# IS THAT NEW? NEW STUFF IN THE FAMILY EDITOR AND SOME STUFF YOU MAYBE JUST FORGOT…

*Paul F. Aubin*

If you create Revit family content, there has been a slow trickle of new features that have crept into the family editor in recent releases. They just have not received much fanfare, so let's play a little game of: "Is that new?" We'll look at new features like: room aware location points, new family templates and let's not forget reordering parameters! We'll explore 3-point adaptive arcs and the newest new feature: "load into project and close". But we won't just focus on new stuff. Using some slightly obscure techniques we can make a parameter hidden in the project environment so you can use it to drive formulas without users inadvertently modifying it (because they won't even know it's there). We will explore using family types parameters to build drop-down lists and play some tricks with categories and shared vs. nonshared families. If you create family content even occasionally, there's bound to be a topic in this session that will have you asking: "Is that new?"

## Learning Objectives

- Learn how to control the order of the parameters in your family's properties
- Create true "three-point" adaptive arcs and work with two-level generic models
- Learn to create drop-down lists to drive parameters and make parameters hidden from your users

### *About the Speaker*

Paul Aubin is the author of many Revit book titles including the widely acclaimed: The Aubin Academy Mastering Series: and the all new Renaissance Revit and Revit video training at www.lynda.com/paulaubin. Paul is an independent architectural consultant providing Revit® Architecture implementation, training, and support services. Paul's involvement in the architectural profession spans over 20 years, with experience in design, production, CAD management, mentoring, coaching and training. He is an active member of the Autodesk user community, and is a high-rated repeat speaker at Autodesk University, Revit Technology Conference and a national speaker at the BIM Workshops. His diverse experience in architectural firms, as a CAD manager, and as an educator gives his writing and his classroom instruction a fresh and credible focus. Paul is an associate member of the American Institute of Architects and a member of the Institute of Classical Architecture. He lives in Chicago with his wife and three children.

## Introduction

I have broken the session and paper down into three major headings: Templates, Modeling and Parameters. In each of these topics will be several subtopics. In playing our game of what's new and what's not, I'll be considering the last few releases of Revit. So if a feature was introduced or enhanced in 2014, 2015 or 2016, I'll call it new. So let's get started.

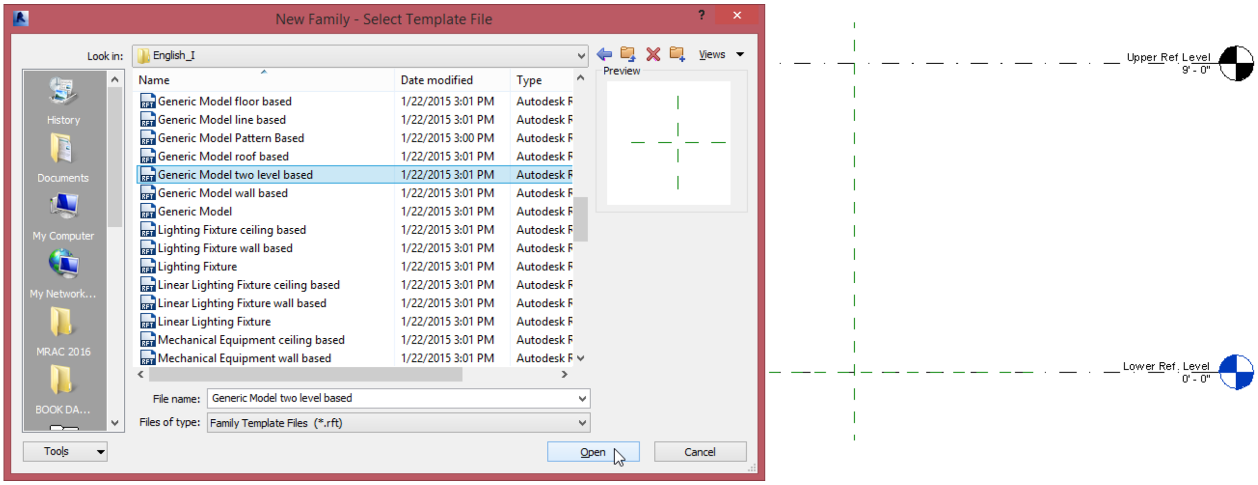
## Templates

Templates provide the starting point for every new family. So has there been anything new in family templates? Let's begin our game of what's new with a look at family templates.

### New Family Templates

Usually when a new feature is added to the software; we will also see some new family templates arrive. For example, when the conceptual massing environment was introduced we got several new templates that supported and enabled it. When stairs and railings were enhanced a few releases ago, we got two new templates for railings called: *Railing Support.rft* and *Railing Termination.rft*. (Actually if we are being really technical, these are new names for these templates which previously did not contain the word "Railing"). Their purpose is to allow you to create custom railing supports and terminations when you customize handrail types. (These are available for handrails only, not top rails or non-continuous rails).

The template that I want to discuss is not tied to any specific new feature and therefore is often overlooked. It is the *Generic Model two level based.rft* template. This template gives you two levels and their associated constraints in the resultant family (see Figure 1).



New Family - Select Template File

Look in: English_I

| Name | Date modified | Type |
|---|---|---|
| Generic Model floor based | 1/22/2015 3:01 PM | Autodesk F |
| Generic Model line based | 1/22/2015 3:01 PM | Autodesk F |
| Generic Model Pattern Based | 1/22/2015 3:00 PM | Autodesk F |
| Generic Model roof based | 1/22/2015 3:01 PM | Autodesk F |
| Generic Model two level based | 1/22/2015 3:01 PM | Autodesk F |
| Generic Model wall based | 1/22/2015 3:01 PM | Autodesk F |
| Generic Model | 1/22/2015 3:01 PM | Autodesk F |
| Lighting Fixture ceiling based | 1/22/2015 3:01 PM | Autodesk F |
| Lighting Fixture wall based | 1/22/2015 3:01 PM | Autodesk F |
| Lighting Fixture | 1/22/2015 3:01 PM | Autodesk F |
| Linear Lighting Fixture ceiling based | 1/22/2015 3:01 PM | Autodesk F |
| Linear Lighting Fixture wall based | 1/22/2015 3:01 PM | Autodesk F |
| Linear Lighting Fixture | 1/22/2015 3:01 PM | Autodesk F |
| Mechanical Equipment ceiling based | 1/22/2015 3:01 PM | Autodesk F |
| Mechanical Equipment wall based | 1/22/2015 3:01 PM | Autodesk F |

File name: Generic Model two level based

Files of type: Family Template Files (*.rft)

**FIGURE 1—TWO LEVEL BASED TEMPLATE GIVES A TOP AND BOTTOM REFERENCE LEVEL TO FAMILIES CREATED FROM IT**

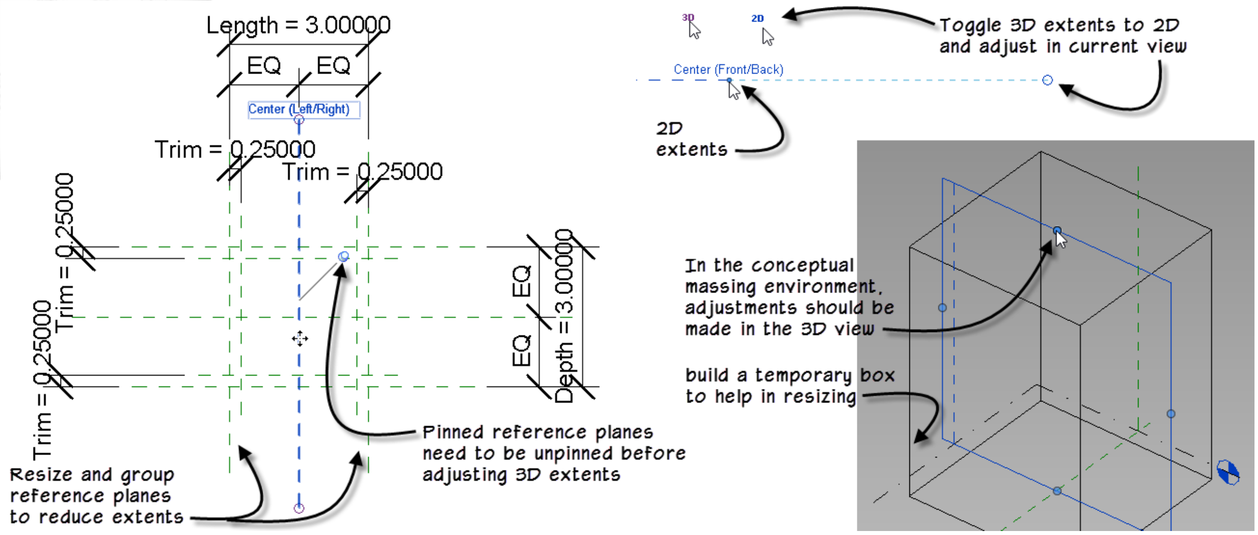
So when you build a family from this template, you will be able to assign both a top and bottom level to the object when loaded in a project. I have found this template useful to create elevator and escalator families. Sure, in the past, you could use the *Column.rft* template and re-categorize it to achieve the same result, but now we have this template instead.

***No surprise, this one is new.***

## Templates and Seeds

Speaking of family templates, do you ever find the configuration of the provided templates to sometimes be lacking? I find that there are many things I like to customize in the templates I am using for a particular family creation task. Naturally the specifics will vary somewhat from one piece of content to the next, but there are some common "themes" that are worth noting. Here is my short list:

**Reference Plane sizing**—This is perhaps the most common thing I modify. I find that the extents of most family templates are larger than I need. I don't like all the extra zooming that is required when reference planes are too long. The other issue is clarity. When you have lots of reference planes, it can be difficult to tell which is which. Since we cannot color code them or create types (and boy do I wish we could…) I instead rely on the lengths. I pair and group them at different lengths; to help give some visual hierarchy (see Figure 2). If you need to resize a pinned reference plane, you will need to unpin it first. (This is the case with the center reference planes in all of the out-of-the-box templates). Also keep in mind that when you add new reference planes, even if you make them short when adding them, they will be full length in other views. Very annoying.



Toggle 3D extents to 2D and adjust in current view

2D extents

In the conceptual massing environment, adjustments should be made in the 3D view

build a temporary box to help in resizing

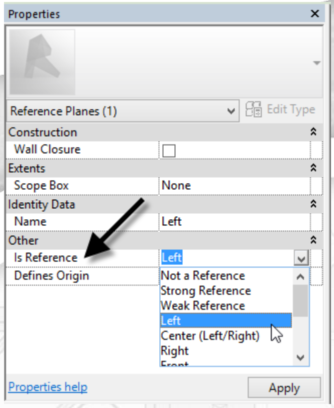
Pinned reference planes need to be unpinned before adjusting 3D extents

Resize and group reference planes to reduce extents

**FIGURE 2—RESIZING REFERENCE PLANES MAKES VIEW NAVIGATION EASIER HELPS WITH VISUAL HIERARCHY**

My preference is to resize the 3D extents. This is the open circle at the end of the selected reference plane. As an alternative (top-right of the figure) you can toggle and resize the 2D extents instead. However, this only affects the current view of course. If you want to resize reference planes in a massing family in the Conceptual Massing Environment (CME), then it is even trickier. If you resize in any orthographic view, it defaults to 2D extents and you cannot toggle to 3D. The only way to resize the 3D extents is to work in a 3D view. I usually draw a simple box in 3D, and then use this assist in sizing the reference planes and levels consistently. Delete the box when you are done.

**Is Reference**—Part of the "smarts" of a family is cluing Revit in to which side is left, and which side it right; where the center is, etc. The "Is Reference" setting controls this as well as helping establish dimension and aligning behavior. Most of the provided templates use the Center (Left/Right) and Center (Front/Back) Is Reference setting by default. So you should not need to adjust these. Some of them also include other reference planes and there Is Reference setting as well. Some don't. I recommend you create four reference planes in plan view (two in each direction) and set them to Left, Right, Front and Back. Name them too. Depending on the family, you may also want to include Top and/or Bottom reference planes as well (see Figure 3).



Properties

Reference Planes (1)

Construction

Wall Closure □

Extents

Scope Box None

Identity Data

Name Left

Other

Is Reference Left

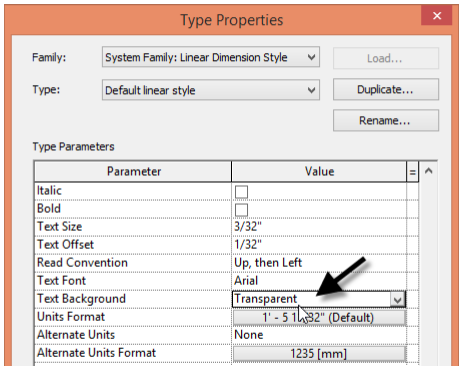
Defines Origin

**FIGURE 3—BE SURE TO CONFIGURE THE IS REFERENCE SETTING**

Also consider your minor reference planes and if you want them to be considered in dimensioning and the align tool. Any that are not needed for dimensioning, should be set to "Not a Reference." This will make the family behave better in projects.

**Scale**—I always adjust the scale of the views in the family editor. This adjusts the size of all the dimensions (and text) in the view. The lineweights also adjust as you would expect. This makes it SO much easier to work in family editor views. Remember, these views only appear in the family editor, so choose any scale that makes things more legible onscreen and more comfortable to work.

**Transparent Dimensions**—Nothing bugs me more than placing a dimension in the family editor (especially radii and diameters) and having the text cover up all of my geometry. Easy fix. Edit the dimension types and set the text background to transparent (see Figure 4). Also see previous tip about scale. That helps a lot too.



Type Properties

Family: System Family: Linear Dimension Style

Type: Default linear style

Type Parameters

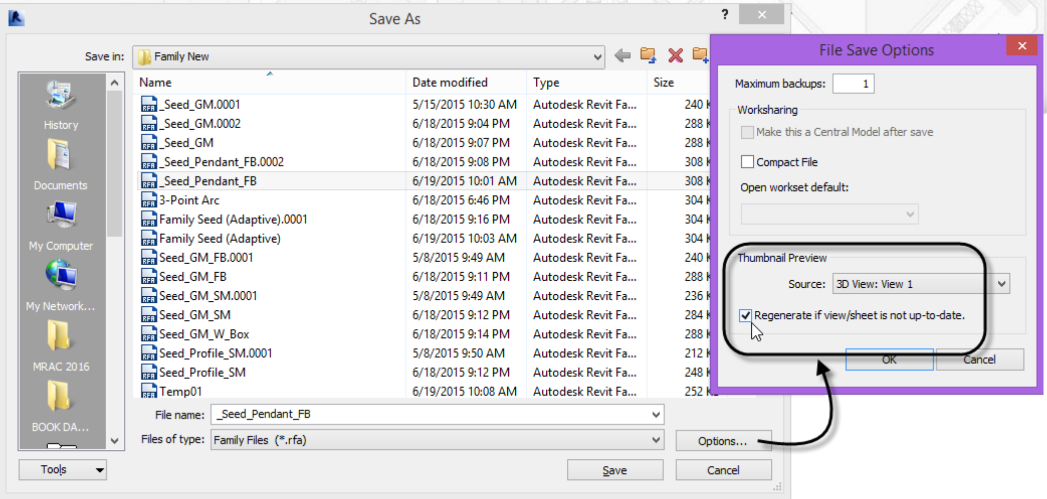
| Parameter | Value |
|---|---|
| Italic | □ |
| Bold | □ |
| Text Size | 3/32" |
| Text Offset | 1/32" |
| Read Convention | Up, then Left |
| Text Font | Arial |
| Text Background | Transparent |
| Units Format | 1' - 5 1/32" (Default) |
| Alternate Units | None |
| Alternate Units Format | 1235 [mm] |

**FIGURE 4—SET DIMENSION TEXT BACKGROUND TO TRANSPARENT**

**Units**—You don't have to leave the units of the family set at their default. Depending on the content I am building, I often change the units to decimal inches or other settings that might be convenient for the family in question. Nice whole number units can make formulas easier to troubleshoot as well.

**Materials**—Always open up the Material Browser in the family editor and see what you are starting with. There may be materials that you can purge as well as ones that you want to add. If you are building a collection of doors, you might want to create a palette of appropriate materials and include them in all of the families. Just be careful to not add too many materials. This can bloat the family. So as an alternative, you can use material libraries to save common materials and make them available for users to load.

**Generating previews**—Make a 3D view for your preview. In this view, you can turn on whatever visual style you like and you can use visibility/graphics to hide categories you don't want to include in the preview. For individual items, you can also use Temporary Hide/Isolate. Just make sure the temporary hide is active in the 3D view when you regenerate the preview. You can do this with Save As (see Figure 5).



Save in: Family New

| Name | Date modified | Type | Size |
|---|---|---|---|
| _Seed_GM.0001 | 5/15/2015 10:30 AM | Autodesk Revit Fa... | 240 K |
| _Seed_GM.0002 | 6/18/2015 9:04 PM | Autodesk Revit Fa... | 288 K |
| _Seed_GM | 6/18/2015 9:07 PM | Autodesk Revit Fa... | 288 K |
| _Seed_Pendant_FB.0002 | 6/18/2015 9:08 PM | Autodesk Revit Fa... | 308 K |
| _Seed_Pendant_FB | 6/19/2015 10:01 AM | Autodesk Revit Fa... | 308 K |
| 3-Point Arc | 6/18/2015 6:46 PM | Autodesk Revit Fa... | 304 K |
| Family Seed (Adaptive).0001 | 6/18/2015 9:16 PM | Autodesk Revit Fa... | 304 K |
| Family Seed (Adaptive) | 6/19/2015 10:03 AM | Autodesk Revit Fa... | 304 K |
| Seed_GM_FB.0001 | 5/8/2015 9:49 AM | Autodesk Revit Fa... | 240 K |
| Seed_GM_FB | 6/18/2015 9:11 PM | Autodesk Revit Fa... | 288 K |
| Seed_GM_SM.0001 | 5/8/2015 9:49 AM | Autodesk Revit Fa... | 236 K |
| Seed_GM_SM | 6/18/2015 9:12 PM | Autodesk Revit Fa... | 284 K |
| Seed_GM_W_Box | 6/18/2015 9:14 PM | Autodesk Revit Fa... | 288 K |
| Seed_Profile_SM.0001 | 5/8/2015 9:50 AM | Autodesk Revit Fa... | 212 K |
| Seed_Profile_SM | 6/18/2015 9:12 PM | Autodesk Revit Fa... | 248 K |
| Temp01 | 6/19/2015 10:08 AM | Autodesk Revit Fa... | 252 K |

File name: _Seed_Pendant_FB

Files of type: Family Files (*.rfa)

Maximum backups: 1

Worksharing

□ Make this a Central Model after save

□ Compact File

Open workset default:

Thumbnail Preview

Source: 3D View: View 1

☑ Regenerate if view/sheet is not up-to-date.

**FIGURE 5—USE THE OPTIONS BUTTON IN THE SAVE AS DIALOG TO SET AND UPDATE A PREVIEW**

**Add Common Parameters**—if you are doing a collection similar families, you might want to add the common parameters. For example, most families need some overall dimensions like width, length and height. You might need some common material parameters: door panel, door frame, etc. And certainly any other parameters that will be part of most families in the collection.

## So what's the point of all this? Is any of this new?

**Save as Seed or Template**—No. Most of this is not new. But the goal of all of these settings is to give you a starting point for your families that is much more complete and avoids much of the redundant effort. You have two options here. I typically use what I like to call a "seed" family. A seed family is just a family saved with all of the above settings and given a good descriptive name. Simply save the file. It will be an RFA file. When you want to use the seed, you open the RFA and do a save as. That's it.

The other alternative is to go one step further. You cannot save a family template within Revit. But you can rename and RFA file in Windows Explorer and change the extension to RFT. When you do so, it will become a Revit family template. You can then locate the custom template on the network server for users to access. Keep one important point in mind if you opt for this approach. Any reference planes, geometry and constrained dimensions that you added in the original family will now be "permanent." This means that they cannot be deleted. (Interestingly, labeled dimensions can be deleted, and constrained dimensions can be unlocked however). So just be certain that these behaviors are what you want before renaming the file. Furthermore, to edit the template later, you will need to rename it back to an RFA. However, "permanent" elements will still be permanent even after this renaming and cannot be deleted.

> Wait, is that new? Surely that bit about renaming the template is new right? Nope. None of this is new.

## Face-Based Templates

One more template tip for you. If you have been creating family content for a while you have no doubt used and/or created face-based content. Face-based content has many advantages which I won't go into here. But there are a few disadvantages as well. There are two which are notable. First, if you try to follow the tip above about using the Is Reference parameter, you will have to make some decisions (and frankly a judgement call) on the most likely orientation of your family. Since it is face-based, you cannot always say for certain where: front, back, left and right are. In this case you will have to just make a judgment call and try to build the family for the most common scenario you can envision.

The other tricky issue with face-based content is the orientation of the thumbnail preview. If you have even seen plumbing fixtures that look like they are standing up sideways on the floor or light fixtures that look like they are upside down, then you have seen the problem. There is a trick you can do using Copy/Monitor that provides a solution to the problem. This will only work with certain categories. **Specifically those allowed by Copy/Monitor**. Here is the process:

1. Create a new family using the: *Generic Model wall based.rft* template.

⇨ Set the category to one allowed by Copy/Monitor such as: Plumbing Fixtures or Lighting Fixtures.

⇨ Open the *Placement Side* elevation and create a simple cylinder pointing away from the wall surface.

2. Load this new family into a project and add an instance of it to any wall in the project.

⇨ Save and close the project file.

3. Create or open another project file and link in the project we just closed.

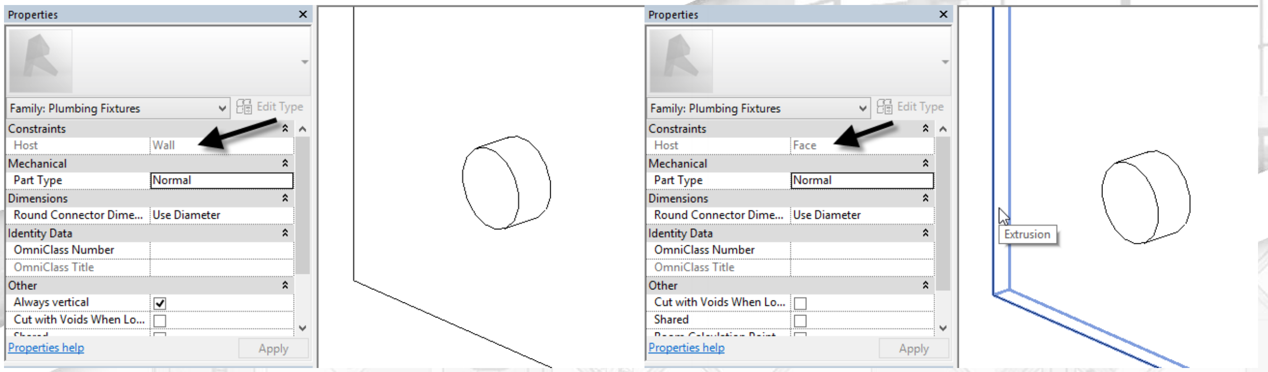
4. On the Collaborate tab, on the Coordinate panel, click the Copy/Monitor drop-down and choose: **Select Link**.

⇨ On the Tools panel, click the Copy tool, select the cylinder (face-based family you created above) and then click the Finish button on the ribbon.

You now have a new family in the current host project. It will have the same name as the one you built above.

5. Select the new family onscreen, and then on the ribbon, click the Edit Family button.

This will open it in the family editor, but if you look on the Properties palette, you will notice that the host is now set to Face instead of Wall. You now have a face-based family that is oriented vertically (see Figure 6).



| Family: Plumbing Fixtures | |
|---|---|
| Constraints | |
| Host | Wall |
| Mechanical | |
| Part Type | Normal |
| Dimensions | |
| Round Connector Dime... | Use Diameter |
| Identity Data | |
| OmniClass Number | |
| OmniClass Title | |
| Other | |
| Always vertical | ☑ |
| Cut with Voids When Lo... | □ |

| Family: Plumbing Fixtures | |
|---|---|
| Constraints | |
| Host | Face |
| Mechanical | |
| Part Type | Normal |
| Dimensions | |
| Round Connector Dime... | Use Diameter |
| Identity Data | |
| OmniClass Number | |
| OmniClass Title | |
| Other | |
| Cut with Voids When Lo... | □ |
| Shared | □ |

**FIGURE 6—ORIGINAL WALL-BASED FAMILY (LEFT) AND VERTICALLY ORIENTED FACE-BASED VERSION (RIGHT)**

6. Delete the cylinder, add some reference planes, parameters or any of the other items you want in your seed/template.

⇨ Save the file as a new Seed or Template family as discussed above.

This approach gives you a nicer preview in Window Explorer and when you browse in the load family dialogs. The Type Selector still seems pretty stubborn about showing the default preview orientation however. *Thanks to Brian Mackey for sharing this tip with me.*

So is it new? Nope. Not new, just cool.

## Modelling

With a good template in place, we can direct our attention to modeling. We have a few items to look at here.

### 3-Point Arcs in the CME

Let's start with a simple one. If you have spent any time in the CME, you have likely created a Spline by Points. Spline by points allows you to create a collection of reference points (as few as two) and then select them and create a spline through those points by simply clicking the button. These points will "drive" the shape of the spline. This means you can move any of the points and the spline will re-shape. This is a very powerful feature. While splines work great for many forms, sometimes you really need a true circular arc. To do that, try making a 3-point arc through points!

1. Create a new family using the: *Generic Model Adaptive.rft* template.

2. On the Draw panel, click the Point icon and place three points randomly anywhere onscreen.

⇨ Cancel the command, select the three points and then on the Draw panel, click the Spline Through Points icon.

This is a spline through points. You can place just two points and you will get a straight line and you can place many more than three to get a more complex curve.

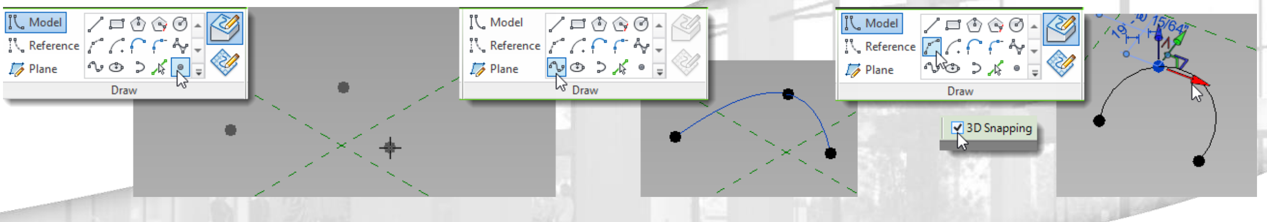


**FIGURE 7—SPLINE BY POINTS ON THE LEFT AND 3-POINT ARC BY POINTS ON THE RIGHT**

⇨ Cancel any command, select one of the points and drag it.

Notice that the spline reshapes as you drag the point(s).

Now the trouble with this is that this is not an arc but a spline. If you want an arc, start with the Start-end-radius tool instead (see Figure 7).

3. On the draw panel, click the Start-end-radius tool.

⇨ On the Options Bar, check the 3D Snapping box. (This is required to make the points and have them be driving points).

⇨ Click three points onscreen to place the arc. Cancel, select a point and drag it.

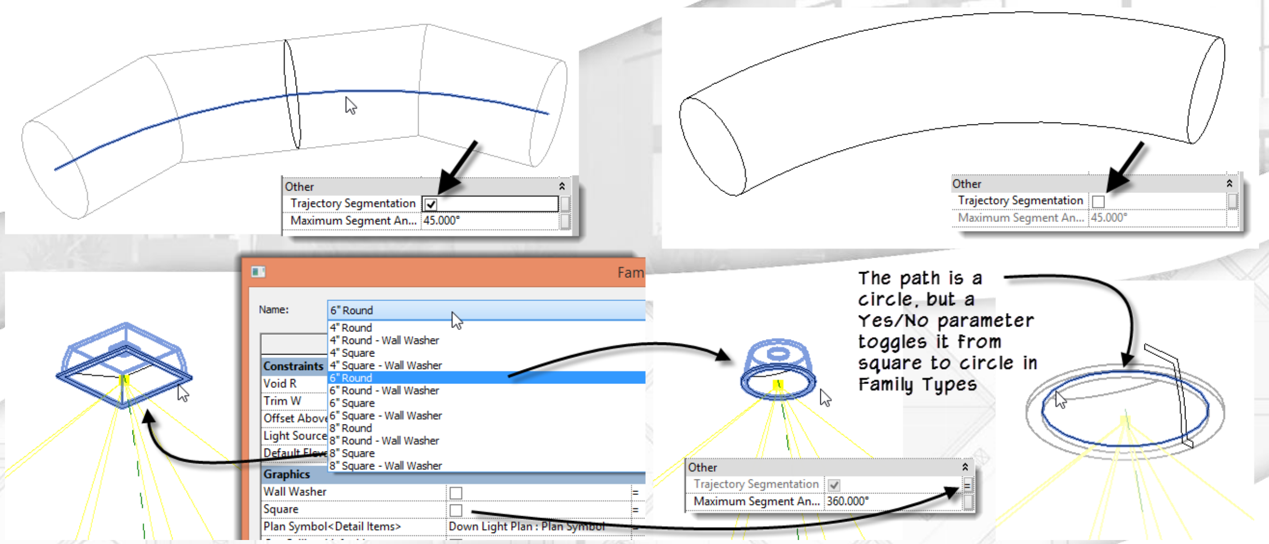
Notice how these points will drive the shape but it will remain an arc! Nice!

Is that new? You bet.

At least it was new about a release or two ago.

## Trajectory Segmentation

Now this one is one of my favorites and I am always pleased with myself when I can utilize it. This feature is affects the path of your sweep. Specifically, it affects only curved segments of the path and renders them in segments instead of a smooth curve. It is listed in the help as an MEP feature. They show an example like the one at the top of Figure 8. While this is certainly a valid use for it, what I find most exciting about the feature is that it can controlled by parameters!



Other
Trajectory Segmentation ☑
Maximum Segment An... 45.000°

Other
Trajectory Segmentation □
Maximum Segment An... 45.000°

Name: 6" Round

Wall Washer □
Square □
Plan Symbol<Detail Items> Down Light Plan : Plan Symbol

The path is a circle, but a Yes/No parameter toggles it from square to circle in Family Types

Other
Trajectory Segmentation ☑
Maximum Segment An... 360.000°

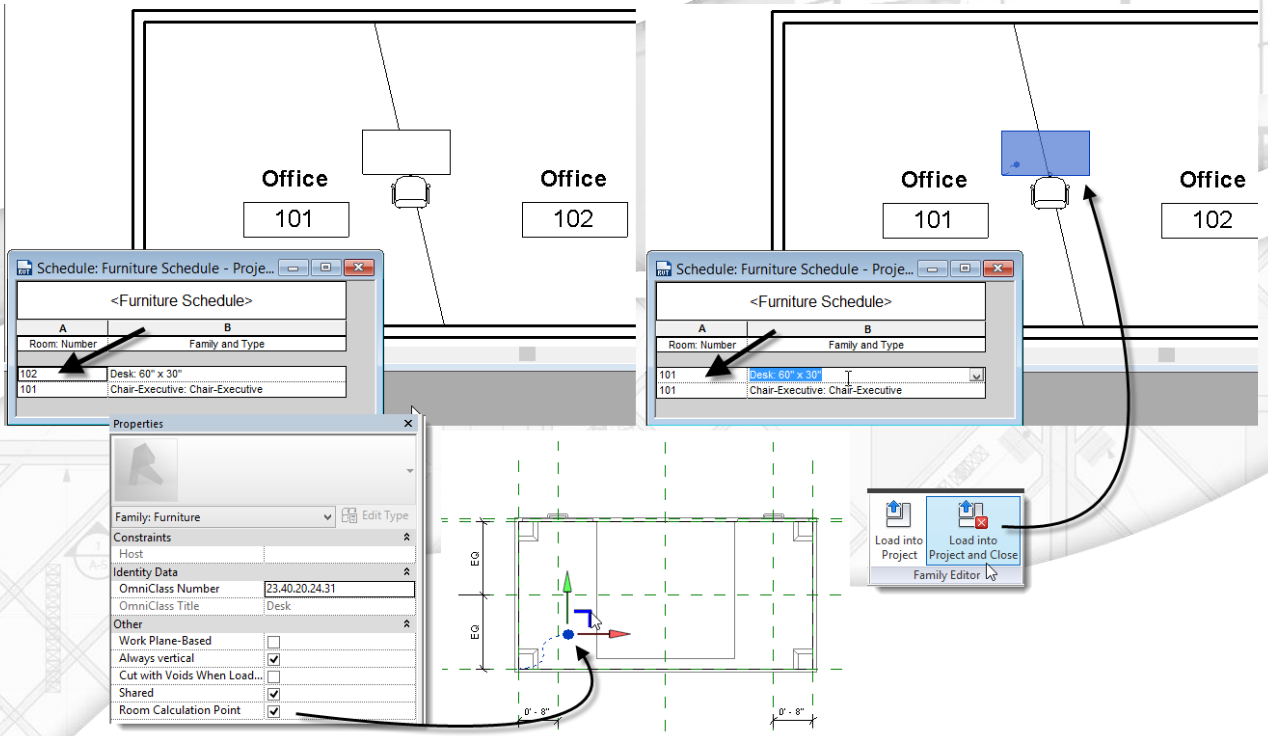
**FIGURE 8—TRAJECTORY SEGMENTATION CAN BE SET MANUALLY OR WITH A PARAMETER**

Looking at the bottom of the figure, I am showing an example where I used this feature to create a recessed downlight fixture. The same fixture can flex to different sizes and be either round or square using trajectory segmentation. The path for the sweep is a circle (shown at the right). Then the trajectory segmentation feature is linked to a family parameter (Yes/No) that is able to toggle it on or off. When it is on, we get a square. When it is off it is circular. Because the path is a circle, these are the only two options. If you use an arc for your path, like the example, at the top of the figure, you can also change the Maximum segmentation angle setting to get more or fewer segments along the curve. Smaller numbers give more segments.

Is that new? No. But very useful just the same.

## Room Calculation Point

The Room Calculation Point feature allows you to adjust how a family decides which room it occupies. This is a rather obscure feature with dubious use case, but here goes. Suppose you had the situation illustrated in Figure 9 with some furniture very near a room separation line. In this case, the two items report in different rooms. You can enable the Room Calculation Point to adjust this.



Schedule: Furniture Schedule - Proje...

| <Furniture Schedule> | |
|---|---|
| A | B |
| Room: Number | Family and Type |
| 102 | Desk: 60" x 30" |
| 101 | Chair-Executive: Chair-Executive |

Schedule: Furniture Schedule - Proje...

| <Furniture Schedule> | |
|---|---|
| A | B |
| Room: Number | Family and Type |
| 101 | Desk: 60" x 30" |
| 101 | Chair-Executive: Chair-Executive |

Properties

Family: Furniture — Edit Type

| | |
|---|---|
| **Constraints** | |
| Host | |
| **Identity Data** | |
| OmniClass Number | 23.40.20.24.31 |
| OmniClass Title | Desk |
| **Other** | |
| Work Plane-Based | ☐ |
| Always vertical | ☑ |
| Cut with Voids When Load... | ☐ |
| Shared | ☑ |
| Room Calculation Point | ☑ |

**FIGURE 9—ROOM CALCULATION POINT DISABLED (TOP-LEFT) USES GEOMETRIC CENTER. ENABLED (TOP-RIGHT) USER DEFINED LOCATION IS USED**

1. Edit the family, and then on the Properties palette, check the Room Calculation Point box.

⇨ Optionally save the family or just load it into your project without saving.

2. If this is your only edit, you can use the "Load into Project and Close" button.

When you open a door or window family, the icon will look slightly different which accounts for the fact that there are two room properties for doors and windows: the From Room and To Room. So the icon will have arrows indicating these directions and you can flip them as required. I like the concept of a Room Calculation Point, but I cannot say I am wild about how it was implemented. Having it as a family-level option seems very limited. If it were something that we could adjust per-element in the project with a control grip, it would be much more useful. So far the only mildly useful place where I have implemented this feature is in lighting fixture families. I enable the feature and drag the grip down in the "Z" direction to ensure that lights read the room below them rather than above. But even here I doubt this is really necessary in most cases.

> So is that new? Two for one actually, Room Calculation Point a few releases ago, and Load into Project and Close is new this release!

## Exploding CAD

When building the geometry for your family it is always preferable to build it from native geometry. However, there are certainly times when imported geometry will be incorporated into a family. If the imported geometry contains solids, did you know that you can explode it and end up with an editable freeform element? I have found this useful in certain situations. For example, it is useful for creating complex void elements. Create a complex solid form in AutoCAD and then explode it in Revit. Select the exploded form and on the Properties palette, you can change it to a void.

## New? Yes, this is new.

## Nested Rigs

When controlling rotation and angular movement in a parametric family, it is common to use reference lines with angular parameters. Angular parameters do not work well on reference planes, as they often shift unexpectedly when you flex them. But reference lines have explicit endpoints, so you can lock them down and make a more stable rotation rig. However, if you want to control rotation in a Profile family, you cannot add reference lines. So what to do? One option is to stick with a rectilinear rig of reference planes. You can use trigonometry to ensure that diagonally opposite corners give the correct angles. But in this example, we'll look at an alternative: using a "nested rig" on which you can hang your profile form.

1. Create a new profile family and add reference planes and dimensions to make a flexible rectangle (see Figure 10).

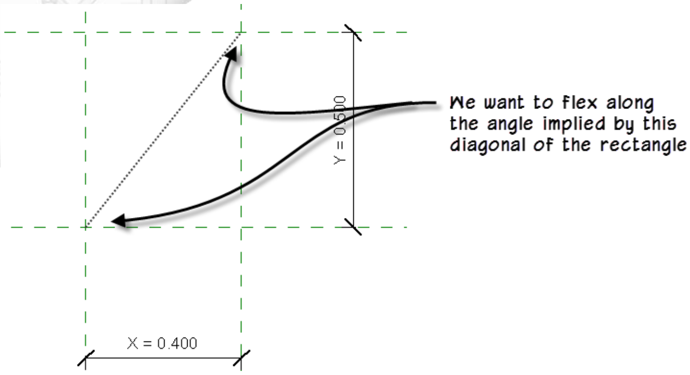

Y = 0.500
We want to flex along the angle implied by this diagonal of the rectangle
X = 0.400

**FIGURE 10—THE ANGLE WE WANT IS DEFINED BY THE WIDTH AND HEIGHT OF A RECTANGULAR REFERENCE PLANE RIG**

The diagonal across the rectangle represents the angle we want to make flexible. We cannot use reference lines in profile families. And while we can use reference planes, as noted they do not work well in controlling angles. So instead of reference lines or planes, we will simply draw lines instead. However, if we draw the lines directly in the profile family, they will be seen by Revit as part of the profile. If we instead draw our guide lines in a Detail Item family they can be used for our framework or "rig" and not be seen as part of the profile.

2. Create another family from the *Detail Item.rft* template.

⇨ Save the file as: **Single Angle Rig**.

3. Create four new reference planes: Left, Right, Back and Front equally spaced around the two center planes.

4. Name them, set the Is Reference settings and dimension and label them so that everything flexes equally.

   Make these instance parameters.

5. On the Create tab, click the Line tool. Snap the first point to the intersection of the Left and Front reference planes and the second point to the Right and Back reference planes.

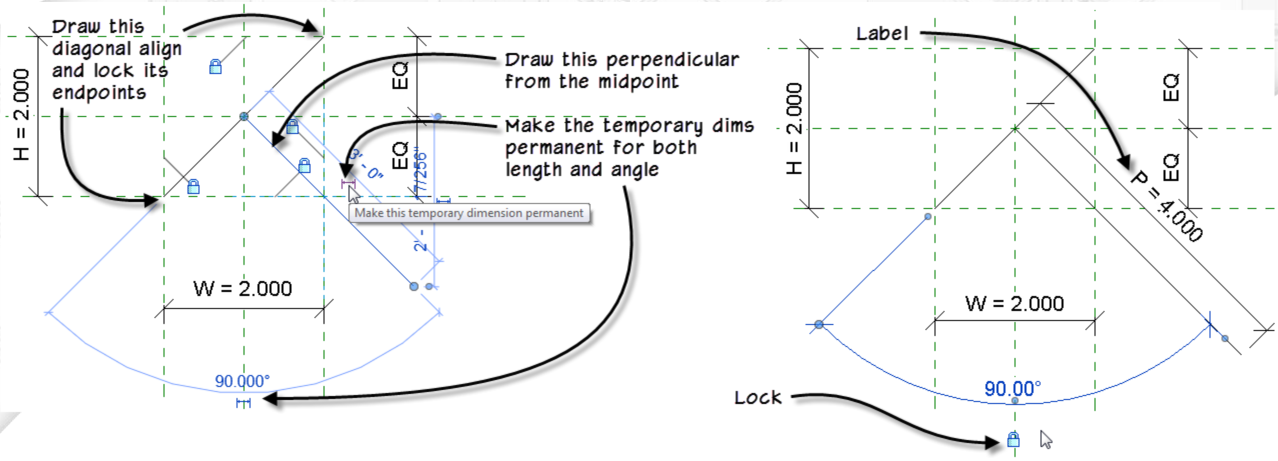
- Align and lock the endpoints to the reference planes in both directions on each end.
- Flex to be sure it adjusts with the width and depth.

6. Draw a second line starting at the midpoint of and perpendicular to the first line. Make it approximately the same length as the other one.

- Select the line and then click the small "Make this temporary dimension permanent" icon for both the length and angle dimensions.

7. Select the new linear dimension and label it with a new parameter.

- Lock the angle dimension to keep it at 90° (see Figure 11).


Draw this diagonal align and lock its endpoints
H = 2.000
EQ
EQ
Draw this perpendicular from the midpoint
Make the temporary dims permanent for both length and angle
Make this temporary dimension permanent
W = 2.000
90.000°
Label
H = 2.000
EQ
EQ
P = 4.000
W = 2.000
90.00°
Lock

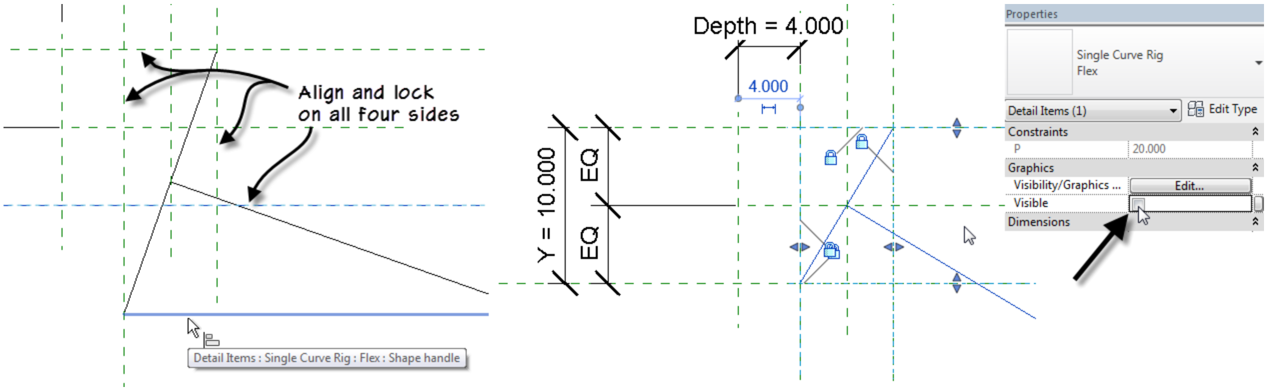
### FIGURE 11— BUILD THE RIG IN A DETAIL ITEM FAMILY

Optionally you can open Object Styles and create a new subcategory. Call it: Guide Lines and then set the color to a light blue. This will help the rig stand out when nested into other families.

8. Save the family and then click the Load into Project and Close button (New button!).

- Click to place it onscreen. Align and lock it on all four sides.

For each alignment, first click one of the reference planes in the host family, and then click the nested shape handle edge in the detail item family. Use TAB if necessary to get "shape handle" each time (see Figure 12).



P 20.000

### FIGURE 12— NEST IN THE RIG, ALIGN AND LOCK IT ON ALL FOUR SIDES

9. Flex the family.

Notice that the nested detail item family changes shape with the host family. The diagonal line stays aligned with the flexing reference planes and the perpendicular line remains perpendicular.

10. Select detail item and on the Properties palette, uncheck the Visible checkbox.

This makes the rig invisible in all families that use the profile; we will only see it here were it is needed.

With the rig in place, you can now draw your profile and constrain linework to it. This linework will flex with the family and rig as it flexes.

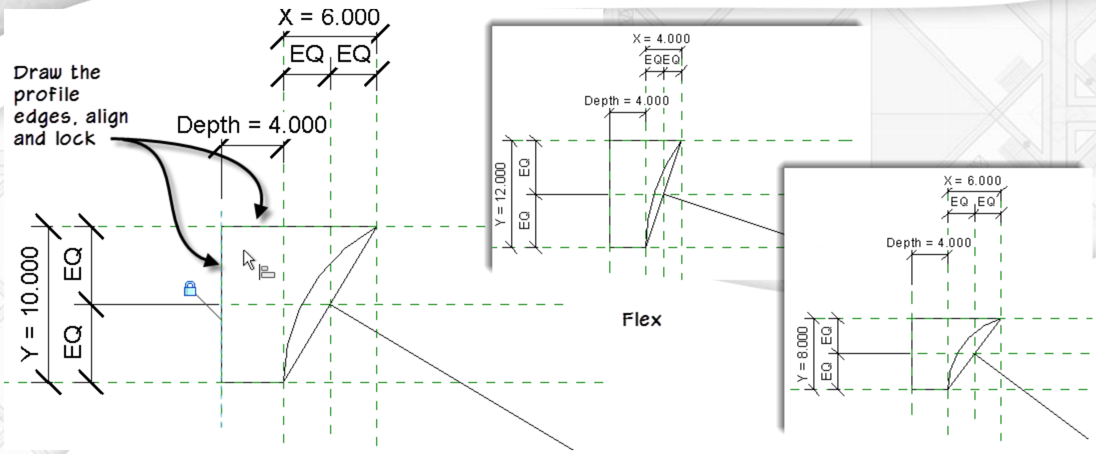


**FIGURE 13— CREATE THE PROFILE SHAPE AND ALIGN AND LOCK AS NECESSARY**

Here I am showing a cavetto curve profile (see Figure 13). This profile will flex nicely based on the rig built here and we can load it into any project and use it define sweeps for moldings or other details as required.

Nested Detail Rigs in Profile Families, that's pretty cool, must be new? Sorry no. Not new. Been there for a while now…

## Nested Family Visibility (Cutable)

I was working on a collection of shower families for a client recently. We wanted the shower pan to show as cut in sections (with a nice profile line around it). Trouble is, the Plumbing Fixture category is not cutable. So what to do? Well, all you have to do is insert a nested family that IS cutable and it will happily display as cut even while its host displays in projection. So in this particular example, the solution was to nest in a Generic Model for the shower pan (see Figure 14).

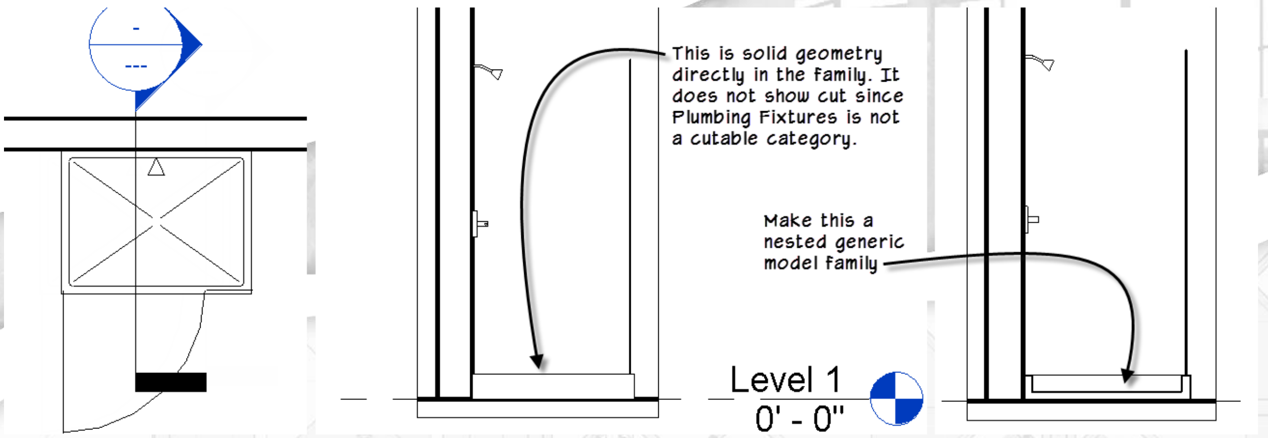

This is solid geometry directly in the family. It does not show cut since Plumbing Fixtures is not a cutable category.
Make this a nested generic model family
Level 1
0' - 0"

**FIGURE 14—TO MAKE PART OF THE FAMILY CUTABLE, NEST IN A FAMILY IN A CUTABLE CATEGORY**

I think you know the answer… no this is not new.

You can see which categories are cutable in Object Styles.

## Nested Family Visibility (Shared vs Non-Shared)

When building the shower family, I had another challenge. When you cut through the shower and look in the opposite direction from the fixtures, they still show in elevations even though the fixtures are behind the cut plane of the section line.

This is another manifestation of the same issue described in the previous tip. Remember, the Plumbing Fixture category is non-cutable. This means that when it displays, the entire object will display (using the projection settings) regardless of whether it intersects the cut plane or not. In other words, the cut plane location is not considered for non-cutable elements. For non-cutable categories, if any part of the object intersects the view extents, the entire object displays.

The part that is potentially confusing is that the fixtures of the shower are a separate nested family. So you might be left to assume that the nested family would use its own extents to determine if it should display. This is not the case. That is unless the nested family is set to "Shared." In that case, it does use its own extents to determine if it displays! Confused yet?

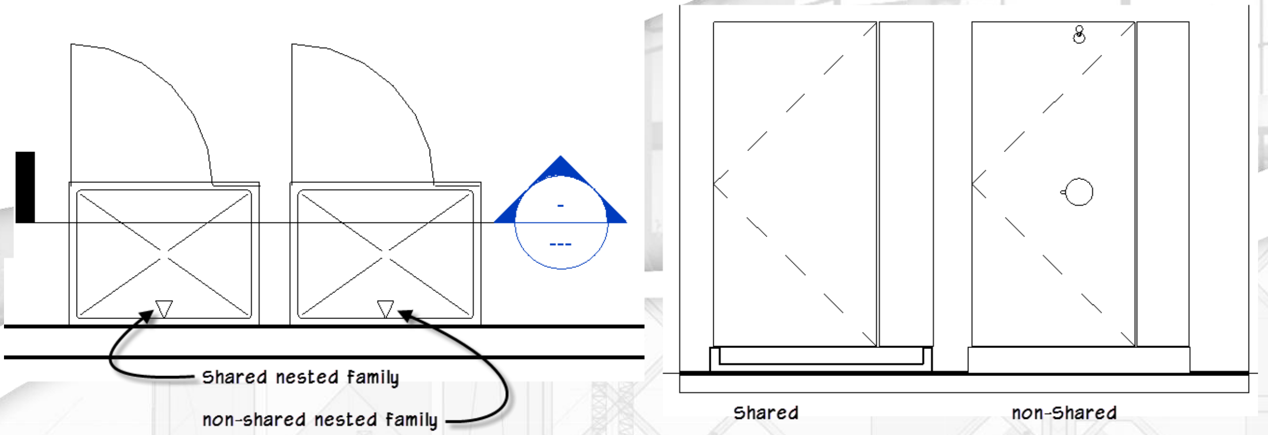


**FIGURE 15—SHARED NESTED FAMILIES WILL NOT DISPLAY IF THEY ARE BEHIND THE CUT PLANE**

The bottom line is, if you use nested families and if you want the extents of the nested family to be used when determining if it (the nested family) should display, then set the nested family to shared.

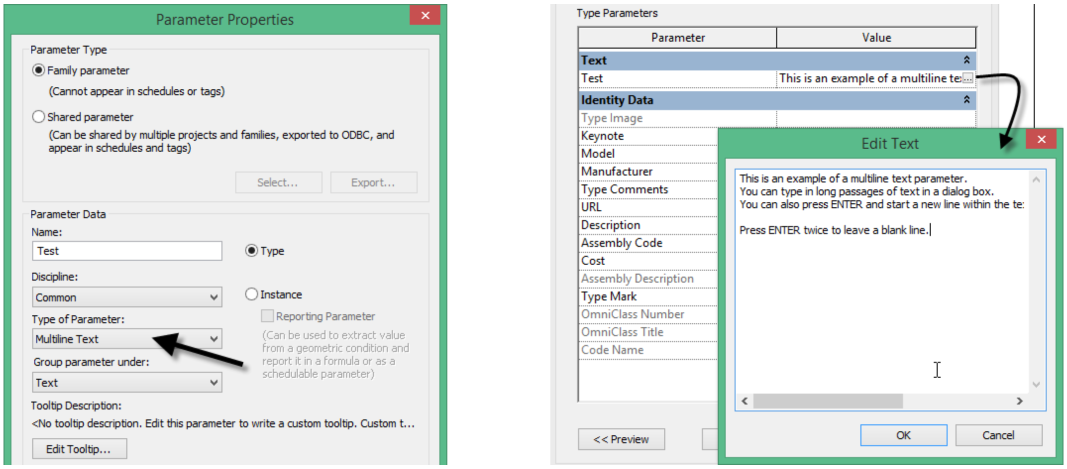
Is it new? Nah. But it is certainly useful…

# Parameters

Parameters are at the heart of every family. In this topic we'll look at several features and tips related to creating and managing parameters.

## Multiline Text

Perhaps the simplest parameter we can create is a Text parameter. A text parameter creates an empty field that can receive any input (numbers, letters or any combination). The Multiline Text parameter is very similar except that it displays a dialog box when you edit its value. In this dialog, you can type in any text and even include returns. Simply press ENTER to start a new line within the dialog (see Figure 16).



Parameter Properties

Parameter Type

● Family parameter

(Cannot appear in schedules or tags)

○ Shared parameter

(Can be shared by multiple projects and families, exported to ODBC, and appear in schedules and tags)

Parameter Data

Name: Test

● Type

Discipline: Common

○ Instance

□ Reporting Parameter

(Can be used to extract value from a geometric condition and report it in a formula or as a schedulable parameter)

Type of Parameter: Multiline Text

Group parameter under: Text

Tooltip Description:

<No tooltip description. Edit this parameter to write a custom tooltip. Custom t…

Type Parameters

| Parameter | Value |
|---|---|
| **Text** | |
| Test | This is an example of a multiline te: |
| **Identity Data** | |
| Type Image | |
| Keynote | |
| Model | |
| Manufacturer | |
| Type Comments | |
| URL | |
| Description | |
| Assembly Code | |
| Cost | |
| Assembly Description | |
| Type Mark | |
| OmniClass Number | |
| OmniClass Title | |
| Code Name | |

Edit Text

This is an example of a multiline text parameter.
You can type in long passages of text in a dialog box.
You can also press ENTER and start a new line within the te:

Press ENTER twice to leave a blank line.

**FIGURE 16—CREATE A MULTILINE TEXT PARAMETER TO ADD MULTIPLE LINES OF TEXT IN THE SAME FIELD**

What say you? Yes it's new.

## Image Parameters

Sometimes you want to display images in a schedule instead of just text and data. Any model element can reference an external image file (BMP, JPG, PNG or TIF). The image will be visible in the "Manage Images" dialog and if you add it to a schedule, it will show in the schedule when it is placed on a sheet. For example, perhaps you would like to show a photograph or illustration of each piece of furniture in a furniture schedule.

1. Open a project with furniture or create one.
2. Add a few Corbu chairs and then select them all. (Use: **Select All Instances > In Entire Project**).

⇨ On the Properties palette, click in the Image field and then click the small browse button that appears.

⇨ In the dialog that appears, click the Add button at the bottom.

3. Locate and image you want to use (such as *Corbu.png* file shown here) and then click Open.

⇨ Click OK to dismiss the "Manage Images" dialog.

This will only display the name of the image file on the Properties palette. To see the image, you have add this field to the schedule and then look at the schedule on a sheet.

4. Open a sheet containing your furniture schedule, or create one.

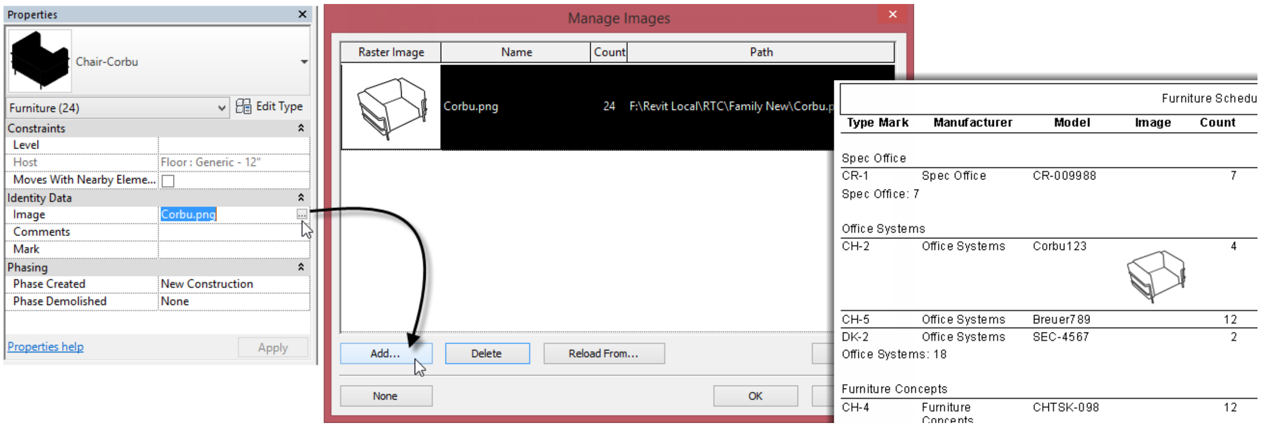
⇨ Double-click the schedule on the sheet (or right-click it and choose: **Edit Schedule**).

5. On the ribbon, on the Columns panel, click the Insert button.

⇨ Add the Image field to the schedule and then click OK.

⇨ In the top right corner of the schedule view, click the close icon to close the window.

The sheet should still be open behind this view and should now show the image column and the *Corbu.png* image that we added (see Figure 17).



| | |
|---|---|
| **Constraints** | |
| Level | |
| Host | Floor : Generic - 12" |
| Moves With Nearby Eleme... | □ |
| **Identity Data** | |
| Image | Corbu.png |
| Comments | |
| Mark | |
| **Phasing** | |
| Phase Created | New Construction |
| Phase Demolished | None |

| Raster Image | Name | Count | Path |
|---|---|---|---|
| | Corbu.png | 24 | F:\Revit Local\RTC\Family New\Corbu.p |

Furniture Schedu

| Type Mark | Manufacturer | Model | Image | Count |
|---|---|---|---|---|
| Spec Office | | | | |
| CR-1 | Spec Office | CR-009988 | | 7 |
| Spec Office: 7 | | | | |
| Office Systems | | | | |
| CH-2 | Office Systems | Corbu123 | | 4 |
| CH-5 | Office Systems | Breuer789 | | 12 |
| DK-2 | Office Systems | SEC-4567 | | 2 |
| Office Systems: 18 | | | | |
| Furniture Concepts | | | | |
| CH-4 | Furniture Concepts | CHTSK-098 | | 12 |

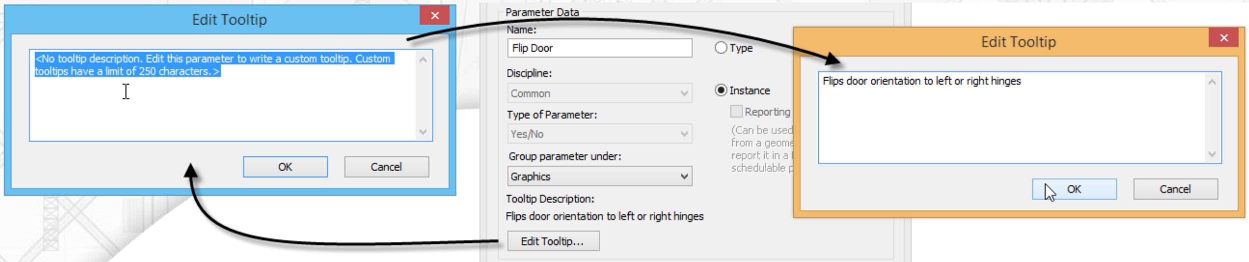
**FIGURE 17—ADD AN IMAGE TO SELECTED ELEMENTS AND THEN DISPLAY IT IN A SCHEDULE**

This is a very simple example using the built in instance-based Image property. There is also a Type Image property which would seem more appropriate in this example. However it takes a little more effort to set it up. You must edit the family in the family editor and add the Type Image there. You then save and reload the family back into the project and replace the existing when prompted. It is also possible to create additional Image parameters in the family editor. However, similar limitations on their use exist. So beyond the very simple example shown here, I having a hard time conceiving of too many practical uses.

So is it new? Yeah, just not super useful…

## Tooltip Description

Ever wish you could provide a little feedback on the purpose of a parameter to your end users? Maybe something a little more descriptive than the parameter name? Let's face it, it is not always easy to arrive at names for your parameters that make it completely clear to end users what their purpose is. Well adding tooltip descriptions can be just the thing you need. When defining the parameter, click the Edit Tooltip button to define a tooltip (see Figure 18).



Edit Tooltip

<No tooltip description. Edit this parameter to write a custom tooltip. Custom tooltips have a limit of 250 characters.>

Parameter Data

Name: Flip Door

○ Type

● Instance

□ Reporting

(Can be used from a geome report it in a schedulable p

Discipline: Common

Type of Parameter: Yes/No

Group parameter under: Graphics

Tooltip Description: Flips door orientation to left or right hinges

Edit Tooltip

Flips door orientation to left or right hinges

FIGURE 18—USE THE EDIT TOOLTIP BUTTON IN THE PARAMETER PROPERTIES DIALOG TO ADD A TOOLTIP

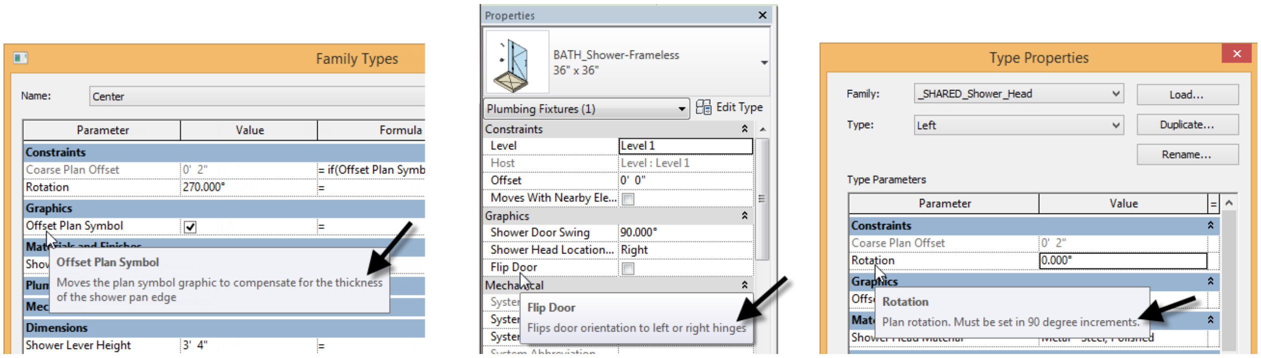
1. Open a family. Open the "Family Types" dialog and select a parameter to modify or create a new one.

2. In the "Parameter Properties" dialog, click the Edit Tooltip button.

⇨ There will be some text in the dialog already. Select all of this text and delete it. Then type in the text for the tooltip you want.

⇨ Click OK to complete the edit.

When you pause your mouse over the parameter name in the "Family Types" dialog, the tooltip will appear. It will also appear when the family is loaded into other families or projects (see Figure 19). You can see the tooltip when pausing your mouse over the parameter name on the Properties palette or in the "Type Properties" dialog (depending on whether the parameter is a type or instance property).



Family Types

Name: Center

| Parameter | Value | Formula |
|---|---|---|
| **Constraints** | | |
| Coarse Plan Offset | 0' 2" | = if(Offset Plan Symb |
| Rotation | 270.000° | = |
| **Graphics** | | |
| Offset Plan Symbol | ☑ | = |
| **Dimensions** | | |
| Shower Lever Height | 3' 4" | = |

Offset Plan Symbol

Moves the plan symbol graphic to compensate for the thickness of the shower pan edge

Properties

BATH_Shower-Frameless 36" x 36"

Plumbing Fixtures (1)

| | |
|---|---|
| **Constraints** | |
| Level | Level 1 |
| Host | Level : Level 1 |
| Offset | 0' 0" |
| Moves With Nearby Ele... | □ |
| **Graphics** | |
| Shower Door Swing | 90.000° |
| Shower Head Location... | Right |
| Flip Door | □ |
| **Mechanical** | |

Flip Door

Flips door orientation to left or right hinges

Type Properties

Family: _SHARED_Shower_Head

Type: Left

Type Parameters

| Parameter | Value |
|---|---|
| **Constraints** | |
| Coarse Plan Offset | 0' 2" |
| Rotation | 0.000° |
| **Graphics** | |

Rotation

Plan rotation. Must be set in 90 degree increments.

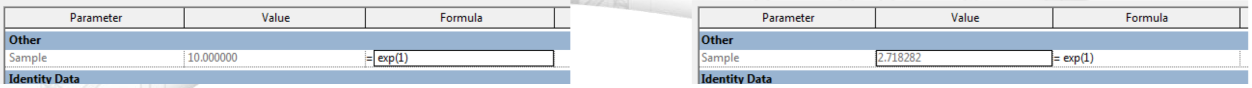
FIGURE 19—ADDING TOOLTIPS TO PARAMETERS CAN MAKE THEIR USE EASIER TO UNDERSTAND

According to the sample text that we delete first, the tooltip is limited to 250 characters. If you want tooltips to appear for shared parameters, they are added to the shared parameter definition and become part of the shared parameter file. Adding tooltips takes a little extra effort, but it can be a really nice finishing touch to add a degree of user friendliness and professionalism to your family content.

But is it new? Or did you just miss it? This one is new! In my tests, using the shared parameter file in previous versions worked OK, but a backup is advisable just the same.

## Formula Syntax

It is hard to play the game of "is that new" with this one. But if you use exponentiation or logarithms in your formulas, there are some syntax changes that can affect your work and any existing families you might have. Previously, the formula: exp(X) would be interpreted as: 10^X. That is ten raised to the power of X. Now, it is interpreted as: e^X.  That is the mathematical constant "e" raised to the power of X.



| Parameter | Value | Formula |
|---|---|---|
| **Other** | | |
| Sample | 10.000000 | = exp(1) |
| **Identity Data** | | |

| Parameter | Value | Formula |
|---|---|---|
| **Other** | | |
| Sample | 2.718282 | = exp(1) |
| **Identity Data** | | |

**FIGURE 20—PREVIOUS USE OF EXP ON THE LEFT AND CURRENT USE ON THE RIGHT**

Another change is that you can you can represent natural logarithm

(logarithm base e) in formulas as: ln(X).

Is that new? Well duh…

## Reporting Annotation

Here's a little bonus one. If you add dimension parameters to an generic annotation family, you can actually report the value of the dimension in a label within the same family. For example if you have a parameter that drives the size of the graphic used in the generic annotation, you can have a label in that same family that tells you how big the family is. I was going to include this in the paper, but I decided against it for two reasons.

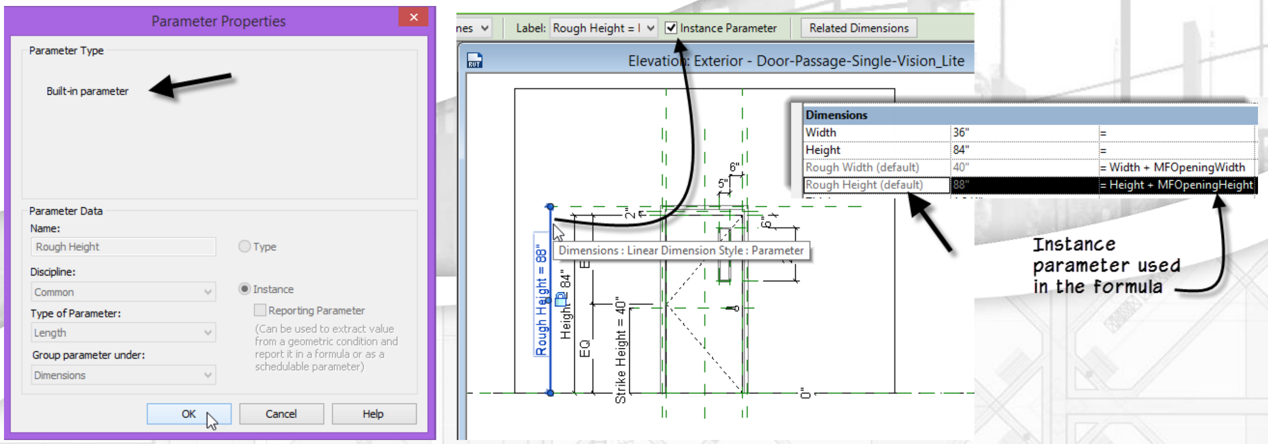
First, I detailed this pretty thoroughly at my blog here:

http://paulaubin.com/blog/generic-annotations-in-revit/.

Second, sadly I still have not really found a practical use for this.

## Modify Built-in Parameters

Have you ever created a family from a template that already contains parameters? For example, if you use the *Casework.rft* or the *Door.rft* templates you will find they contain some basic dimensions already like Width and Depth for casework and Width, Height, Thickness, Rough Width and Height for doors. If you try to modify one of these parameters in "Family Types" all of its properties will be grayed out and a message will display indicating that it is a "Built-in parameter" (see Figure 21).



Parameter Properties

Parameter Type

Built-in parameter

Parameter Data

Name: Rough Height

○ Type

Discipline: Common

● Instance

Type of Parameter: Length

□ Reporting Parameter

(Can be used to extract value from a geometric condition and report it in a formula or as a schedulable parameter)

Group parameter under: Dimensions

Label: Rough Height = I ☑ Instance Parameter

Elevation: Exterior - Door-Passage-Single-Vision_Lite

| Dimensions | | |
|---|---|---|
| Width | 36" | = |
| Height | 84" | = |
| Rough Width (default) | 40" | = Width + MFOpeningWidth |
| Rough Height (default) | 88" | = Height + MFOpeningHeight |

Instance parameter used in the formula

**FIGURE 21—BUILT-IN PARAMETERS CAN BE CHANGED FROM TYPE TO INSTANCE (AND VICE-VERSA) ON THE OPTIONS BAR**

While it is true that most settings of a built-in parameter cannot be modified (it can't be renamed, you can't change the type or its grouping) you can actually change it from a type to an instance parameter. Consider doors as an example. You might want to use formulas to calculate the Rough Width and Rough Height values based on other settings and parameters. However, since these default to type parameters, you would need to make sure you only used type parameters in the formula (instance parameters cannot drive type parameters). So if you are using instance parameters in the formula, what do you do? You can change the Rough Width and Height parameters to instance parameters and then use other instance parameters in their formulas. To do this:

1. Select a dimension onscreen that is labeled with the parameter you want to change.

⇨ If you don't have one, add a couple reference planes off to the side and dimension them. Then label this dimension with the Rough Height (or Width) parameter.

2. With the labeled dimension selected, on the Options Bar, check the Instance Parameter checkbox (see the middle of the figure).

   Back in "Family Types" the parameter will now be an instance parameter as indicated by including: **(default)** after the name.

⇨ If you don't need the dimension in the model, you can delete it onscreen and the parameter will remain an instance parameter.

Perhaps you're saying: "That's not new!" Well, right you are. Changing a built-in parameter from type to instance with the checkbox on the Options Bar is not new.

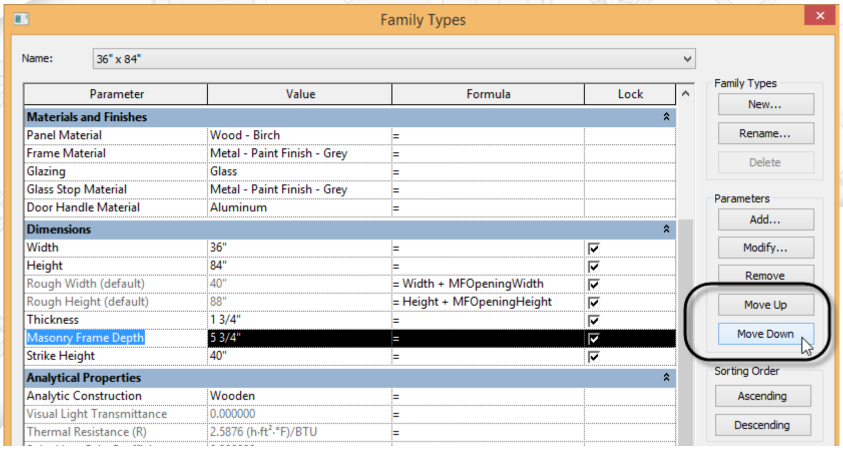
But

In a little twist, being able to click the Modify button in "Family Types" and having it display "Built-in Parameter" **is** new. Subtle for sure. But definitely new.

**Bonus:** this procedure was used (for Rough Width and Height) in the new out-of-the-box door content provided with 2016.

## Reordering Parameters

So you're building that complex family containing lots of custom parameters. And you've noticed that if you add the parameters in just the right sequence, they may actually end up displaying in the "Family Types" dialog and on the Properties palette the way that you want. But then you have to add just one more parameter… and wouldn't you know it appears in a completely random location. Very frustrating… But wait, why wouldn't you just use the **Move Up** and **Move Down** buttons? See Figure 22.



Family Types

Name: 36" x 84"

| Parameter | Value | Formula | Lock |
|---|---|---|---|
| **Materials and Finishes** | | | |
| Panel Material | Wood - Birch | = | |
| Frame Material | Metal - Paint Finish - Grey | = | |
| Glazing | Glass | = | |
| Glass Stop Material | Metal - Paint Finish - Grey | = | |
| Door Handle Material | Aluminum | = | |
| **Dimensions** | | | |
| Width | 36" | = | ☑ |
| Height | 84" | = | ☑ |
| Rough Width (default) | 40" | = Width + MFOpeningWidth | ☑ |
| Rough Height (default) | 88" | = Height + MFOpeningHeight | ☑ |
| Thickness | 1 3/4" | = | ☑ |
| Masonry Frame Depth | 5 3/4" | = | ☑ |
| Strike Height | 40" | = | ☑ |
| **Analytical Properties** | | | |
| Analytic Construction | Wooden | = | |
| Visual Light Transmittance | 0.000000 | = | |
| Thermal Resistance (R) | 2.5876 (h·ft²·°F)/BTU | = | |

FIGURE 22—ARRANGE PARAMETERS IN WHATEVER ORDER YOU LIKE WITH THE MOVE UP AND DOWN BUTTONS

Wait a minute! That **HAS** to be new?
Yup, sure it!

## Sorting Parameters

Or perhaps you'd prefer to just sort them alphabetically instead? Using the Sorting Order buttons directly beneath Move Up and Down, you can sort all of the parameters in the list in Ascending or Descending order. (Also in Figure 22).

Yes, I am sure you guessed that this one is new too.

## Invisible Shared Parameters

This is one of my favorites! Ever wanted to hide a parameter? Maybe you are creating a custom family and you have some detailed nested formulas. Parameter A is used in the formula of parameter B, which might in-turn drive parameter C. Wouldn't it be nice to not have to rely on just stashing the parameter in the "Other" grouping and hoping that your users don't mess with it? Well, guess what? You can make a parameter that shows in the family environment, can be used to drive formulas, but is hidden in the project environment!

To do this, you just have to perform a simple manual edit to your shared parameter file. Let's take a look at the procedure.

Make sure you have a Shared Parameter file and that it contains at least one group. I like to create a special group just for my invisible parameters that I call "Invisible Parameters." Further, I always name my invisible parameters with the prefix: INV_

There are two important steps to creating an invisible parameter. First, you have to make the parameter invisible *before* you add it to a family. Second, you have to make it invisible outside of Revit by manually editing the shared parameter TXT file. Therefore you might want to begin in a project instead of a family.

1. Open a project file (or create a temporary one) and create one or more shared parameters (Manage tab).

   For this example, I'll make a Yes/No parameter.

2. Repeat for as many invisible parameters you need to create.

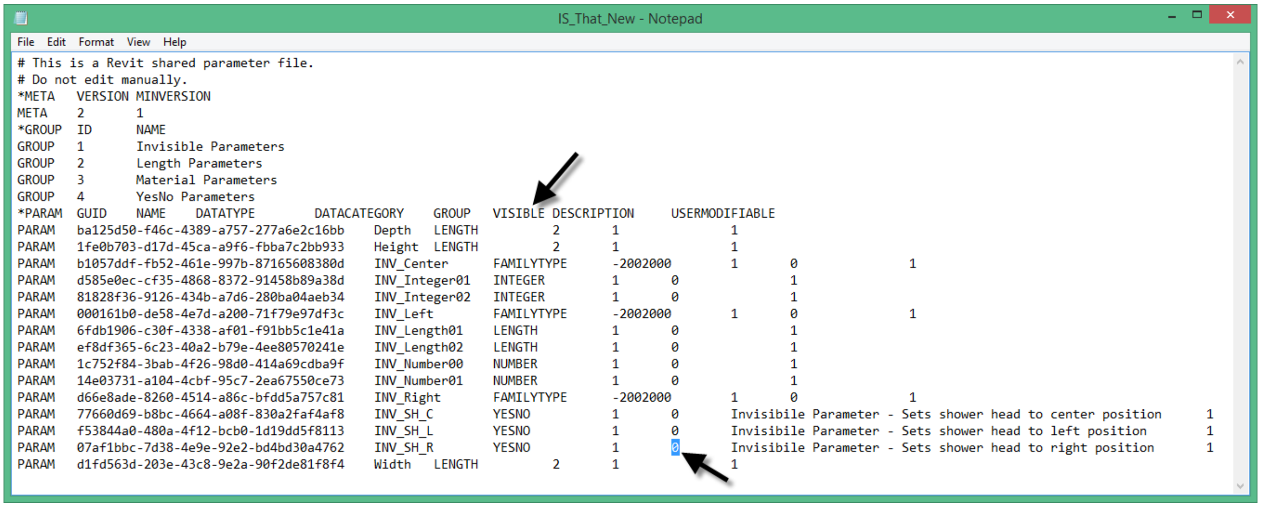
⇨ Close the project and do not save it.

3. Run Windows Notepad, browse to and open the shared parameter file.

The first two lines in the file will read:

> \# This is a Revit shared parameter file.
>
> \# Do not edit manually.

Well, we are going to ignore that. The group information comes first. Skip over that. Next you will see column headers. Look for the position of the VISIBLE column. This is the one we need to edit. Depending on your Revit version, VISIBLE might be the last column or the third column from the right. VISIBLE is a yes/no toggle and take values 0 and 1. Zero means the parameter is invisible and one means it is visible. All parameters default to: 1. Simply change any that you want to make invisible to: 0 (see Figure 23).



```
# This is a Revit shared parameter file.
# Do not edit manually.
*META   VERSION MINVERSION
META    2       1
*GROUP  ID      NAME
GROUP   1       Invisible Parameters
GROUP   2       Length Parameters
GROUP   3       Material Parameters
GROUP   4       YesNo Parameters
*PARAM  GUID    NAME    DATATYPE        DATACATEGORY    GROUP   VISIBLE DESCRIPTION     USERMODIFIABLE
PARAM   ba125d50-f46c-4389-a757-277a6e2c16bb    Depth   LENGTH          2       1               1
PARAM   1fe0b703-d17d-45ca-a9f6-fbba7c2bb933    Height  LENGTH          2       1               1
PARAM   b1057ddf-fb52-461e-997b-87165608380d    INV_Center      FAMILYTYPE      -2002000        1       0               1
PARAM   d585e0ec-cf35-4868-8372-91458b89a38d    INV_Integer01   INTEGER         1       0               1
PARAM   81828f36-9126-434b-a7d6-280ba04aeb34    INV_Integer02   INTEGER         1       0               1
PARAM   000161b0-de58-4e7d-a200-71f79e97df3c    INV_Left        FAMILYTYPE      -2002000        1       0               1
PARAM   6fdb1906-c30f-4338-af01-f91bb5c1e41a    INV_Length01    LENGTH          1       0               1
PARAM   ef8df365-6c23-40a2-b79e-4ee80570241e    INV_Length02    LENGTH          1       0               1
PARAM   1c752f84-3bab-4f26-98d0-414a69cdba9f    INV_Number00    NUMBER          1       0               1
PARAM   14e03731-a104-4cbf-95c7-2ea67550ce73    INV_Number01    NUMBER          1       0               1
PARAM   d66e8ade-8260-4514-a86c-bfdd5a757c81    INV_Right       FAMILYTYPE      -2002000        1       0               1
PARAM   77660d69-b8bc-4664-a08f-830a2faf4af8    INV_SH_C        YESNO           1       0       Invisibile Parameter - Sets shower head to center position     1
PARAM   f53844a0-480a-4f12-bcb0-1d19dd5f8113    INV_SH_L        YESNO           1       0       Invisibile Parameter - Sets shower head to left position       1
PARAM   07af1bbc-7d38-4e9e-92e2-bd4bd30a4762    INV_SH_R        YESNO           1       0       Invisibile Parameter - Sets shower head to right position      1
PARAM   d1fd563d-203e-43c8-9e2a-90f2de81f8f4    Width   LENGTH          2       1               1
```

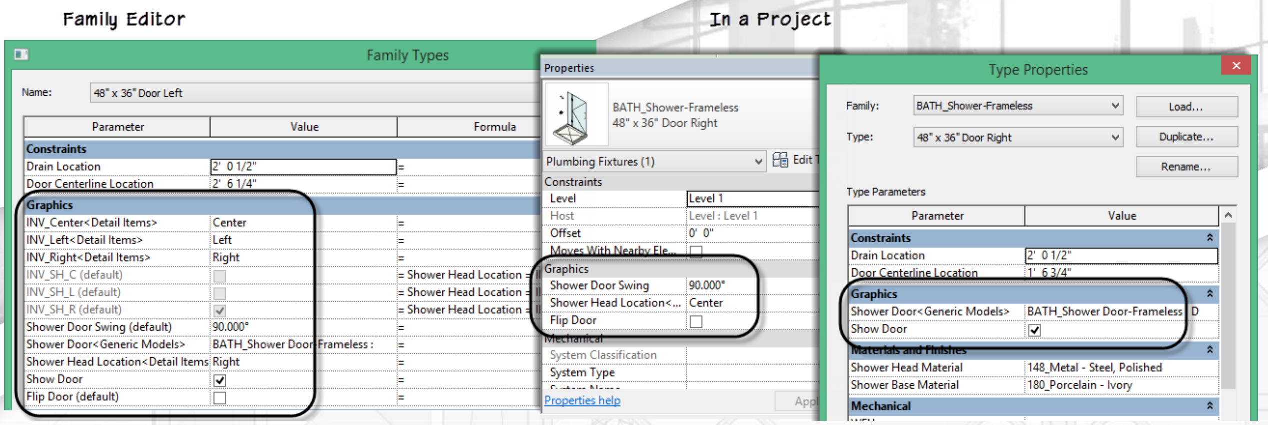
**FIGURE 23—SET PARAMETERS YOU WANT INVISIBLE TO ZERO IN THE VISIBLE COLUMN**

4. When finished, close and save the shared parameter file.

Limit your edits to the VISIBLE column only and you will not risk breaking the shared parameters in the file. Avoid editing the GUID or parameter names.

5. Now you can open a family, and add this parameter(s) to the family.

6. Finish the family and then load it into a project.



Family Types

Name: 48" x 36" Door Left

| Parameter | Value | Formula |
|---|---|---|
| **Constraints** | | |
| Drain Location | 2' 0 1/2" | = |
| Door Centerline Location | 2' 6 1/4" | = |
| **Graphics** | | |
| INV_Center<Detail Items> | Center | = |
| INV_Left<Detail Items> | Left | = |
| INV_Right<Detail Items> | Right | = |
| INV_SH_C (default) | □ | = Shower Head Location = |
| INV_SH_L (default) | □ | = Shower Head Location = |
| INV_SH_R (default) | ☑ | = Shower Head Location = |
| Shower Door Swing (default) | 90.000° | = |
| Shower Door<Generic Models> | BATH_Shower Door-Frameless : | = |
| Shower Head Location<Detail Items | Right | = |
| Show Door | ☑ | = |
| Flip Door (default) | □ | = |

Properties

BATH_Shower-Frameless
48" x 36" Door Right

Plumbing Fixtures (1) Edit T

| | |
|---|---|
| Constraints | |
| Level | Level 1 |
| Host | Level : Level 1 |
| Offset | 0' 0" |
| Moves With Nearby Ele... | □ |
| Graphics | |
| Shower Door Swing | 90.000° |
| Shower Head Location<... | Center |
| Flip Door | □ |
| Mechanical | |
| System Classification | |
| System Type | |

Properties help

Type Properties

Family: BATH_Shower-Frameless

Type: 48" x 36" Door Right

Load... Duplicate... Rename...

Type Parameters

| Parameter | Value |
|---|---|
| **Constraints** | |
| Drain Location | 2' 0 1/2" |
| Door Centerline Location | 1' 6 3/4" |
| **Graphics** | |
| Shower Door<Generic Models> | BATH_Shower Door-Frameless D |
| Show Door | ☑ |
| **Materials and Finishes** | |
| Shower Head Material | 148_Metal - Steel, Polished |
| Shower Base Material | 180_Porcelain - Ivory |
| **Mechanical** | |

**FIGURE 24—IN THE FAMILY EDITOR (LEFT) THE PARAMETERS SHOW, BUT IN THE PROJECT, THEY ARE INVISIBLE (RIGHT)**

I know that you might be nervous about manually editing your shared parameter file. Well of course you will want to make a backup of it first. But there is a terrific tool available from CAD Technology Center (CTC) called: Shared Parameter Manager. This tool is part of the CTC BIM Manager Suite and will allow you to perform many tasks on your shared parameter files including making them invisible without your needing to manually edit the file. CTC has a booth here at the conference. Why not check them out?

> New, not New? New!… well not exactly. Making parameters invisible is not new. But the structure of the shared parameter file has changed, specifically moving the VISIBLE column from the end to third from the right… so that little bit is new… sort of.

## Make a "List" Parameter

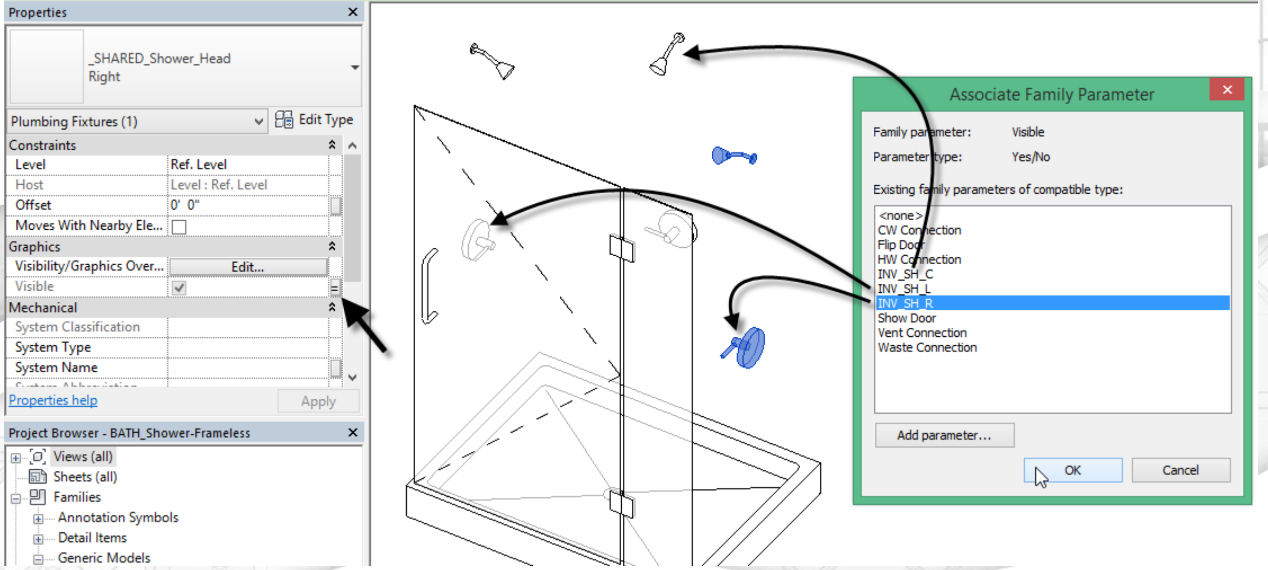
And now for my favorite item in the collection. I have always wanted Revit to add a "list" parameter. By this I mean a parameter that will give a drop-down list of choices. Well, read on! Because by following this procedure you can! For this example, I will continue with the shower family discussed elsewhere and add three possible mounting positions for the shower head and controls. You can use this technique any time you need three or more choices for a particular item. However, it does start to become a bit impractical if you need more than say five or six items, but your results may vary.

1. Create or open a family. Add three "optional" items that you want to control with a list.

   In this example, the show head will be copied to make three copies and positioned for left, right and center mounting options. We then want a list where the user can choose from Left, Right or Center.

2. Select each instance and associate a family parameter to the Visible setting (see Figure 25).

I used invisible parameters for these (see the previous tip). This is not required, but I think it ultimately makes a nicer solution. This is because you will not have to worry about users messing with the visibility parameters or being confused by them.



Properties

_SHARED_Shower_Head
Right

Plumbing Fixtures (1)

Constraints

| | |
|---|---|
| Level | Ref. Level |
| Host | Level : Ref. Level |
| Offset | 0' 0" |
| Moves With Nearby Ele... | □ |

Graphics

| | |
|---|---|
| Visibility/Graphics Over... | Edit... |
| Visible | ☑ |

Mechanical

| | |
|---|---|
| System Classification | |
| System Type | |
| System Name | |

Project Browser - BATH_Shower-Frameless

- Views (all)
- Sheets (all)
- Families
  - Annotation Symbols
  - Detail Items
  - Generic Models

Associate Family Parameter

Family parameter: Visible

Parameter type: Yes/No

Existing family parameters of compatible type:

- <none>
- CW Connection
- Flip Door
- HW Connection
- INV_SH_C
- INV_SH_L
- INV_SH_R
- Show Door
- Vent Connection
- Waste Connection

FIGURE 25—ASSIGN A VISIBILITY PARAMETER TO EACH INSTANCE

Now that we have the three instances each controlled by its own Yes/No visibility parameter, it is time to move on and create the list. This will be achieved with <Family Types> parameters. We have three choices. So ultimately we will need four <Family Types> parameters. One for the list, and three used for "comparison" by formulas. This is how it works:

We make a <Family Types> parameter for each condition: Left, Right and Left. You set the value of each one permanently to these values. Then you have a fourth one that users can manipulate. In that fourth one, is where the list will appear. By choosing one of the options from the list, the formulas will do a comparison and thereby determine which item the chose and set the visibility parameters accordingly. I recommend invisible parameters for the comparison <Family Types> parameters. This will greatly reduce the possibility that an end user will inadvertently break the family.

One last point. It doesn't really matter what category of <Family Types> parameter you use. I recommend something that is not likely to be used elsewhere in the family. I chose Detail Items for this example, but it can really be anything.

3. Create a new family from the *Detail Item.rft* template and save it as: **Right**. You do not need to add any geometry to it.

4. Repeat creating two more: **Left** and **Center**.

⇨ Load all three of these into your family.

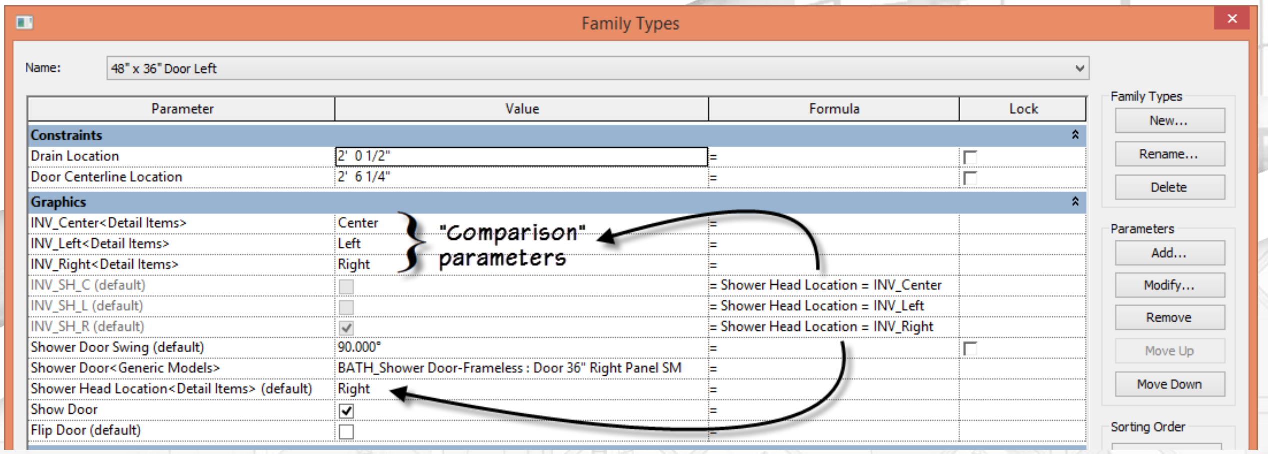
5. Following the procedure outlined in the previous tip, create three invisible: Family Types:Detail Items shared parameters. I called mine: INV_Right, INV_Left and INV_Center.

   **Important:** Remember to make them invisible by editing the TXT file before adding them to the family.

6. Add each of the invisible shared parameters to the family.

⇨ Set INV_Right to: Right. INV_Left to: Left and INV_Center to: Center (see Figure 26).

These need to be set permanently to these values, or it will not work. That is why I recommend invisible parameters. You easily set it and forget it that way.



Name: 48" x 36" Door Left

| Parameter | Value | Formula | Lock |
|---|---|---|---|
| **Constraints** | | | |
| Drain Location | 2' 0 1/2" | = | ☐ |
| Door Centerline Location | 2' 6 1/4" | = | ☐ |
| **Graphics** | | | |
| INV_Center<Detail Items> | Center | = | |
| INV_Left<Detail Items> | Left | = | |
| INV_Right<Detail Items> | Right | = | |
| INV_SH_C (default) | ☐ | = Shower Head Location = INV_Center | |
| INV_SH_L (default) | ☐ | = Shower Head Location = INV_Left | |
| INV_SH_R (default) | ☑ | = Shower Head Location = INV_Right | |
| Shower Door Swing (default) | 90.000° | = | ☐ |
| Shower Door<Generic Models> | BATH_Shower Door-Frameless : Door 36" Right Panel SM | = | |
| Shower Head Location<Detail Items> (default) | Right | = | |
| Show Door | ☑ | = | |
| Flip Door (default) | ☐ | = | |

"Comparison" parameters

**FIGURE 26—SET UP FAMILY TYPES TO EVALUATE THE CHOICE FROM SHOWER HEAD LOCATION AND USE IT TO SET VISIBILITY**

Almost there. We need one more parameter. This will be another <Family Types> that is not invisible. This is the one that we want the user to choose from.

7. Create another Family Types:Detail Items parameter and call it: **Shower Head Location**.

   This one is NOT invisible.

All four <Family Types> parameters will show a list of the three detail items we loaded into the file. The three invisible ones we permanently set to the values we needed as indicated above. When the use chooses one of the three values for Shower Head Location, it will make that parameter equal to one of the three invisible comparison parameters. Using simple formulas, we can have this condition evaluated by each of the three Yes/No visiblity parameters.

8. For the INV_SH_C parameter, input: **Shower Head Location = INV_Center** in the formula field.

⇨ For the INV_SH_L parameter, input: **Shower Head Location = INV_Left** in the formula field.

⇨ For the INV_SH_R parameter, input: **Shower Head Location = INV_Right** in the formula field.

9. That's it. Save the family and load it into a project to test.

We need the comparison parameters because we cannot specify the value we are looking for directly in the formula. So instead, we are simply asking: is the value the user chose equal to this first one? No? how about the second or third? When it finds the one that is equal, it hides the others.

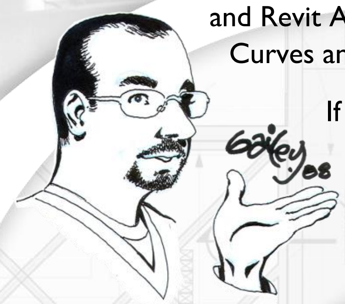
So there you have it! You can make list parameters after all! It is a little tedious I grant you that. But I find it incredibly powerful. However, I would certainly prefer if the factory just adds a proper list parameter sometime soon. Then I'll be able to say:

> Is that new? YES! But until then, no this one is not new, but I think it is way cool just the same!

## Further Study

You can find more information and tutorials in my books and video training. Please visit my website at: www.paulaubin.com for more information on my books.

I also have Revit video training available at: www.lynda.com/paulaubin. I have several courses at lynda.com including: Revit Essential Training, Revit Family Editor and Revit Architecture Rendering, Advanced Modelling in Revit Architecture, Formulas and Curves and many more.



If you have any questions about this session or Revit in general, you can use the contact form at www.paulaubin.com to send me an email.

Follow me on twitter: **@paulfaubin**

**Thank you for attending. Please fill out your evaluation.**