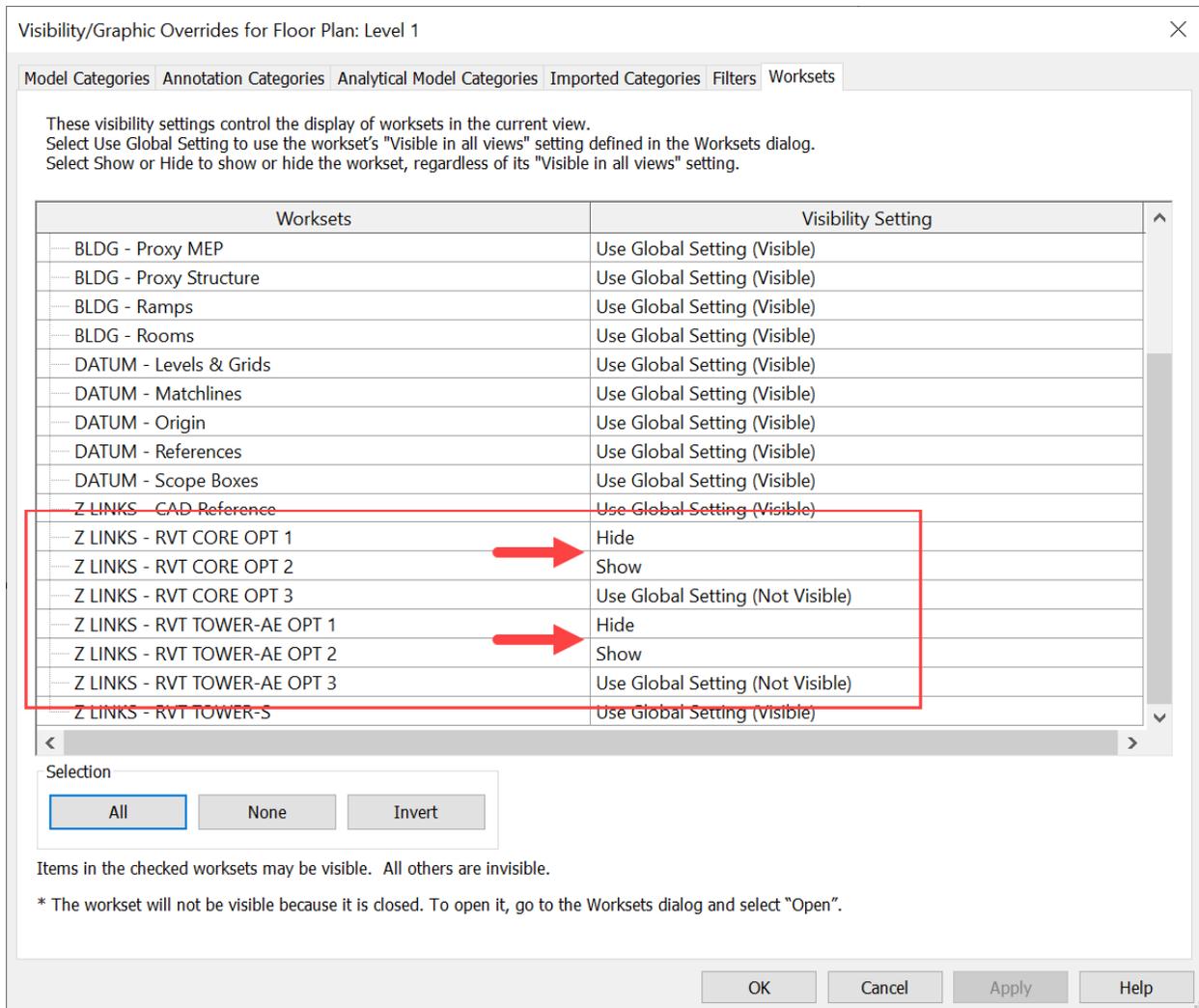


Figure 70: Changing Workset Visibility

This is why this technique is called DIY Design Options, because the set up behaves very closely to how Design Options behave. A “primary” option is selected and by changing the workset visibility properties, only the primary option is visible by default. When a different option needs to be shown, the workset visibility of that specific view is changed. By using this technique, linked files can become design options and all options can be documented from the same file.

View Templates and Alternate Schemes

Once all of this is established and the initial visibility settings made, attention then turns to managing the views in the most efficient way. View templates are normally deployed to make sure

all views of the same type look the same. However, when there are alternate schemes, views will need to have different workset settings to display the correct scheme. If this is the only difference between views, it does not make sense to create multiple view templates, one for each option. If this is done, every time that there is a change unrelated to the options, multiple view templates would need to be updated. The solution is to eliminate the V/G Overrides Worksets from the view template that is assigned to all or most of the views.

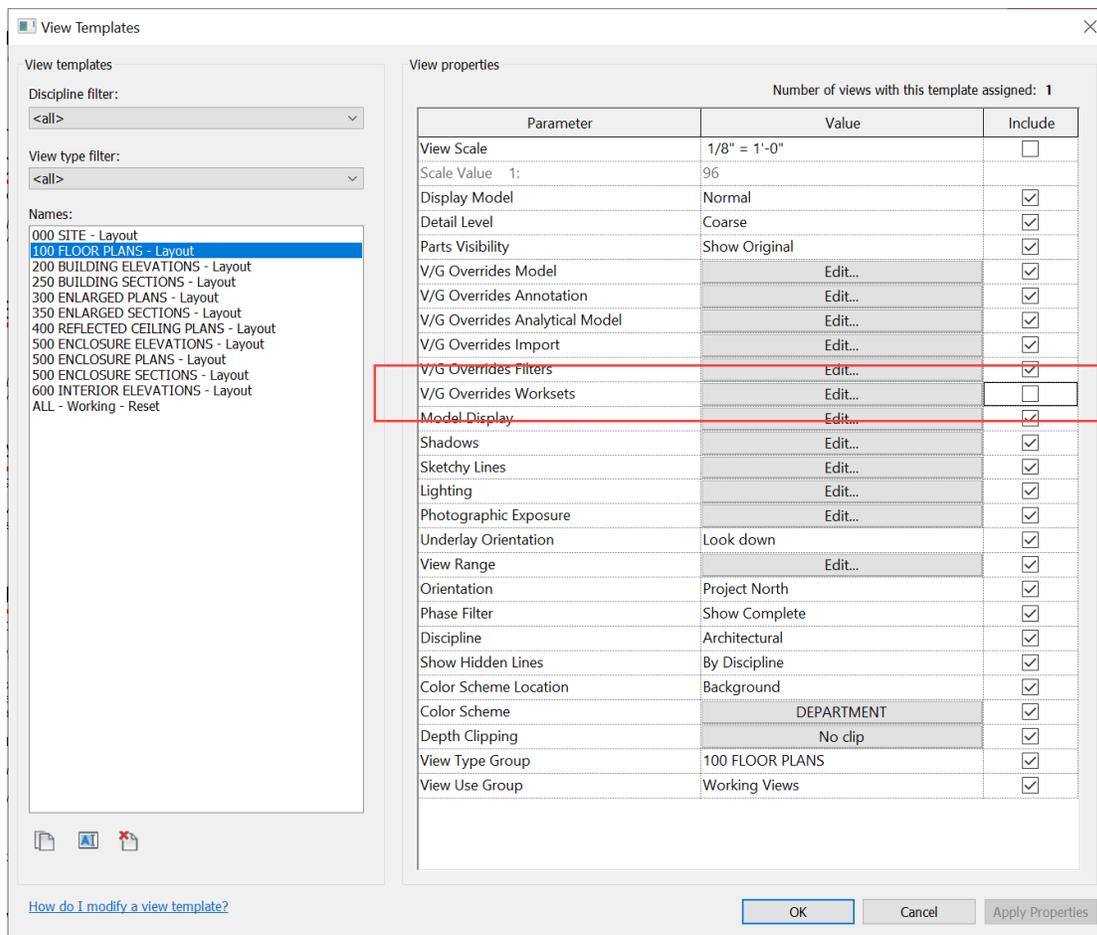
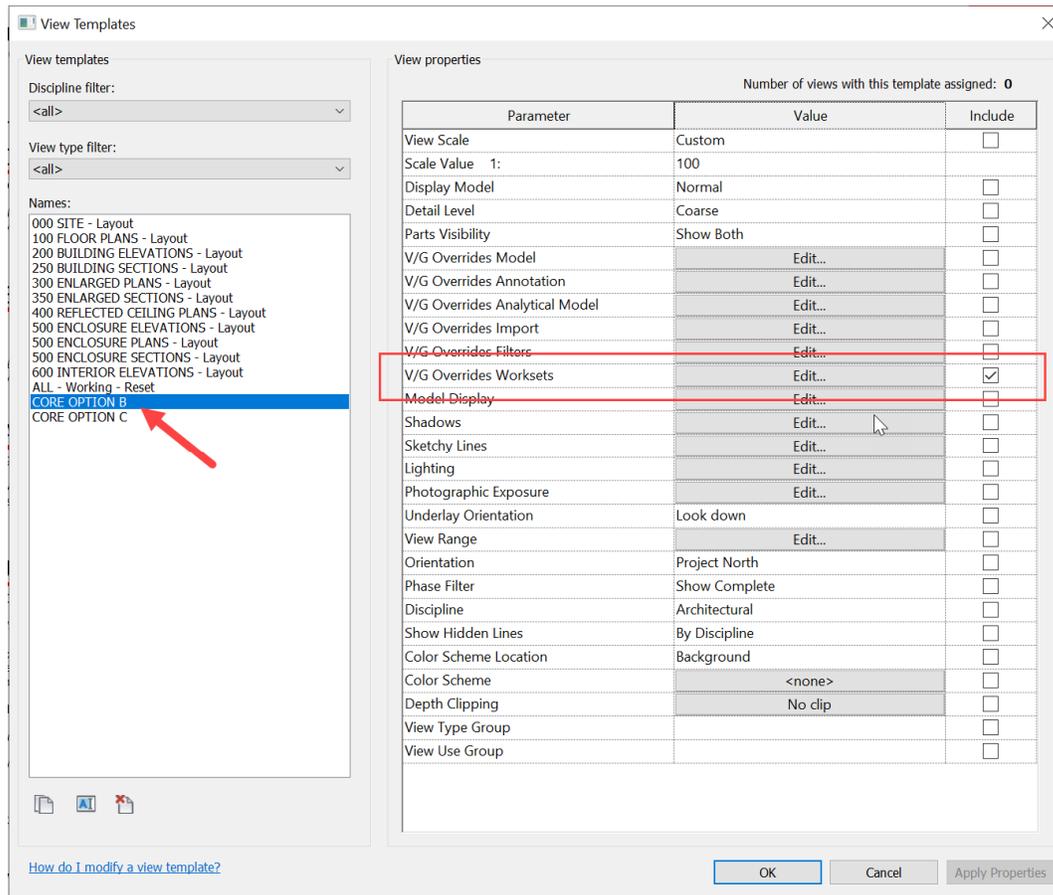


Figure 71: View Template settings

This creates a hole in the assigned view template that can be filled with a second view template that only controls the “V/G Overrides Worksets.”



A “V/G Overrides Worksets” only view template is created for all options except for the “primary” option. For any view that needs to display Option B, the user right clicks on the view in the project browser and selects “Apply Template Properties.” Because the assigned view template is not controlling the worksets, Revit allows the second view template to set the workset overrides in the view. Because this view template is being applied, it must be applied to each view that displays Option B. The same goes for all Option C views. Of course, a Dynamo script could also be created to control the application of view templates to all required views.

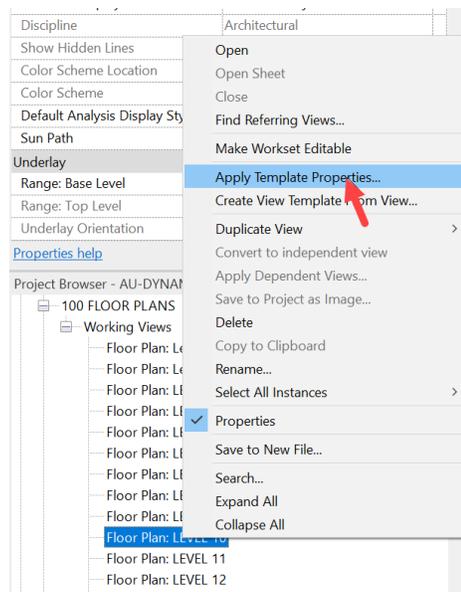
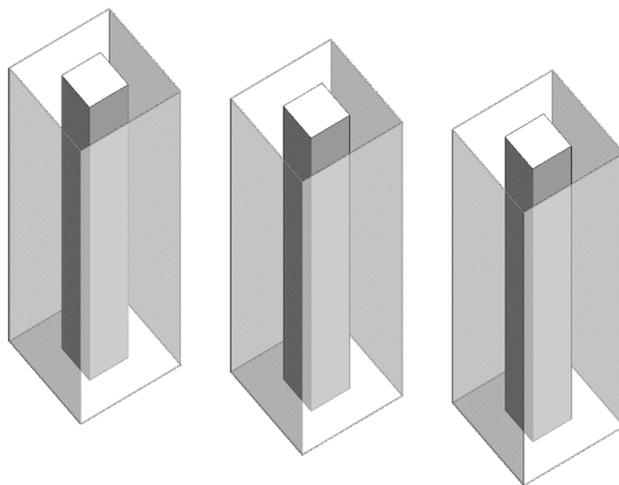


Figure 72: Apply View Templates

This technique works when working with alternate schemes that are only a portion of a project such as cores, exterior walls, or even test-fits, but does it work when there are alternate schemes for the entire building? The answer is typically no. Alternate schemes for entire buildings are handled by a different method, and it's much more straightforward.

Full Alternate Schemes (Everything)



In the diagram to the left, there are three different project schemes. These schemes include alternate versions of everything. In this case, the files are not linked into each other and the documentation for each scheme comes from each scheme file. The three files are treated as three completely independent files.

Figure 73:

Option A

Option B

Option C

When the entire building is an alternate scheme there are no special workset settings or visibility controls. There is simply one file for each alternate scheme and each file is self-contained.

The Winning Scheme

Hopefully, there will come a time in which either the lead designer or the client chooses a single design out of all the alternate schemes created. When this time comes, all the discarded alternate schemes should be removed from the project, and the winning scheme should be reintegrated back into the original file. This is a simple three step process.

1. Remove all linked files that are no longer needed. (The rejected schemes)
2. If the winning scheme should remain as a linked file (e.g. an exterior wall option) no further work is required. If the winning scheme is in a linked file that needs to be re-incorporated back into the original file (e.g. a core option) then select the linked file and click on "Bind Link."
3. When a linked file is bound to another file, as in step 2. The former linked file becomes a group. The group may remain if that is appropriate, but in most cases the group will need

to be ungrouped, so that all the elements that were once part of the linked file become ungrouped native elements in the original file.

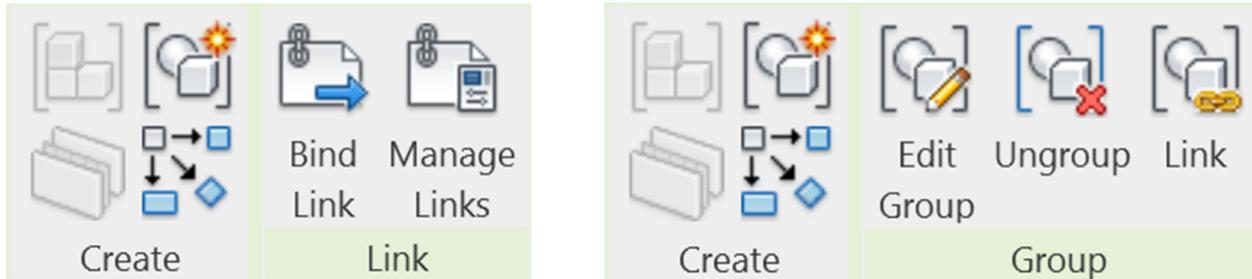


Figure 74: Bind link and Ungroup

Conclusions

Using Revit in Early Design Phases can give teams a head start in later design phases, and can simplify workflows by not having to jump back and forth between software platforms to produce views that Revit can easily produce. In the class and in this handout, it has hopefully been demonstrated that designing in Revit can be done if teams don't get too wrapped up in the details. Teams can mobilize quickly by setting up consistent and reliable files with the help of Dynamo. Teams can control and manage rapidly changing projects with templates, saved groups and Dynamo. Lastly, teams can create and manage multiple and overlapping alternate schemes by creating some "do it yourself" Design Options.

This class has covered tools that all Revit users deal with daily. No new features were covered. This entire class is about looking at the same old Revit tools and seeing them with different eyes. It has been about using them in new and different ways. Because remember,

"What you're supposed to do when you don't like a thing is change it.

If you can't change it, change the way you think about it."

- Maya Angelou