



Incorporating Engineering into Autodesk® Revit® Structure: Project Procedures

Desirée Mackey, PE – Martin/Martin

SE1592 This class demonstrates how to use custom families that incorporate structural calculations within a project environment. In addition, this class shares some project procedures and tools that will enhance the capabilities of Autodesk Revit Structure software for engineers, such as working schedules, calculated values, and conditional formatting. By integrating engineering into custom families and project procedures, a structural engineer can make use of more features of Revit to aid in the engineering design process. This class covers several custom families and procedures and how to incorporate various methods, through conditional formatting and visibility settings, to notify the engineer if any of the components are overstressed or otherwise “not working.” In the companion class, “SE1591 Incorporating Engineering into Autodesk® Revit® Structure: Advanced Families,” the custom families are discussed and demonstrated within the Family Editor.

Learning Objectives

At the end of this class, you will be able to:

- Incorporate engineering into Revit Structure
- Describe custom families that incorporate structural engineering calculations
- Use working schedules and conditional formatting to enhance the capabilities of Revit Structure
- Use custom families in the project environment to enhance the capabilities of Revit Structure

About the Speaker

Desirée (Dezi) received her bachelor's degree from University of California, Davis and her master's degree from Massachusetts Institute of Technology and now is a practicing structural engineer at Martin/Martin in Denver, Colorado. In the past several years she has been a regular Autodesk University speaker and has spoken at Revit Technology Conference USA. In addition, she sits on the committee for RoMBIS (Rocky Mountain Building Information Society – a Denver area users group), the Chair of the Structural Engineers Association of Colorado's BIM committee, and she is currently serving as an AUGI board member and Treasurer. Finally, she also acts as a partner in her husband's BIM consulting company, BD Mackey Consulting.

Email: dmackey@martinmartin.com

Blog: bdmackeyconsulting.com/blog

Twitter: [@RevitGeeksWife](https://twitter.com/RevitGeeksWife)

Introduction

By integrating engineering into custom families and project procedures, a structural engineer can make use of more features of Revit to aid in the engineering design process. This session is a compilation of families and project procedures that incorporate structural engineering and design. This session will present a wide range of possible opportunities for the incorporation of engineering, separated into three categories: families that perform calculations, families that design themselves and project procedures and workflows. This class will discuss specifically how to use these families and procedures within the project environment. In the companion class, "SE1591 Incorporating Engineering into Autodesk® Revit® Structure: Advanced Families," the custom families are discussed and demonstrated within the Family Editor.

Families that Perform Calculations

A useful application of adding engineering into families is to have families perform calculations. The calculations could be for design or engineering purposes, or simply code checks or other verification-type calculations that are meant to report back information to the user. The following families are some examples of this application.

Reinforced Beam Detail Item

This family was originally conceived based on the idea that drafting 2D concrete details was time consuming. Originally, the item was just intended to combine/nest several other detail items into one. However, once all incorporated, it seemed a logical next step to add some engineering, after all, most of the design information was already present.

Within the project, the user places this 2D detail item by using the "Component: Detail Component" command on the Annotate Tab.

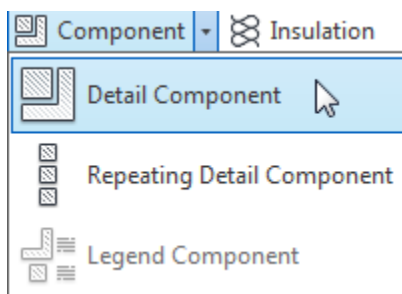


Figure 1: Detail component

The component is placed with a single click, and can be moved and aligned as usual. Once placed, the user simply enters information in the instance properties such as width, depth, concrete cover, reinforcing, concrete strength, etc. Given the values input, the geometry of the detail item updates. In addition, the family takes the information and calculates the flexural capacity of the beam. This value is also visible within the properties.

Properties	
Reinforced Concrete - Rectangular - Section	
Detail Items (1) Edit Type	
Constraints	
b	2' 0"
h	3' 0"
d	2' 7 1/2"
d'	0' 3"
Rebar Set	
Top Bars Visible	<input type="checkbox"/>
Top Bars Diameter #	8
# of Top Bars	4
Bottom Bars Visible	<input checked="" type="checkbox"/>
Bottom Bars Diameter #	7
# of Bottom Bars	4
Tie Visible	<input checked="" type="checkbox"/>
Tie Diameter #	3
Tie Spacing	12"cc
Graphics	
Hook Top Left	<input type="checkbox"/>
Hook Top Right	<input checked="" type="checkbox"/>
Hook Bottom Left	<input type="checkbox"/>
Hook Bottom Right	<input type="checkbox"/>
Text	
Tag Top Bars Dia	
Tag Tie Spcng	12"cc
Tag Tie Dia	3
Tag Text Part 2	bottom bars and
Tag Text Part 1	
Tag Bottom Bars Dia	7
Tag # of Top Bars	
Tag # of Bottom Bars	4
Top Bars Only	<input type="checkbox"/>
Bottom Bars Only	<input checked="" type="checkbox"/>
T&B Equal	<input type="checkbox"/>
T&B Unequal	<input type="checkbox"/>
Structural	
f _y	60.00 ksi
f _c	4.00 ksi
Phi	0.900000
et	0.042518
c	0' 2 19/256"
Dimensions	
Cover Top	0' 11/2"
Cover Bottom	0' 3"
Cover Left	0' 11/2"
Cover Right	0' 11/2"
Top Bars	0' 11/2"
Bottom Bars	0' 11/2"
Left Bars	0' 11/2"
Right Bars	0' 11/2"
Identity Data	
Comments	
Mark	
Structural Analysis	
RebarClearSpaceMin	0' 1"
RebarClearSpace	0' 5 1/8"
PhiMn	330.67 kip-ft
Mn	367.41 kip-ft
As	0.02 SF
As'	0.00 SF
a	0' 1 49/64"
As min	0.02 SF
As and Spcng min Check	<input type="checkbox"/>
Other	
Top Bar As'	0.01 SF
Bottom Bar As	0.00 SF

Figure 2: Concrete component properties

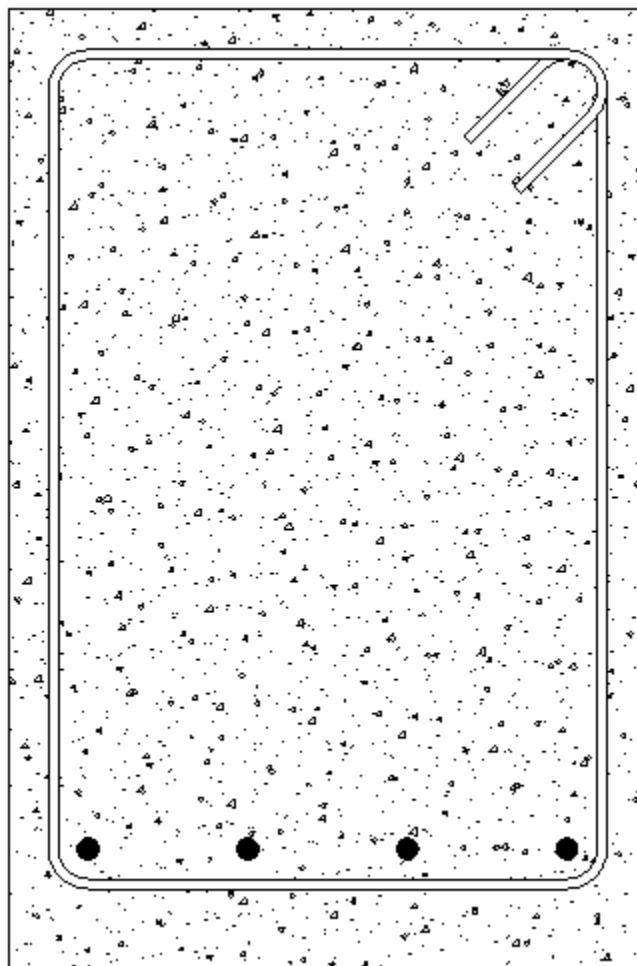


Figure 3: Placed component

Furthermore, the component also checks a few code values, such as minimum reinforcement and minimum clear spacing between reinforcement. If these values are violated, the detail item alerts the user by turning the reinforcing red.

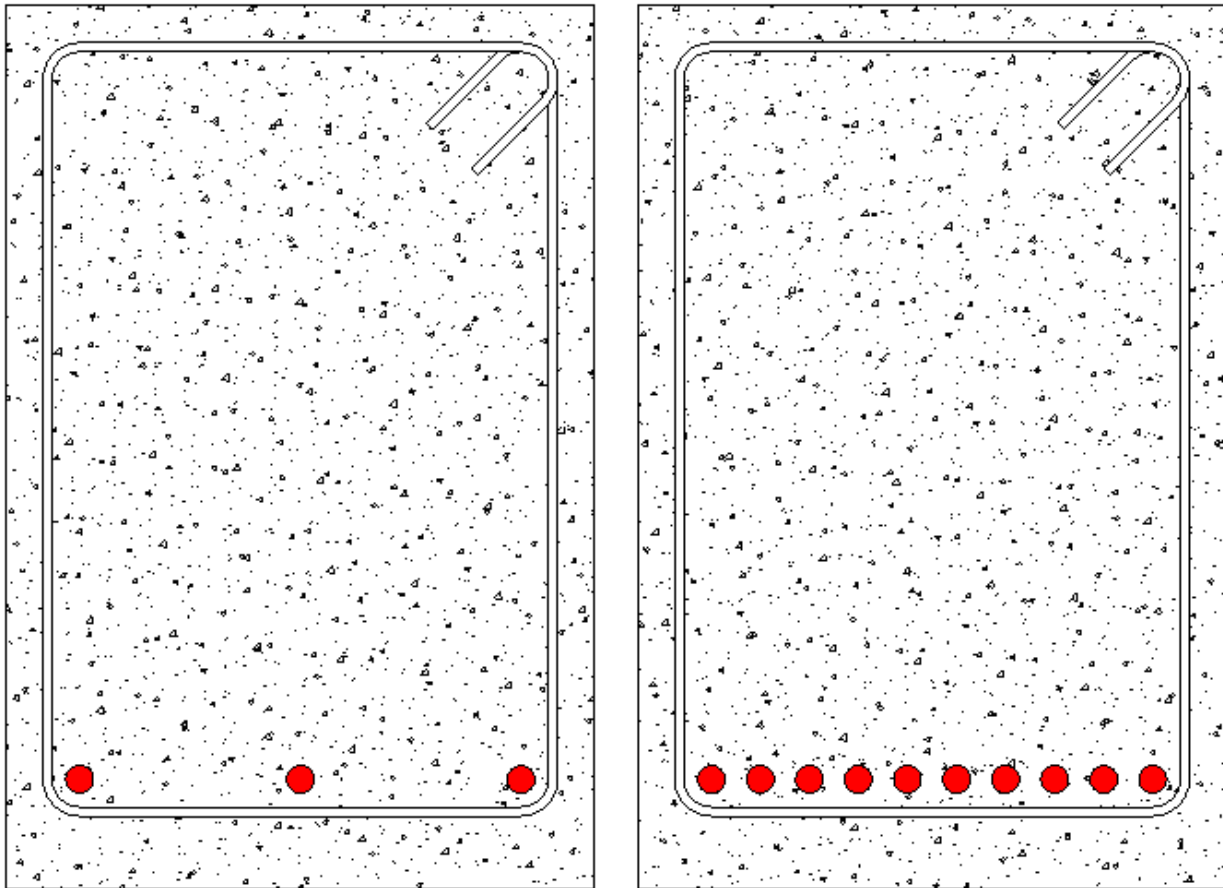


Figure 4: Code checks

Finally, some custom text parameters were added to the family such that a tag could be created to describe everything in the family. There are other ways this could have been achieved – the nested families can be shared and tagged individually – but this seemed to be a useful approach. The tag family that was created to accompany this detail item combines the custom text parameters into one phrase that describes the entire component.

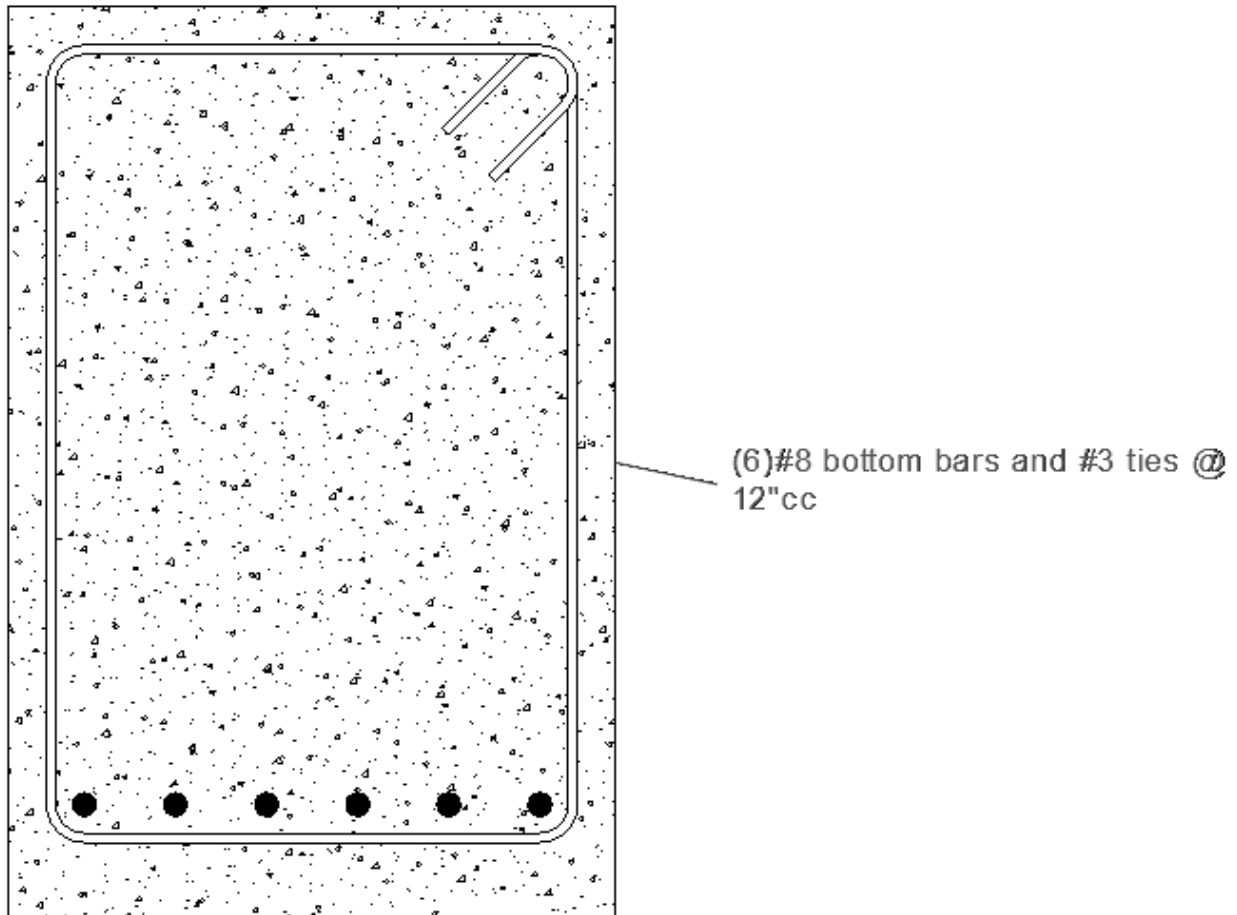


Figure 5: Tagged component

This idea can be expanded to a plethora of other components – a concrete column, for example. This is just one example of how to incorporate some engineering and design information, as well as some basic code checks, into an already useful and robust detail component.

Shear Wall Family

One of the author's favorite families that performs calculations is a component that performs wood shear wall calculations. Over the years, as new features became available within Revit, this component has evolved and has been rebuilt numerous times. The latest version of this family is an adaptive component.

Since this family has been several years in the making, the calculations within it have developed and evolved, and are quite extensive. The calculations use several types of parameters, numerous formulas, logical tests, and even lookup tables, which are newly incorporated into Revit 2014.

The component is placed by four placement points. The user selects the left, right, bottom, and top of a wall, following the segmented shear wall procedure/guidelines. The component then places a symbolic line at the base of the wall that shows the extent of the shear wall. This symbolic line has a parameter that controls its offset, and is also placed on its own subcategory to enable the user to toggle it on and off in various plans. This family can be placed on linked architectural walls, thus eliminating the need to remodel (however if the link is lost, so is the component).

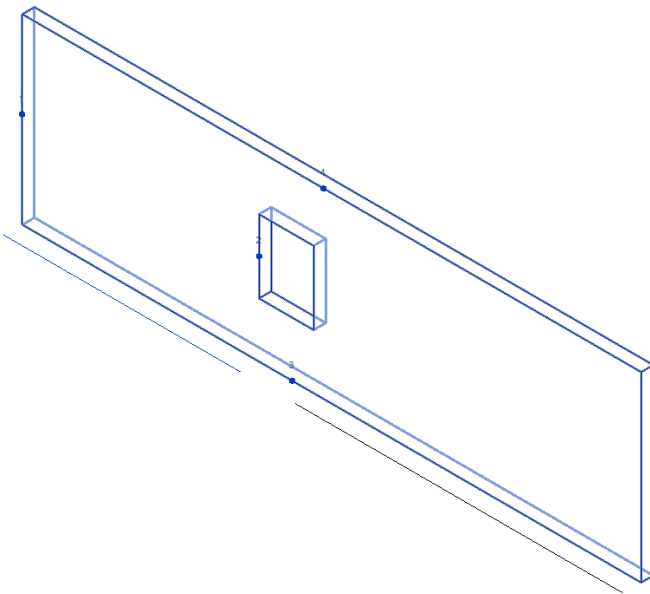


Figure 6: Component placed in project

Once placed, the engineer simply inputs some basic loading and design information into the component (load to the wall, wall weight, desired sheathing type and thickness, desired nail

size, etc.). Once all information has been provided, the component calculates the required shear wall nail spacing, the required anchor bolt spacing, and the required hold downs.

Properties	
Shear Wall	
Structural Connections (1) Edit Type	
Constraints	
Level	Level 1
Elevation	0' 0"
Moves With Nearby Elements	<input type="checkbox"/>
Graphics	
Symbolic Offset	1' 0"
Text	
HDx	No Net OT
Error Message	Inputs okay
Lookup Table Name	Shear Capacities
Structural	
Rebar Cover	Interior (framing, columns) ...
Dimensions	
SP Thickness	0' 0 7/16"
Nail Spacing	0' 6"
Length OT	12' 0"
Length	13' 0"
Height	10' 0"
Volume	
Identity Data	
Comments	
Mark	
Phasing	
Phase Created	New Construction
Phase Demolished	None
Structural Analysis	
Wall Weight	0.0100 ksf
Wall Shear Demand	0.000 kip/ft
Wall Shear Capacity	0.260 kip/ft
Wall Shear Above	0.000 kip/ft
Tension Above	0.00 kip
Net Tension	-0.42 kip
Net Compression	0.70 kip
M Res	8.45 kip-ft
M OT	0.00 kip-ft
Load	0.00 kip
HD Capacity	0.00 kip
Dead Load on Wall	0.000 kip/ft
Compression Above	0.00 kip
SW Type Code	3.000000
Adaptive Component	
Flip	<input type="checkbox"/>
Other	
Wind	<input type="checkbox"/>
Struct 1	<input checked="" type="checkbox"/>
Sheathing	<input type="checkbox"/>
Seismic	<input checked="" type="checkbox"/>
SW Nailing	6
Nails xd	8
AB Spcg	36
6inCapacity	260.000000
4inCapacity	399.000000
3inCapacity	511.000000
2inCapacity	678.000000

Figure 7: Shear wall instance properties

Next, some custom tags were created to be used with the component. These tags show the required nail and anchor bolt spacing, the approximate length of the shear wall segment and the required hold down. However, since all of the parameters in the family are shared, the tags could incorporate any information from the family such as capacities, demand/capacity ratios, etc.

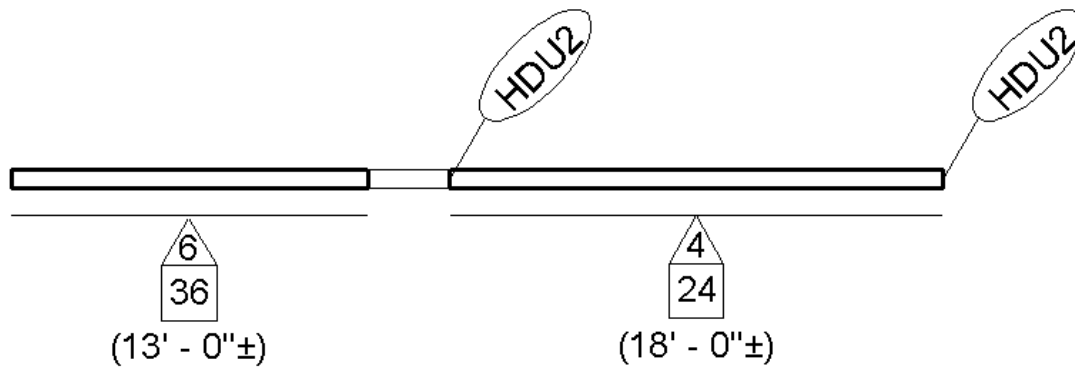


Figure 8: Shear wall component with tags

Since the component is hosted on the wall, and is adaptive, it will update if and when the wall and/or wall openings change, and since the component will update, the design will update as well. Of course if any of the elements are deleted, the component will also be deleted, and the downside is that it will be deleted with no warning.

Finally, since all of the parameters in the family are shared, the components can be scheduled. In addition to becoming a useful view to be added to a sheet, this would allow for editing within the schedule, and potentially could be printed as calculations to submit to the building department, etc.

Shear Wall Schedule																									
Length	Height	Load	Structl	SP	Wall Shear	Wall Shear	Nails	Nail	Wall Shear	AB	MOT	Wall	Dead	M Res	Length	Tension	Net	HD	HD	Compression	Net	Seismic	Wind	Shear	HD
			l	Thickness	About	Demand	sd	Spacing	Capacity	Spcc	T	Weight	Load on		OT	About	Tension	Box	Capacity	About	Compression			D/C	D/C
13' - 0"	10' - 0"	6.00 Kip	Yes	2' - 0 7/16"	0 lb/ft	462 lb/ft	8	0' - 3"	511 lb/ft	16	42.00 Kip-ft	10.0 psf	0 lb/ft	2.45 kip-ft	12' - 0"	1.00 Mp	+0.06 Mp	HD US	5.67 Mp	1.50 Mp	5.70 Mp	Yes	No	0.903206	0.719771
18' - 0"	10' - 0"	6.00 Kip	Yes	2' - 0 7/16"	0 lb/ft	278 lb/ft	8	0' - 4"	368 lb/ft	24	35.00 Kip-ft	10.0 psf	0 lb/ft	16.20 kip-ft	17' - 0"	1.00 Mp	2.49 Mp	HD US	3.28 Mp	1.50 Mp	4.51 Mp	Yes	No	0.696185	0.757056

Figure 9: Shear wall schedule

This component is an excellent example of maximizing what Revit can do – not only does the component provide useful graphics, but it also provides design and calculations. As it has been updated over the years, this component has evolved to incorporate many things, and will likely further evolve and grow in the future.

Families that Design Themselves

It is possible to not only perform calculations within families, it is possible to have those calculations drive sizes, configurations, etc. of families such that they could essentially design themselves.

Pad Footing

The first example of this idea is a pad footing that adjusts its size based on its load demand. To calculate the required footing size, the values required are area load demands (Dead, Live, Snow, etc.), the area that is tributary to the footing, and the allowable soil bearing capacity.

In the project, this custom family is placed just like any other footing. However, once placed, the engineer needs to align and lock four reference planes that help the family calculate the tributary area for the footing. Given the orientation of the footing and the reference planes, the tributary area needs to be a regular rectangle for this to be accurate (but the family could always be modified for custom cases!)

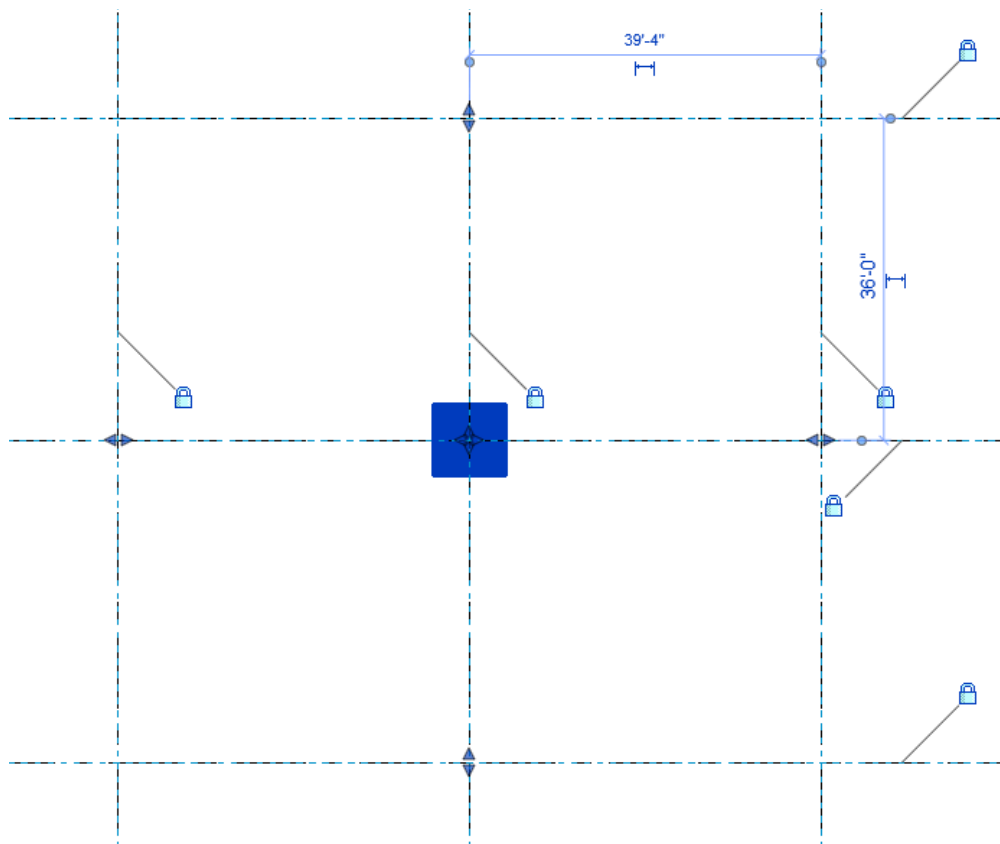


Figure 10: Footing placed in project showing tributary dimensions locked to grids

The engineer also needs to input some basic loading information. Once the family is placed and the loading and tributary area are defined, the family calculates the total load demand, and from the demand, the family calculates the required footing area. Instead of stopping here, this

family then updates its own geometry. One item to note is that since the size of the footing automatically updates, the various types of the family are used for different loading scenarios, configurations, etc., and not for different sizes, which is different than the typical footing.

Also, since the geometry is instance-based, instead of type-based, a custom instance-based tag was created for the footing.

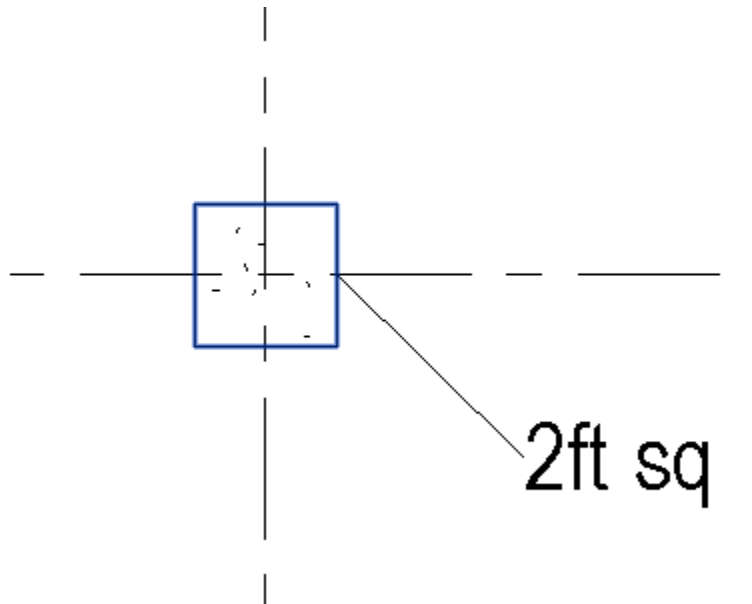


Figure 11: Tagged footing

Once placed in a project, this family allows the basic design of the footings to update as the model evolves.

Structural Column

This structural column is placed and used very similarly to the pad footing discussed in the previous section. The column is placed like any other column, and then the engineer inputs basic loading information and aligns four reference planes for the tributary area.

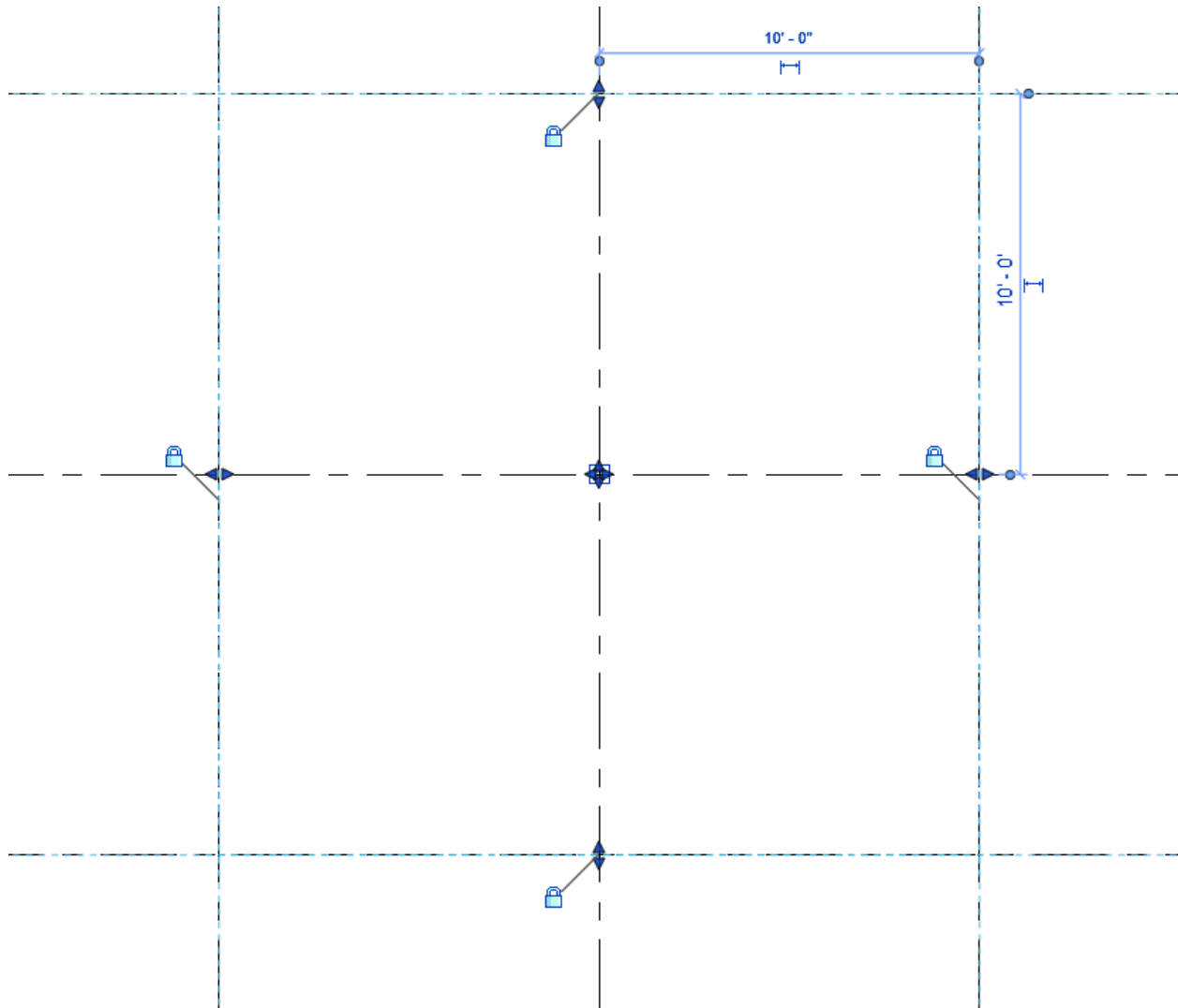


Figure 12: Column placed in project showing tributary dimensions locked to grids

Once placed, and once loads and tributary area have been defined, the column collects other important information automatically (height), designs itself, and updates its geometry. As was true with the footing, the size of the column is instance-based, so the various types of the family are used for different loading scenarios, configurations, etc., and not for different sizes, which is different than the typical column. Also like the footing, the instance-based tag is used to tag the column.

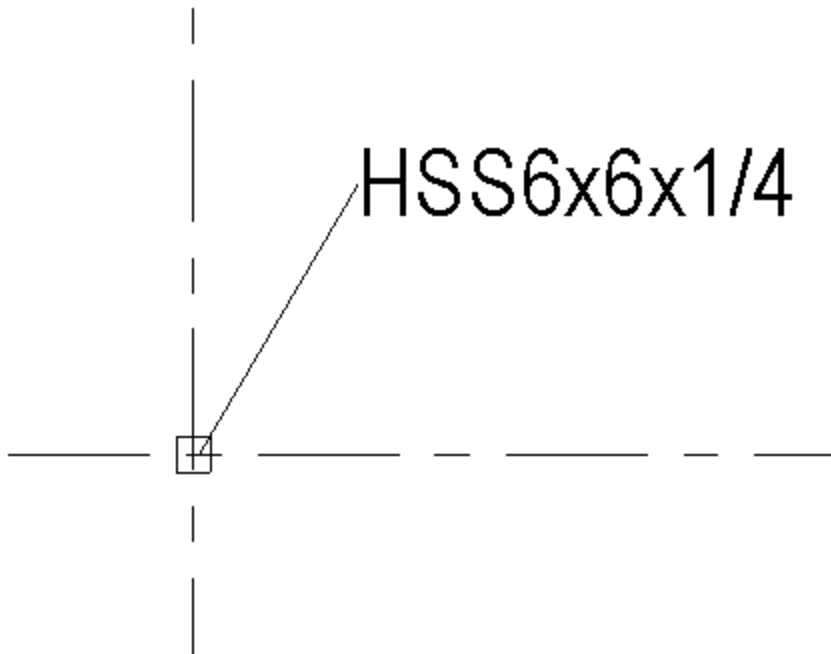


Figure 13: Tagged Column

This idea of inputting minimal information and drawing dimensions and other information out of the project once the family is placed can be applied to any number of other structural elements. Furthermore, these elements could be combined into a project such that nearly every element in an entire building could “design itself”.

Deflecting Double Tee (Bonus Material)

This family is covered in another class, “SE1595: Using Autodesk® Revit® Structure in Investigative Engineering”, but it is another great example of a family that incorporates engineering. This family was created for a specific client and project, so it has a very specific purpose. Given several user-inputted values, including a value for a “rain event”, the family calculates the expected mid-span deflection and updates the geometry of the beam to show the deflection.

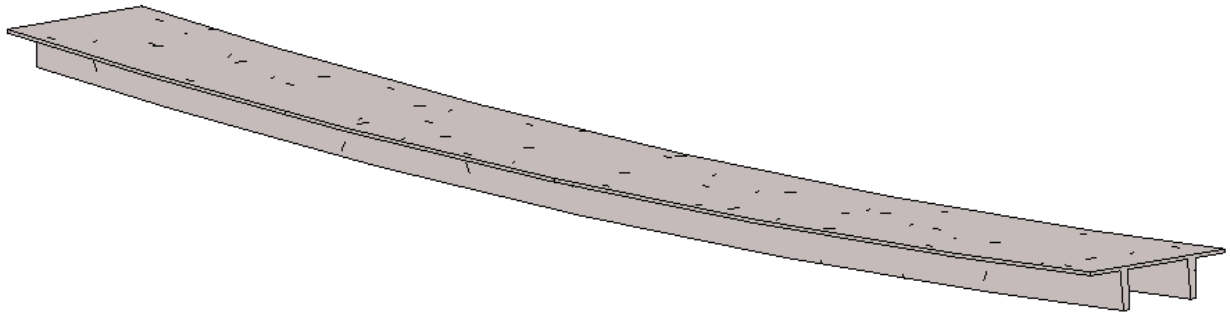


Figure 14: Double tee deflected shape

This, combined with an adaptive component that is intended to show the water on the roof, creates a visual aid for exactly what occurs on the roof when it rains.

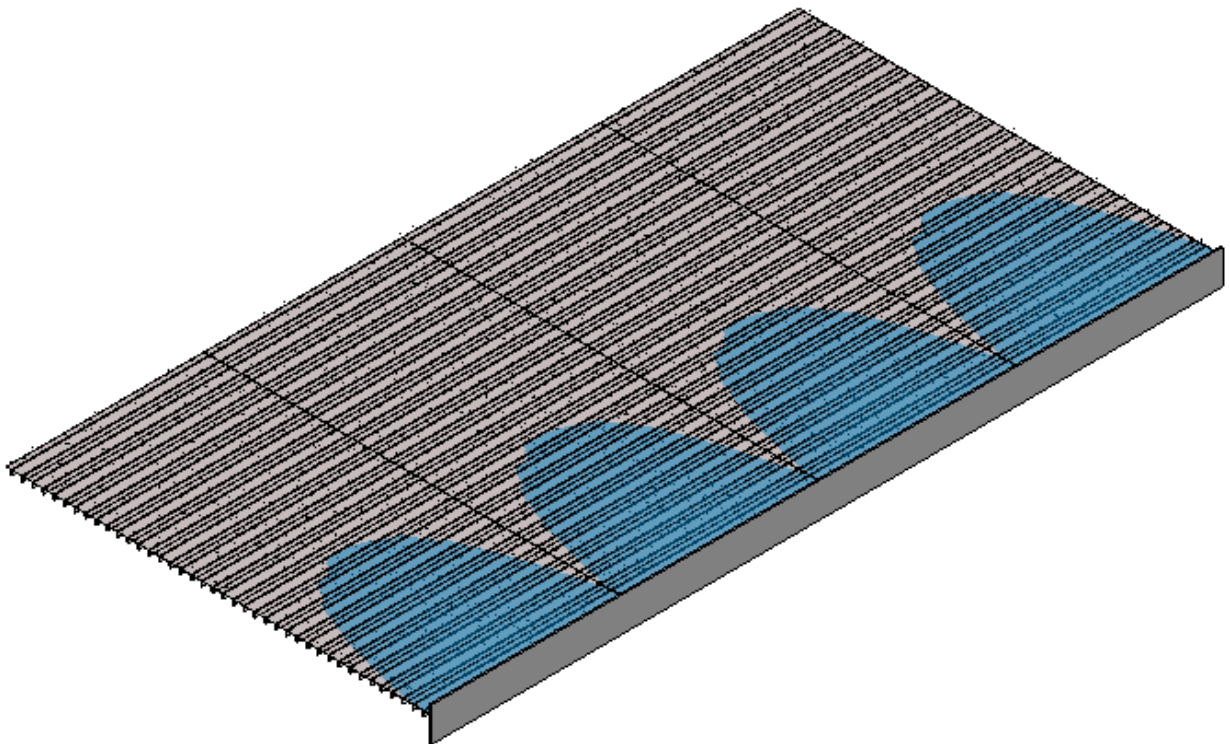


Figure 15: Rain on roof

Once loaded into the project, the beams are given a couple of inputs: row number (to help determine each beam’s distance from the wall) and initial height of water. Each rain event can then be scheduled and summarized to show values such as final weight of water, final deflection, etc. Also within the schedule, calculated values can be used to determine the final height of water at the parapet wall. This was achieved by subtracting the volume of water that would “sink” into each deflected shape from the initial volume of water, summing these values for the entire roof, and then using geometry to determine the final height of water.

Finally, all of the parameters can be scheduled, and using calculated values (which will be discussed in the Project Procedures section), additional information, such as total water volume on the roof can be calculated and displayed. The figure below shows an example of what was shown on the schedules.

Water Volume Summary 7" Rain Event					
Row Number	Count	Sag Initial	Volume of Water	Weight of Water	Sag Final
1	4	0' - 0 1/2"	1416.67 CF	90.67 kip	0' - 8"
2	4	0' - 0 1/2"	1316.67 CF	84.27 kip	0' - 7 1/8"
3	4	0' - 0 1/2"	1216.67 CF	77.87 kip	0' - 6 1/8"
4	4	0' - 0 1/2"	1116.67 CF	71.47 kip	0' - 3 7/8"
5	4	0' - 0 1/2"	1016.67 CF	65.07 kip	0' - 3 5/8"
6	4	0' - 0 1/2"	916.67 CF	58.67 kip	0' - 3 3/8"
7	4	0' - 0 1/2"	816.67 CF	52.27 kip	0' - 3 1/8"
8	4	0' - 0 1/2"	716.67 CF	45.87 kip	0' - 2 7/8"
9	4	0' - 0 1/2"	616.67 CF	39.47 kip	0' - 2 5/8"
10	4	0' - 0 1/2"	516.67 CF	33.07 kip	0' - 2 3/8"
11	4	0' - 0 1/2"	416.67 CF	26.67 kip	0' - 2"
12	4	0' - 0 1/2"	316.67 CF	20.27 kip	0' - 1 5/8"
13	4	0' - 0 1/2"	216.67 CF	13.87 kip	0' - 1 1/4"
14	4	0' - 0 1/2"	116.67 CF	7.47 kip	0' - 0 7/8"
15	4	0' - 0 1/2"	0.00 CF	0.00 kip	0' - 0 1/2"
16	4	0' - 0 1/2"	0.00 CF	0.00 kip	0' - 0 1/2"
17	4	0' - 0 1/2"	0.00 CF	0.00 kip	0' - 0 1/2"
18	4	0' - 0 1/2"	0.00 CF	0.00 kip	0' - 0 1/2"
19	4	0' - 0 1/2"	0.00 CF	0.00 kip	0' - 0 1/2"
20	4	0' - 0 1/2"	0.00 CF	0.00 kip	0' - 0 1/2"
			10733.33 CF	686.93 kip	

Figure 16: Rain event schedule

While the application of this family is rather specific, the idea behind what was completed could apply to any number of situations. Combining engineering information with Revit with this approach provides a way to achieve not only a visual representation, but also a way to simplify and compile engineering and calculations that may otherwise be rather repetitive and iterative.

Project Procedures and Workflows

Top of Footing Elevation in a Footing Tag

A great example of a working schedule that can be leveraged by structural engineers is one that allows the engineer to include the top of footing elevation in the structural footing tag. Out-of-the-box, only the bottom of footing elevation is available, but engineers often prefer to tag the top elevation. This working schedule, along with a custom tag family and a few slight modifications to the out-of-the-box footing family, allows engineers to do just that.

The basic premise of this procedure is to utilize a working schedule, calculated values, and conditional formatting to create a value that can be tagged. Since the footing elevation is only available once the footing is placed within the project, the top of footing elevation will be calculated in the schedule. Since calculated values cannot be tagged, they will be manually copied such that they appear in the tag, and then conditional formatting and logical tests allow for a way to verify that all tagged values are correct.

Within a project, all foundations should be scheduled, utilizing filters to remove any unwanted foundation elements. Also, the “Datum” value can be quickly set for all foundations by utilizing sorting and grouping and editing the value directly within the schedule. The user should add a calculated value, “TO Ftg Calc”, with the formula “Elevation at Bottom + Thickness + Datum”.

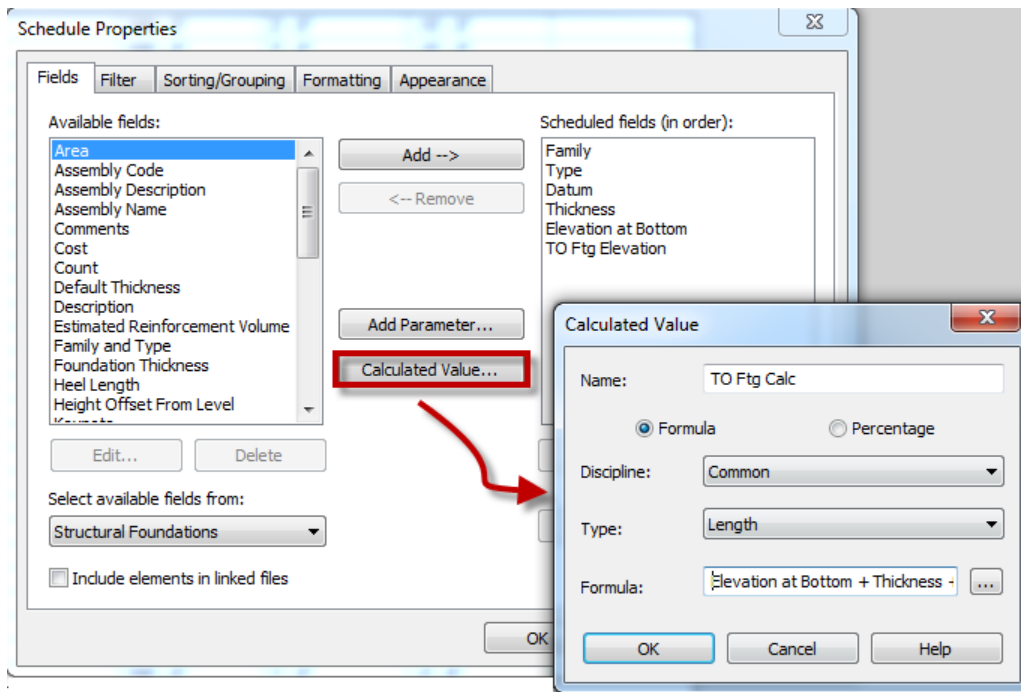


Figure 17: Calculated value to determine top of footing elevation

Another calculated value, called “Check”, needs to be added, and its formula is “TO Ftg Calc - TO Ftg Elevation”.

Next, in the Formatting Tab, conditional formatting should be added to the “TO Ftg Elevation” field such that that it would be highlighted red if the “Check” field is not equal to zero. Note that if the “Show conditional format on sheets” box is checked, the red fill will print.

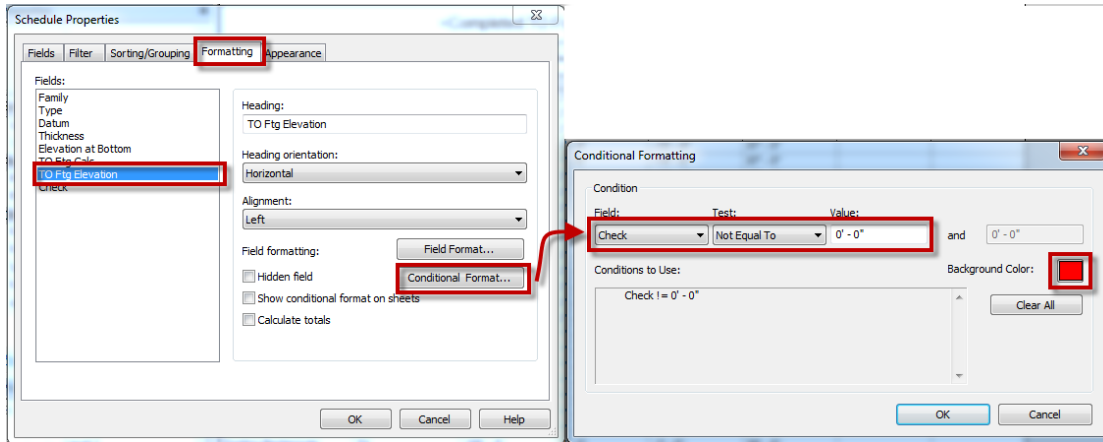


Figure 18: Conditional Formatting to check that values match

Next, again utilizing sorting and grouping, the “TO Ftg Elevation” fields should manually be entered to match the calculated value that is shown in the “TO Ftg Calc” field. Once this field is populated, the tags will show the elevation. If a value is entered improperly, or if the footing moves such that the elevation is no longer accurate, the incorrect value will highlight in the schedule, prompting the user to edit the value.

	F	G	
otto	TO Ftg Calc	TO Ftg Elevation	
	87' - 0"	87' - 0"	0' - 1
	87' - 0"	87' - 0"	0' - 1
	87' - 0"	87' - 0"	0' - 1
	87' - 0"	87' - 0"	0' - 1
	87' - 0"	87' - 0"	0' - 1
	87' - 0"	87' - 0"	0' - 1
	87' - 0"	87' - 0"	0' - 1
	87' - 0"	87' - 0"	0' - 1
	91' - 0"	91' - 0"	0' - 1
	91' - 0"	91' - 0"	0' - 1
	93' - 6"	87' - 0"	6' - 4
	93' - 6"	93' - 6"	0' - 1

Figure 19: Conditional formatting alerting the user that a value is incorrect

While this workaround does include some manual input, it also provides an automated way to maintain accuracy. Also, note that this schedule could be something included in the project template, so the user would not have to recreate it for every project.

Slender Column

While this family is also an example of a family that performs calculations, it is also an example of utilizing working schedules, conditional formatting, as well as formatting and view filters within the project to show how a custom family can be utilized within the project environment. The family utilizes custom parameters and calculations such that, once placed in a project, the column calculates its slenderness ratio and its axial compressive capacity. In this example, working schedules with conditional formatting and view filters can be utilized to highlight columns that have an unacceptable slenderness ratio.

The column is modeled just like any other, and then the user should create a column schedule.

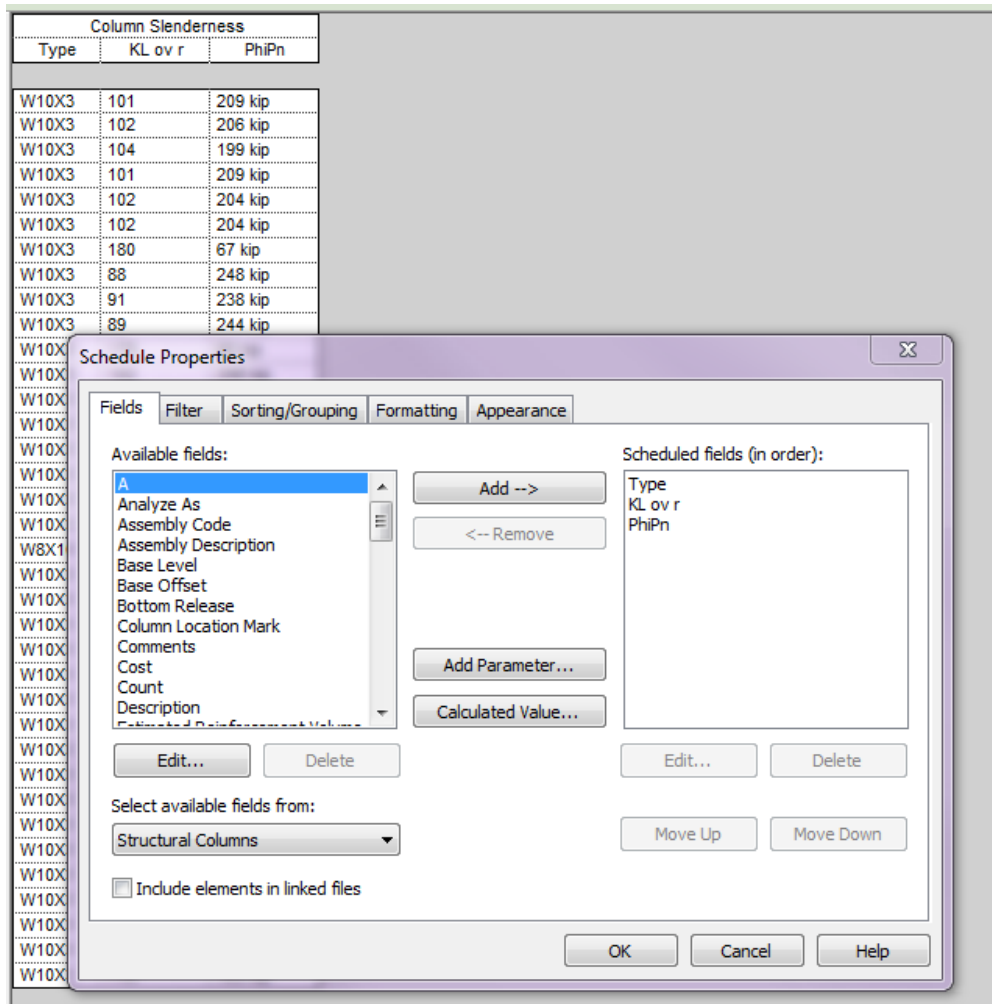


Figure 20: Column schedule

Similarly to the top of footing example, conditional formatting should be added to the “KL ov r” field such that the field is highlighted yellow if the value is unacceptable.

Column Slenderness		
Type	KL ov r	PhiPn
W10X3	101	209 kip
W10X3	102	206 kip
W10X3	104	199 kip
W10X3	101	209 kip
W10X3	102	204 kip
W10X3	102	204 kip
W10X3	180	67 kip
W10X3	88	248 kip
W10X3	91	238 kip
W10X3	89	244 kip
W10X3	179	68 kip
W10X3	102	206 kip
W10X3	90	243 kip
W10X3	181	67 kip
W10X3	213	48 kip
W10X3	98	215 kip
W10X3	97	220 kip
W10X3	109	182 kip
W8X10	256	10 kip
W10X3	109	182 kip
W10X3	91	238 kip
W10X3	91	238 kip
W10X3	91	238 kip
W10X3	91	238 kip
W10X3	91	238 kip
W10X3	91	238 kip
W10X3	91	238 kip
W10X3	91	238 kip
W10X3	91	238 kip
W10X3	91	238 kip
W10X3	91	238 kip
W10X3	91	238 kip
W10X3	116	163 kip
W10X3	116	163 kip
W10X3	116	163 kip
W10X3	116	163 kip
W10X3	116	163 kip

Figure 21: Column schedule with conditional formatting

In addition to utilizing a working schedule, the user could also use view filters and graphic overrides to convey the same information. A view filter called “Column Slenderness” should be created to filter the columns with slenderness ratios that exceed 200.

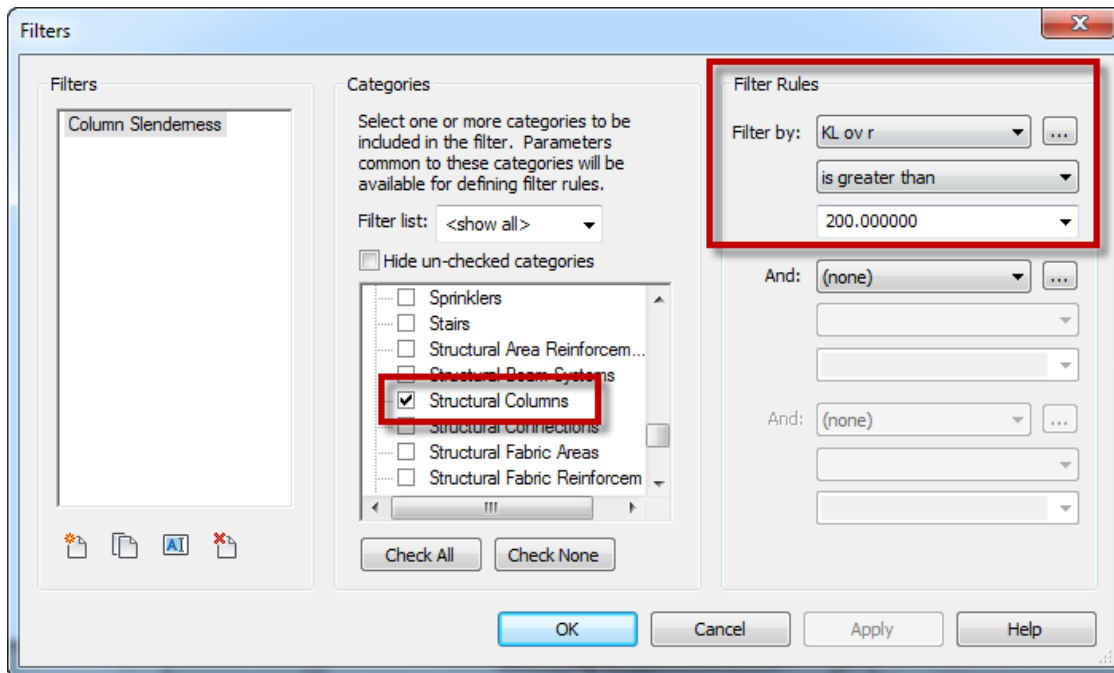


Figure 22: View filter

Then, within Visibility Graphics, the view filter should be added, and overrides should be set to highlight the unsatisfactory columns yellow.

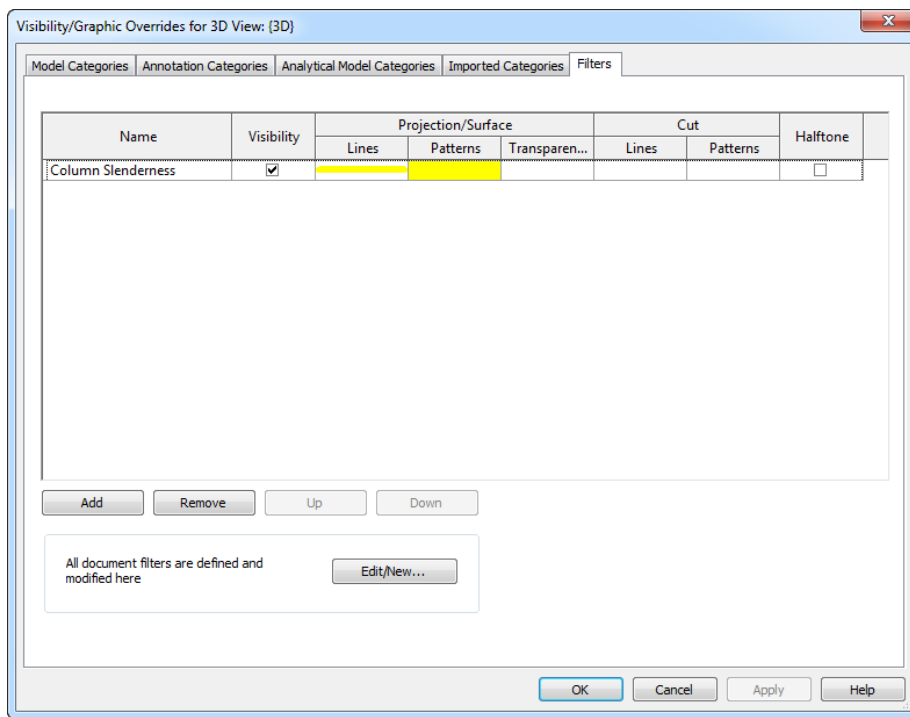


Figure 23: Graphic overrides

Once the view filters and graphic overrides are set, the columns can be easily seen in the view. The combination of the schedules and the filters can be used to edit the unsatisfactory columns such that if edited in the schedule, the view would also show that the columns have been fixed.



Figure 24: 3D view of project with filter and overrides applied

While the slenderness ratio is a relatively basic piece of information (structurally speaking), the same process could be employed to display any other sort of engineering information as well. With this particular family, since the column family also reports each column's capacity, a similar strategy could be implemented to schedule and display demand versus capacity information. While it would likely only be useful in early project stages, utilizing views with these sorts of conditional formatting already established could aid in early sizing to the extent that nearly anyone could choose somewhat appropriate sizes simply by following the formatting indicators. Also, as mentioned in the top of footing example, these schedules, filters and views could be incorporated into the project template such that the user would not need to go through many steps before being able to begin utilizing the views and schedules to view and edit information and elements in the model.

Markup Symbol (Bonus Material)

This final example is simply a custom annotation family that can be utilized for an internal (or possibly even external) workflow process. This custom family is a symbol that can be placed in various views throughout a project, and then can be scheduled to aid in communication among a project team.

To utilize this process, a user would simply place the symbol, selecting the appropriate type – which in this case is based on who generates the comments, and to whom the comment is directed. The various types are represented by different colors. The user would add leaders as needed and fill out the custom parameters to convey whatever information is desired.

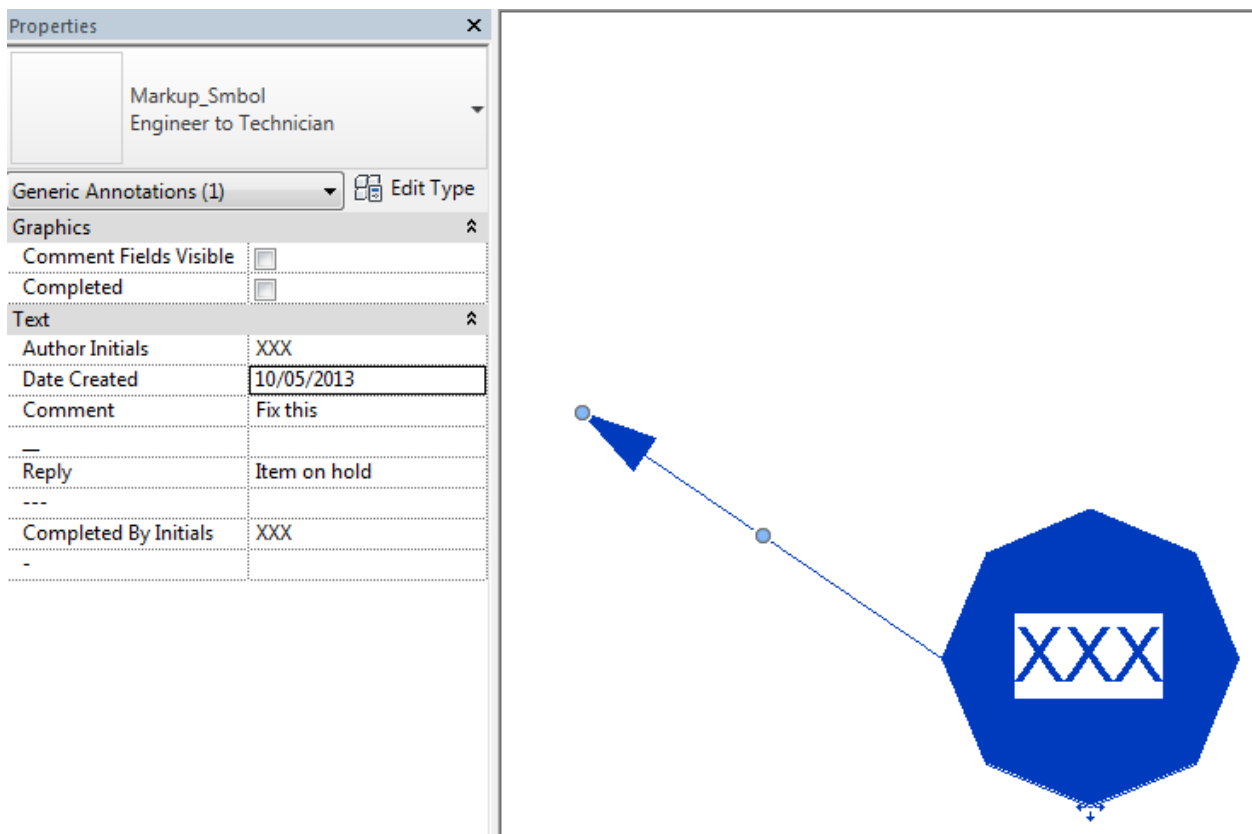


Figure 25: Mark-up symbol

Since it would not be reasonable to expect another user to “hunt” for each of these symbols, they should be scheduled to consolidate all the mark ups in one location. From here, all the users can read through, comment on, address, and complete tasks based on the mark up comments.

<Mark Up Schedule>						
A	B	C	D	E	F	G
Author Initials	Date Created	Comment	Reply	Count	Completed	Completed By Initials
Engineer to Engineer						
RJL	12/12/13	Check capacity of footing for uplift		1	<input type="checkbox"/>	
DPM	8/22/13	Does reinforcement meet min steel?	yes	1	<input checked="" type="checkbox"/>	JNF
Engineer to Technician						
DPM	10/05/13	Change column to extend 2' above roof	Item on hold	1	<input type="checkbox"/>	BEM
Technician to Engineer						
BEM	01/06/13	What weld size?		1	<input type="checkbox"/>	
BEM	8/22/13	Does not match detail		1	<input type="checkbox"/>	
BEM	10/05/13	Should this beam be at the same elevation as the one on grid B?	Item on hold	1	<input type="checkbox"/>	DPM

Figure 26: Mark-up schedule

Furthermore, by using the “Highlight in Model” command on the Modify Schedule/Quantities Tab, a user can easily navigate between the markup schedule and the locations where the symbols exist/where the task is to be completed.

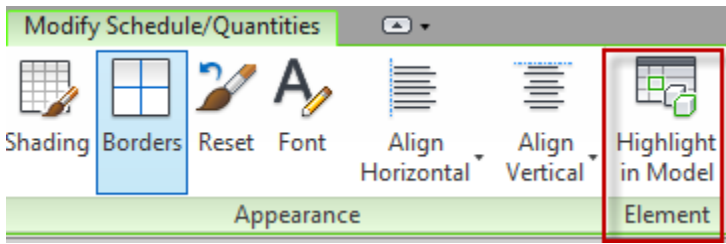


Figure 27: Highlight in model

In addition to the schedule, an instance parameter is available in the symbol that allows the comments to be visible wherever the symbol is placed, so the user would not need to use the schedule if he/she did not want. Also, this would allow very long comments to be more easily viewed (as opposed to looking at them in the instance properties.)

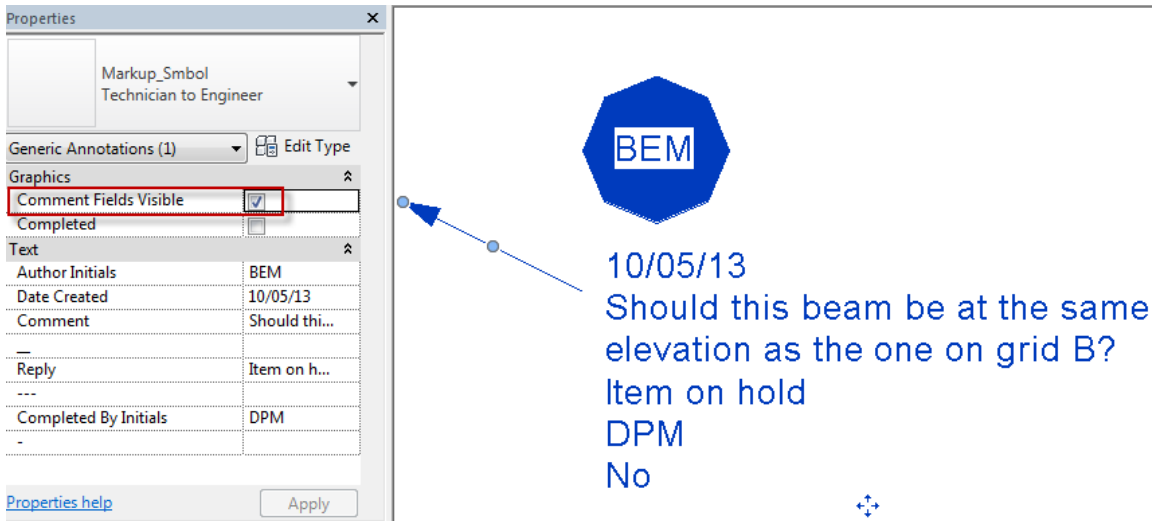


Figure 28: Mark up symbol with visible comments

Finally, once a comment has been completed, checking the “Completed” parameter turns the symbol yellow, indicating it has been addressed. To keep these symbols from printing, either view filters, or simply deleting them would work.

This particular workflow family is something the author created many years ago, but it seems that these simple items are the ones that have the largest effect on efficiency, so it seemed relevant to share in this section.

Conclusion

This session shared a few possible applications, but many various custom families, tools, and methods to incorporate engineering and design into Revit projects. It is this author’s opinion that the potential for such integration is vast and ever-growing. In the companion class, "SE1591 Incorporating Engineering into Autodesk® Revit® Structure: Advanced Families," the custom families are discussed and explained.