

AS-122882

## Creating Intelligent Details in Revit

Brian Mackey

BD Mackey Consulting

### Learning Objectives

- See examples of intelligent details
- Learn how to create the detail components
- Learn how to gain flexibility in nesting detail components
- Learn how to notate the details intelligently

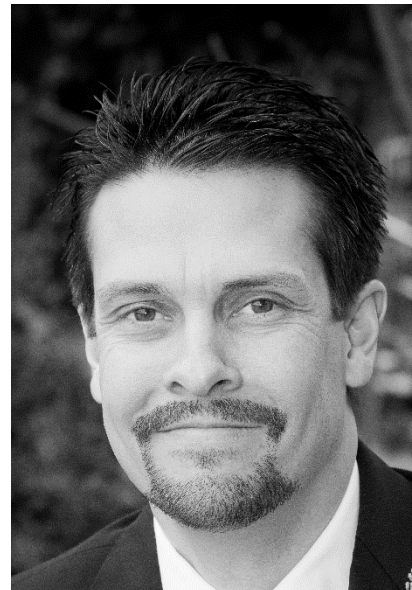
### Description

Everyone knows that Revit software is a Building Information Modeling (BIM) program, and that it has intelligence when modeling. However, this intelligence is sometimes lost when creating details. In this session, we'll show how to create intelligent details using detail components and nested detail components. In doing this, we'll be able to keynote these elements to maintain a consistent noting procedure. If keynoting isn't your choice, we'll also cover how to tag these elements so that there's consistent notation of the details on every project.

### Speaker

Known as "The Revit Geek", Brian has spent more than 25 years in the industry, more than 10 of which have been focused on Revit. Over a decade of working with Architects and Engineers to advance BIM in their companies, Brian started his BIM consulting company in 2011 to focus on custom high-level training/mentoring. Brian has clients all over the US/Canada that generally tolerate his sarcastic nature in exchange for his wide breadth of BIM knowledge. Brian showcases his love of talking about Revit, or maybe just his love talking, in a monthly light-hearted, occasionally irreverent, free Q&A webcast, Revit Radio.

Brian is a regular speaker at many conferences, including Autodesk University, BIM Workshops and BiLT, where he has been awarded top speaker several times. Brian and his wife, who met at AU, welcomed Vienna in 2013, for a total of three daughters and in 2016 welcomed a son Paxton.

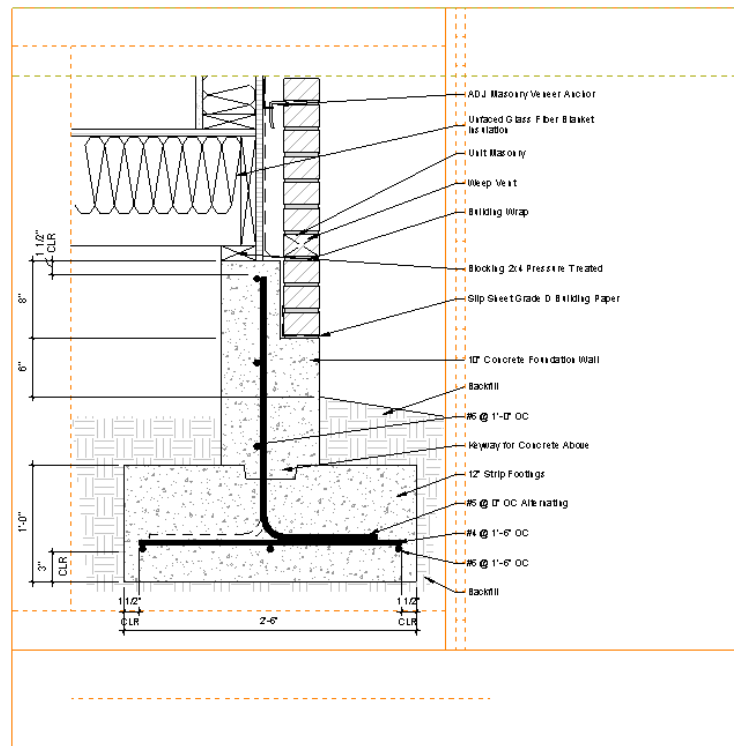


## Creating the Detail

There are different ways to create details: Model everything and only notate the model, use the model as a place to start and then embellish by adding detail items to the view, create standard details without any reference to the model. It is unreasonable to expect to model every element that needs to appear in details. As an example it is difficult to model flashing and waterproofing, or welds and washers. This then necessitates the use of 2D information as part of the set of construction documents. This session will cover how to use 2D Detail Items to best take advantage of their intelligence. These principles apply to both drafting views and detail views.

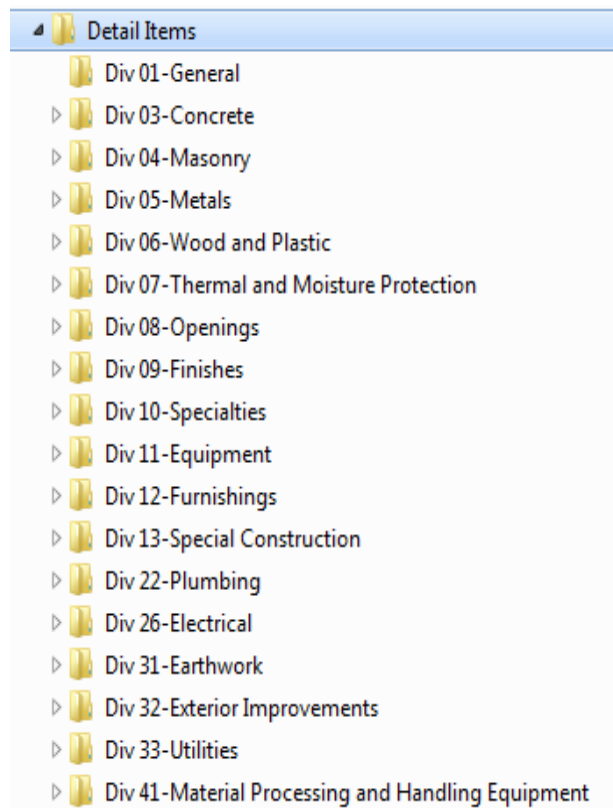
## Detailing Best Practices

If the detail is to be 2D yet still remain intelligent then a rule of thumb is to **not** use any lines or text in the detail, instead use detail items (formerly detail components) families, along with tags (either keynote or component tags). This will accomplish a few things: it will allow for consistency in appearance, information, and spelling, while limiting liability due to the incorrect information noted in a detail. It is also a good rule to use a guide family to assist in alignments and the boundary of the detail.



## OOTB Detail Item Families

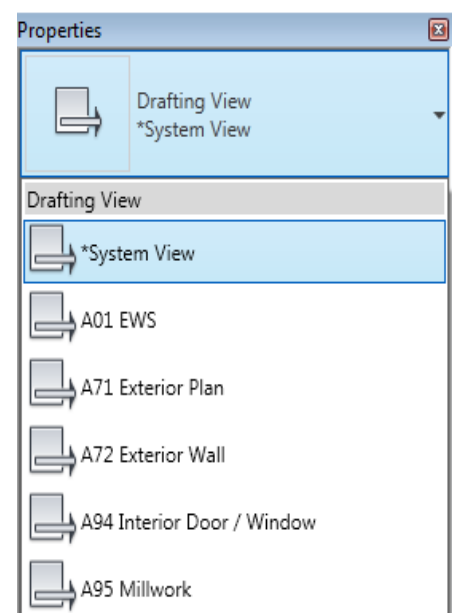
A good starting place is with the Autodesk Revit out of the box (OOTB) families. Autodesk Revit 2018 ships with 665 detail item families. Not everything needed will be in there and more intelligence will likely need to be added to some families, but it is a good start. One complaint about the OOTB detail item families is the line weights aren't consistent with the office standards. If this is the case each family doesn't have to be edited. Instead the object styles can be changed in the project (or project template), or view templates can be created to accomplish the desired line weights. If keynoting is utilized then all of the OOTB detail item families are aligned with the OOTB keynote file. If element tagging is used, then these families will have to be edited to include the desired information. Despite the method chosen, a new keynote or detail item tag will probably need to be created since the OOTB ones might not match what is needed.



## Starting the Detail View

Since this session will discuss 2D details, the place to start is a Drafting View. A good practice is to create different view types for drafting views to help sort and keep track of details in the project.

Once the view is created, load a guide family. Create the guide family from a generic annotation family so it will scale properly and provide a way to judge the available space for a detail. This family can be set to adapt to different layouts and sheet sizes such that the available drafting area is always accurate. Reference planes can snap to the insertion point of the guide family which will then allow for snapping the view into place when placing it on a sheet.

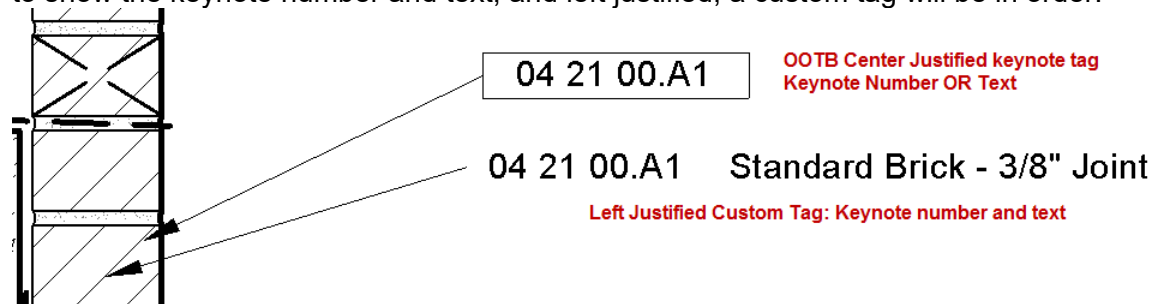


## Creating the Detail

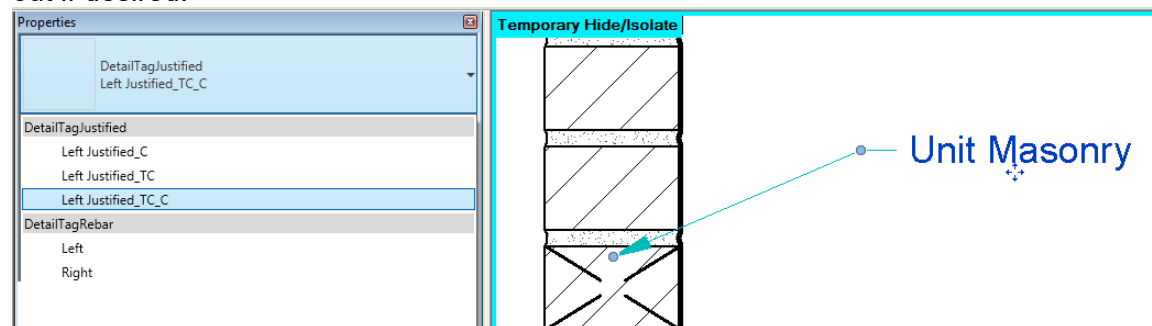
Once the view has been created, the graphics are all created by placing detail items. These detail items can be anything from advanced families that are used on a daily basis, to non-parametric families that are used on a project specific basis. The advantage of using intelligent detail items is the future capability to tag these elements; even simple elements that look like lines should be a family for this reason. If you are using repeating detail components, there is a limitation on tagging them, a single item will have to be placed in a detail for tagging purposes, or think about creating a line based detail family that utilizes an array to repeat the nested detail item.

## Annotating the Detail

Despite the chosen approach, keynoting or tagging, it is necessary to create a custom tag that will meet company standards. That also is more flexible than those available OOTB. So if the company standards include consistent left or right justified text, dictate the use of a box around the tag (or not), the need for a custom tag will become inevitable. In addition as families get more advanced, multiple tags might be required to tag different shared parameters, rather than just tagging Type Comments, etc. The OOTB keynote family only allows for the keynote number OR the keynote text, not both together, plus, it is center justified. So if the standard is to show the keynote number and text, and left justified, a custom tag will be in order.



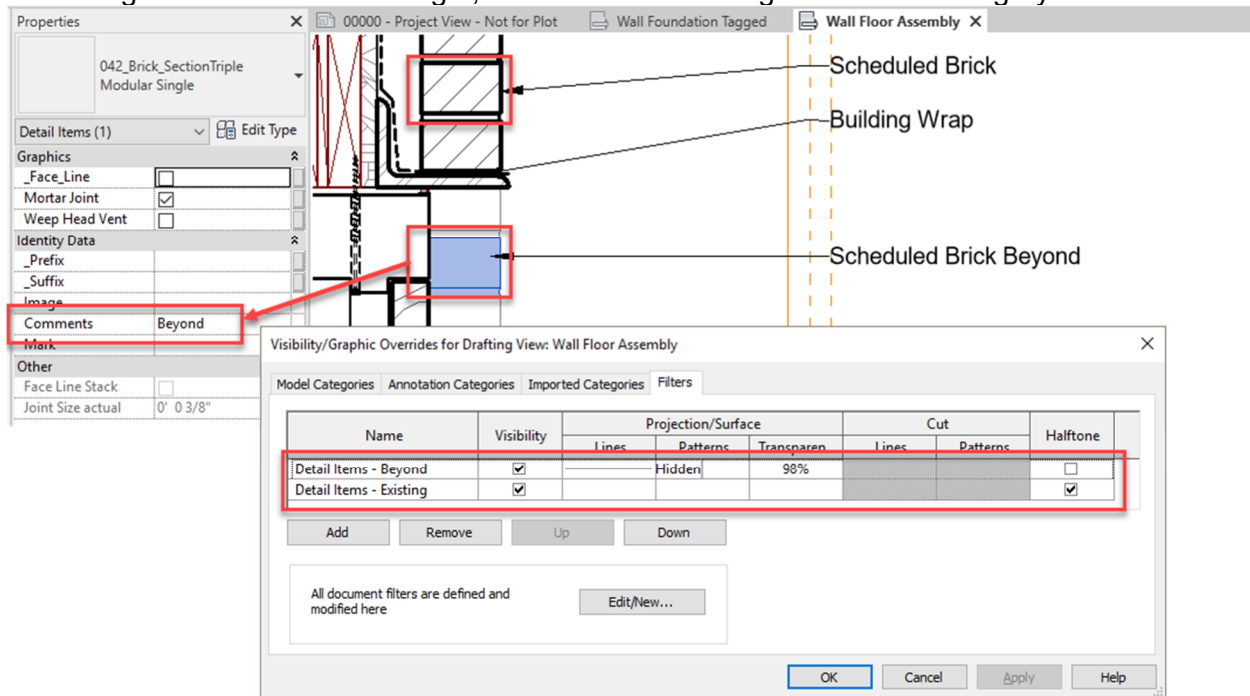
The OOTB detail item tag references the Type Mark value is center justified, and can be boxed or not. Utilizing a value other than Type Mark might be beneficial to gain more flexibility while tagging. For instance, a tag can use the Type Comments and/or Comments to allow for consistency in the Type Comments, but to also allow for flexibility with individual notes by using the instance based comments. A tag can then be set up to allow a user to change the tag type out if desired.





## Filters

Filters play a major role in modeled elements in both documentation and quality assurance, and can play just as large a role in detailing as well. Whether the file is set up for keynoting or tagging, the way the filters work will be similar. Because comments are for tagging purposes, the filters change the graphic look of the detail item based on the value entered. In this example, if “Beyond” is entered into the comment field, then the fill pattern is Hidden, the lines are changed to a thinner line weight, and the color is changed to a shade of grey.



Using filters can eliminate the need for additional family types as well as the manual labor of Overriding graphic in view by Element. Filters also add consistency to a project

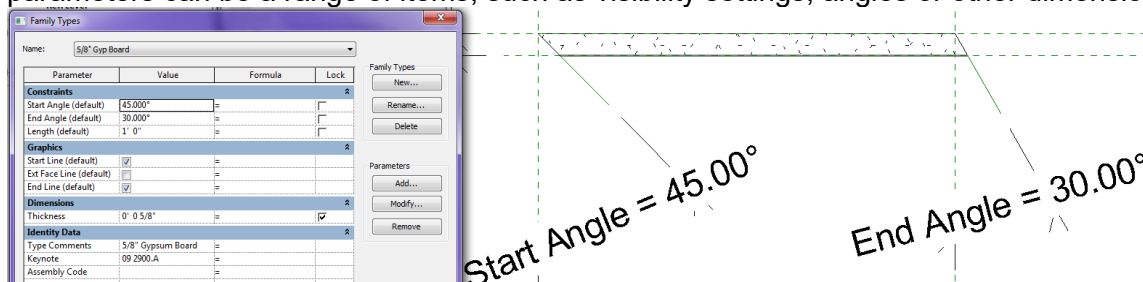
## Detail Item Families

In order to compile a complete library of detail items, creating new, custom content will become necessary, the OOTB content will not cover every required family. Thinking about all the options that can be added is a big step, and can seem daunting. Considering line weights is usually one of the easiest places to start. What line types, weights and styles are going to be needed to match the company standards? What fill patterns are needed for families? Are there going to be custom parameters for tagging or for creation? Will having custom Reference Plane subcategories make the family creation process more clear. All of these situations should be considered first then a detail item family template should be created. Utilizing a template will cause all the new families to be consistent, and creating each family will become less time consuming. Revit doesn't allow for the creation of family templates, however a family can be renamed within windows explorer from .rfa to .rft. So, create a family that includes the template information and then simply change the file extension. This process should be used for both detail item and detail item line based family types. (note: elements created in a template might not be able to be changed, for this reason keep the original .rfa file in a separate folder for future changes)



## Detail Items Line Based

Line based detail items also play an important role in detailing. Most of the OOTB families such as gypsum board or structural framing are line based.. These types of detail items can play a large role in detailing. For example a line based concrete family could be used as a wall, column, foundation, slab, or beam and this could a single family. Structural rebar, either in elevation or in section could also be a line based family. One of the most useful detail items is simply a line, which can be any line that needs to be called out: building wrap, vapor barrier, lines of elements beyond, etc. a line family is virtually limitless. Adding parameters to line based components adds another level of sophistication and adds to the flexibility of the family. These parameters can be a range of items, such as visibility settings, angles or other dimensions.

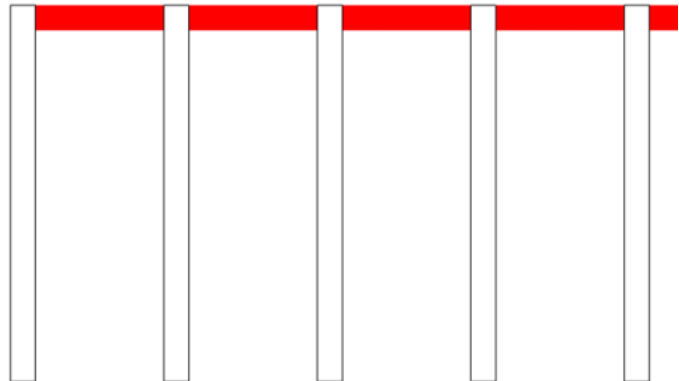




## Embedding Intelligence

A benefit of using detail items is the intelligence, automation, and efficiency they can add to details via tags. Instead of manually entering it, information can be embedded into a family such that it can be tagged once placed into a project. There are different ways this can be approached all with advantages and disadvantages. One of the ways is to use Revit Keynoting, a benefit of which is anyone in the office can edit the Keynote.txt data file without needing to have Revit installed (**Keynote Manager from Revolution Design is a must**), which can push information back into the project file without needing knowledge of Revit. In addition rights can be granted to only specific users in the company virtually locking down the taggable information for elements. Unfortunately, the keynotes can only have two values for tagging, the keynote number and keynote description, therefore any element specific information needed has to be added as text next to the keynote. One nice part about this approach is a keynote legend (schedule) can be created and then given to a spec writer, thereby limiting the time looking for all of the elements that have been defined. Another method is to embed information into parameters, (i.e. type comments) so that the intelligence can be extracted and updated as the element changes. With this method individual notes could be added to components by using an instance based parameter (i.e. comments).

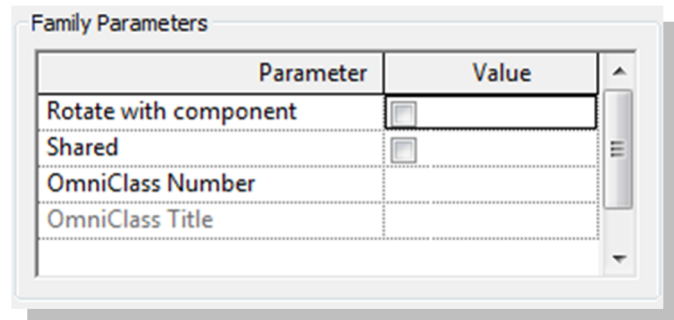
Intelligence in a detail family can also come via graphics and/or calculations, along with information. For instance, rebar placed in section can be set so that if the rebar becomes too close per code then a note, color change, or something else can happen in the family notifying the user that the configuration doesn't meet certain criteria. Or if a family is based on a manufacturer's data and one of the inputted values doesn't meet their specifications or sizes then, a user can be alerted.





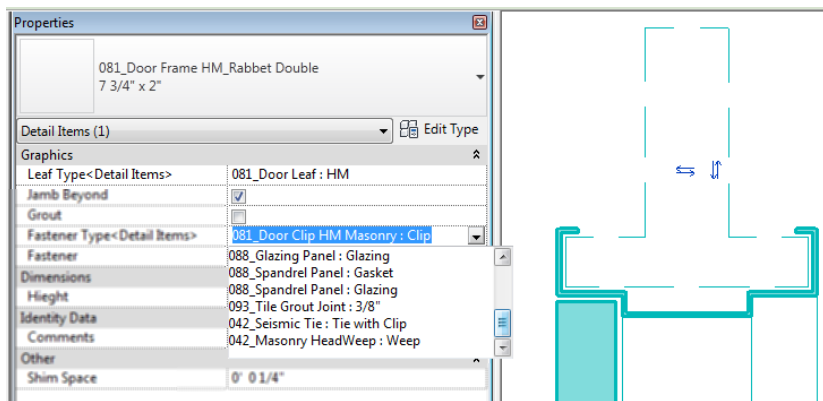
## Nested Detail Items

As more detail components are created, and as their complexity grows, nesting of other detail items becomes necessary. Nesting becomes vital if an array is to be used, if notes or flags are to be displayed, or simply if the swapping of one subcomponent for another is required.



One thing to take into consideration when nesting is tagging: if the nested element is going to be tagged then it will have to be shared. In each family type, under the Family Category and

Parameters, there is a check box for Shared. If this is selected the family can then be tagged or scheduled once it is loaded into another family. Unfortunately if a family is marked to be shared then type based parameters can't be mapped in the host. This means all parameters of a nest that are to be mapped must be instance based values.



Nesting is also required for family swapping, or changing one nested family type out for another. Using nesting for this purpose adds consistency, as well as flexibility, into a family. However, the drawback to this approach is that a user can change the value to any other shared family of the same type. This is why consistent family naming and understanding is a must. (In the below image the fastener type for the door can be changed to any nested shared detail item in the project)

## Schedules

If using type comments and comments, as mentioned above, then a must have tool for working in a project is a schedule. Schedules allow users to see all of the Type Comments and Comments (plus any additional parameters) being used within each project.

<Detail Item Schedule>					
A	B	C	D	E	F
Family	Type	Keynote	Type Comments	Comments	Count
000_Grid	Vertical		DO NOT TAG FAKE GRID		1
000_Nested CL	000_Nested CL				18
030_Concrete Keyway	6x2		Keyway for Concrete Above		2
030_Concrete_Section	6" Wall		6" Concrete Wall		2
030_Concrete_Section	10" Wall Foundation	03 3000.C	10" Concrete Foundation Wall		2
030_Concrete_Section	12" Footing		12" Strip Footings		2
030_Concrete_Section	Slab Typical		Scheduled Floor Assembly		4
032_Reinforcing_Bar_Side	05		#5 BAR		4
032_Reinforcing_Bar_Top	05		#5 BAR		3
032_Reinforcing_Bar_Top	05		#5 BAR	Centered in Wall	1
040_Air_Space	2"	-	Nominal Air Space		2
040_Masonry_Head_Weep	Weep		Weep Vent		1
042_Adj_MasonryVeneerAnchor_Side	0	04 2000.AA	ADJ Masonry Veneer Anchor		4
042_Brick_Section	Modular		Unit Masonry		10
042_Brick_SectionTriple	Modular Single		Scheduled Brick	Beyond	6
050_Rail_AnchorDeck_Side	12		12" Anchor Bolt		1

An additional benefit of a schedule is to use it for spell check. Since Revit can't spell check properties of families, a schedule may be exported out to a .txt file, where you can spell check the document and verify any corrections that are needed.

## Conclusion

This session shared a method of detailing by using only detail items, and by not using lines and filled regions, therefore allowing the use of tags instead of text, and bringing a level of intelligence to the 2D side of Revit that is often limited to the 3D side. The session also discussed how to embed intelligence into your component.



[Brian@BDMackeyConsulting.com](mailto:Brian@BDMackeyConsulting.com)



@TheRevitGeek



/in/MackeyBrian



[BDMackeyConsulting.com/Blog](http://BDMackeyConsulting.com/Blog)