AS121828

Global Parameters, Global Control: Revit Global Parameters in Practice

Paul F. Aubin
Paul F. Aubin Consulting Services, Inc.

Learning Objectives

- Learn how to set up a global parameter to drive length or materials and more
- Learn how to apply global parameters to labeled dimensions in the project environment
- Learn how to associate global parameters to several families and flex them simultaneously
- Learn how to use global parameters to drive radius and diameter dimensions

Description

Global parameters (GP) bring the power of the Family Editor into the project environment, letting you label dimensions and drive parameter values directly in the project environment. Imagine being able to drive an offset distance between elements in multiple locations around the entire project, or drive instance parameters of several independent families from a single control panel. These are the kinds of things that are possible with global parameters. In this session, we'll walk through several scenarios using GP to establish relationships in your projects. We'll explore using them for establishing critical design dimensions and helping with design exploration. We'll also look at how GP can make your content even more powerful by letting you control several separate families at once from a single parameter, without needing to embed the families into one another first. If you want to explore the exciting global parameters feature, then this is the session for you.

Speaker

Paul F. Aubin is the author of many CAD and BIM book titles including the widely acclaimed: The Aubin Academy Mastering Series and his "deep dive" into the Revit family editor: *Renaissance Revit*. He is also the author of dozens of Revit video training titles on **LinkedIn Learning** (powered by **lynda.com** content) Paul covers all aspects of Revit from beginning to advanced. Paul is an independent architectural consultant who travels internationally providing Revit content creation, implementation, training and support services. His involvement in the architectural profession spans over 27 years, with experience that includes design, production, CAD management, coaching and training. He is an active member of the Autodesk user community, and has been a top-rated speaker at Autodesk University, the Revit Technology Conference (now BILT), Midwest University (MU) and the BIM workshops for many years. He recently gave the keynote address to the Campus FM Technology Association (CFTA) annual conference. 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, an Autodesk Expert Elite and an Autodesk Certified Professional. He lives in Chicago with his wife and their three children currently attend universities around the country.

Introduction

Global parameters are created in the project environment and can be used to drive the instance or type values of inserted families and can also be used to label dimensions thereby providing constraints that can be flexed from a single location. This paper was written with Revit 2018. If you are using an older version, some of the features shown herein may not apply to you.

Global parameters work well for driving multiple objects or dimensions that must all maintain the same value. They are similar to other constraints like locked dimensions and equality constraints but are more flexible. This is because unlike locked dimensions that must be unlocked to be changed and which are individual constraints that are not coordinated with other locked dimensions, global parameters can be applied to multiple elements and can be flexed from the “Global Parameters” dialog and will thereby update throughout the project. They also offer an alternative to equality dimensions and provide the flexibility to be applied to discontinuous elements; something equality dimensions cannot do. They can also be used to drive the instance and type parameters of many families.

The lessons that follow highlight some of these benefits across several examples.

Setting a common hold dimension

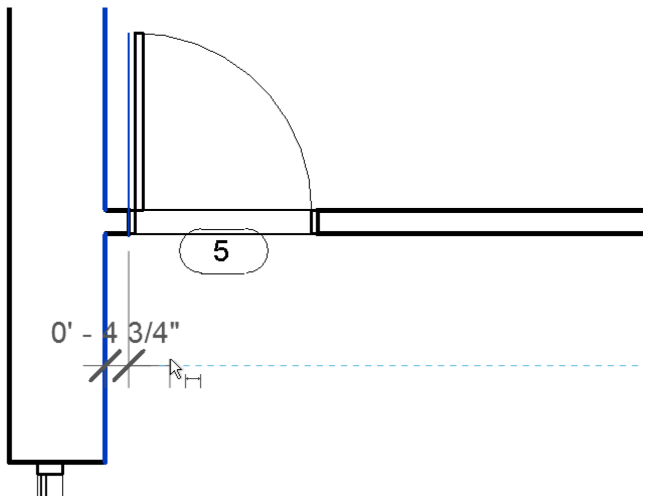
A very basic use of a global parameter is to define a simply “hold” dimension. This can be useful whenever you have a situation where a particular dimension value is critical to the design in some way. This might be to maintain a design relationship or to adhere to a building code requirement.

These kinds of things can be done with locked dimensions. But if you need to change them after locking, you would need to unlock and modify each one. A global parameter gives the same constraint potential, but is much easier to modify.

Setting a “standard” door jamb

In this example, we will establish a common jamb dimension for the doors across a floor plan.

1. Open the file named: *01_Door Jambs_!Start.rvt*.
2. Create an aligned dimension between the face of wall and the left edge of the door on the left.

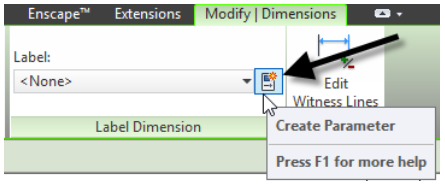


3. Repeat for the other doors.

There are only five doors and it is easy enough to create these dimensions, but if you would rather skip, here's a catch-up file:

CATCH UP! You can open the file completed to this point named: ***01_Door Jambs_A.rvt***.

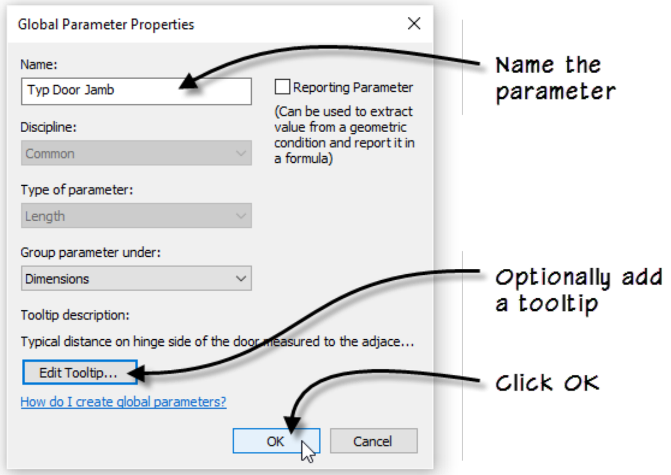
4. Select the first dimension that you created.
5. On the Modify | Dimensions tab, on the Label Dimension panel, click the small Create Parameter icon.



If you have worked in the family editor before, this is like labelling a dimension in a family. All we need to do is define the new parameter.

6. Input the name for the parameter such as: **Typ Door Jamb**.
7. Accept the default "group under" and optionally add a tooltip.

A tooltip provides a more detailed reminder of what the parameter is intended for. You will see the tip when you pause over the name of the parameter in dialogs. While it is not required, it is recommended.


Global Parameter Properties

Name: Typ Door Jamb

Reporting Parameter (Can be used to extract value from a geometric condition and report it in a formula)

Discipline: Common

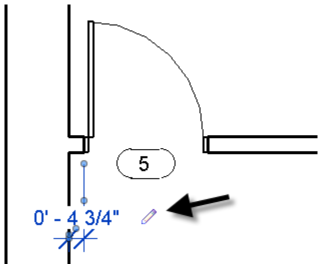
Type of parameter: Length

Group parameter under: Dimensions

Tooltip description: Typical distance on hinge side of the door measured to the adjace...

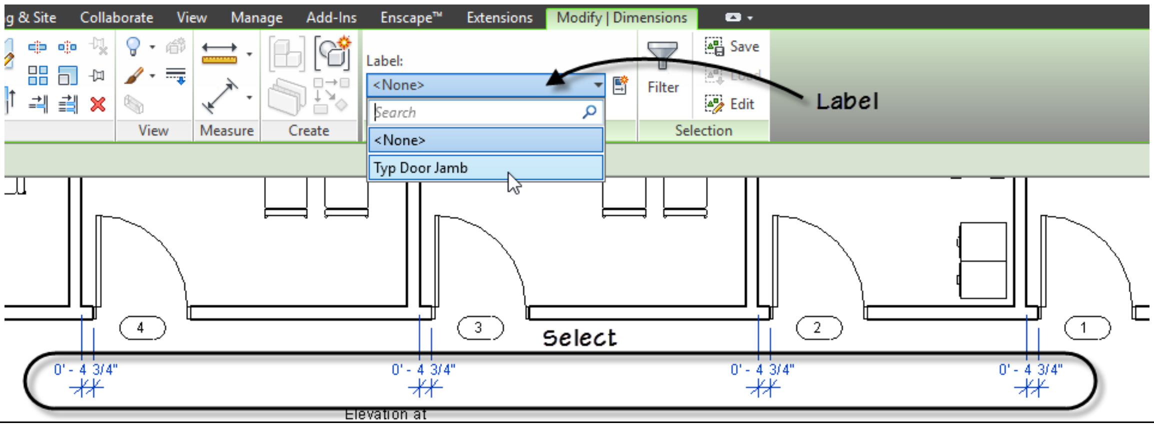
8. Click OK to finish defining the parameter.

Whenever the labelled dimension is selected, it will now display a small pencil icon next to it to indicate that it is labelled.



Next you can apply this same label to the other dimensions.

9. Select all the other four dimensions added above.
10. On the ribbon, from the label drop-down, choose: **Typ Door Jamb**.

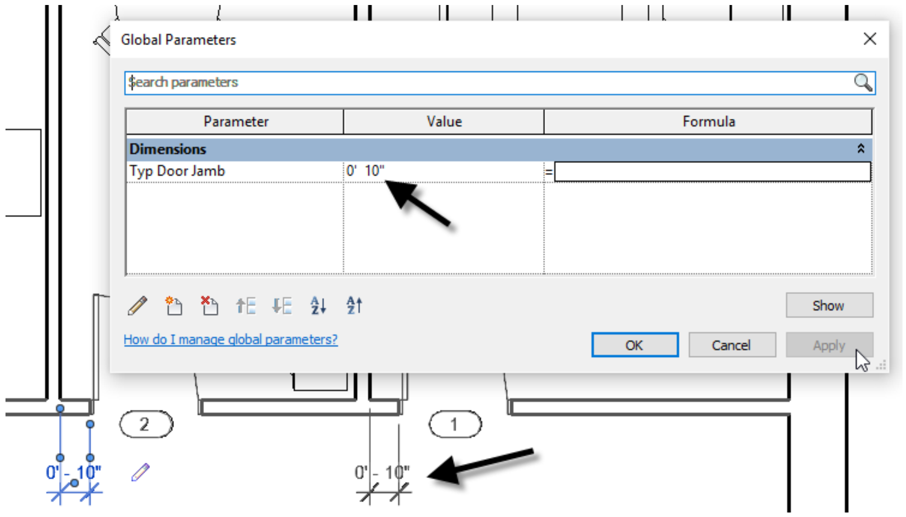


CATCH UP! You can open the file completed to this point named: ***01_Door Jambs_B.rvt***.

Now we can flex this parameter to test it out. There are two ways to do this. The simplest way to access the “Global Parameters” dialog and edit is to select one of the dimensions, and then click the small pencil icon. The other way is to use the Global Parameters button on the Manage tab of the ribbon.

11. Select one of the dimensions, and then click the small pencil icon.

12. In the “Global Parameters” dialog, change the value of Typ Door Jamb to: **10"** and then click Apply.



Parameter	Value	Formula
Dimensions		
Typ Door Jamb	0' 10"	=

All the doors will adjust to reflect the change. That’s it; your first global parameter. This is admittedly a very simple example, but it covers the overall steps. But we have many other examples to explore. Before we leave this file, let’s look at one other aspect.

Reveal Constraints

Labelled dimensions will behave like other constraints in the model. Therefore, reveal constraints is a very useful tool when used with global parameters.

13. Click OK to close the “Global Parameters” dialog.

14. Delete one of the dimensions.

A warning will appear. (*Note that it is the same warning you would see when deleting a locked or equality dimension*).

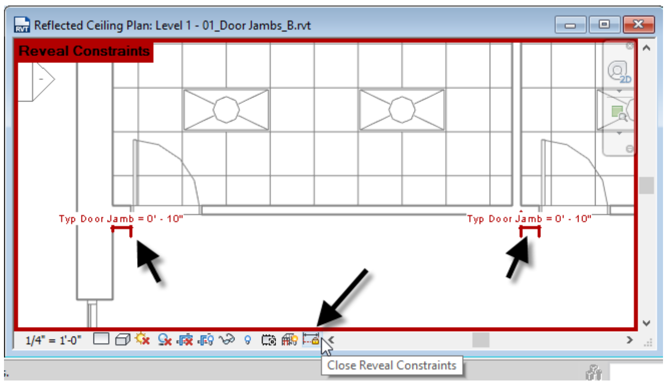
15. Cancel the warning.

16. On the Project Browser, open the *Level 1* reflected ceiling plan view.

17. On the View Control Bar, click the Reveal Constraints icon.

Notice that the labelled dimensions appear as constraints in this view.

18. Close the file (CTRL + W) there is no need to save.



Reporting Parameters

Using the concepts from the previous example, we can get slightly more advanced and use a global parameter to help us identify places in our model that do not meet minimum code requirements. To do this, we will leverage three features: first we will set up a new global parameter as a “reporting parameter.” Second we will create a formula to test the value of the reporting parameter against the code rule and finally we will set up a filter in the view to display the results. (*Note: This example is a “proof of concept” type example, but is slightly impractical for regular usage*).

Check for code compliance

Let’s start by adding a new global parameter in the location where we need to check for code compliance.

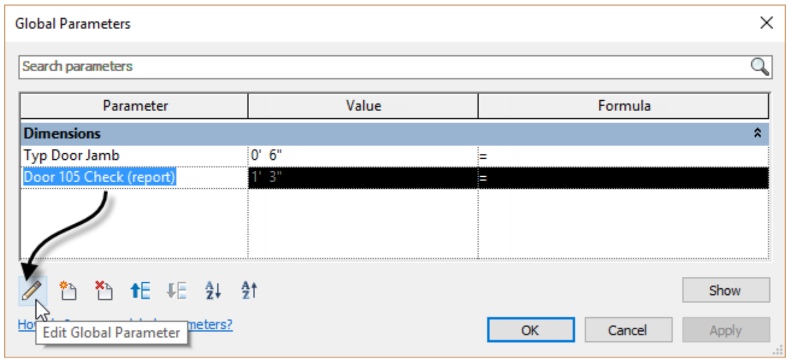
1. Open the file named: *02_Door Check_!Start.rvt*.
2. Select the dimension on the right side of door 1 (currently 1'-3").
3. On the Modify | Dimensions tab, on the Label Dimension panel, click the Create Parameter icon. Name it: **Door 105 Check**. Optionally add a tooltip and click OK.

This parameter is now like the other one we created and if you try to flex it, it will create a warning. Go ahead and try it… This is because we are currently constraining both sides of the door.

To remedy this, we need to change our new parameter to a “Reporting” parameter. Like the reporting parameters we can create in the family editor, a reporting global parameter will not drive geometry, but rather will simply “report” the value of the labelled dimension. In the context of a project, you may wonder why we would need this and how it differs from a simple non-labelled dimension? Onscreen, there will not appear to be much difference. When geometry moves, this dimension will update.

The difference is that we can use reporting parameters in formulas.

4. On the Manage tab of the ribbon, click the Global Parameters button.
5. Select *Door 105 Check* and then click the small edit icon.


Global Parameters

Search parameters

Parameter	Value	Formula
Dimensions		
Typ Door Jamb	0' 6"	=
Door 105 Check (report)	1' 3"	=

6. In the "Global Parameter Properties" dialog, check the Reporting Parameter checkbox and then click OK.
7. Flex the *Typ Door Jamb* parameter.

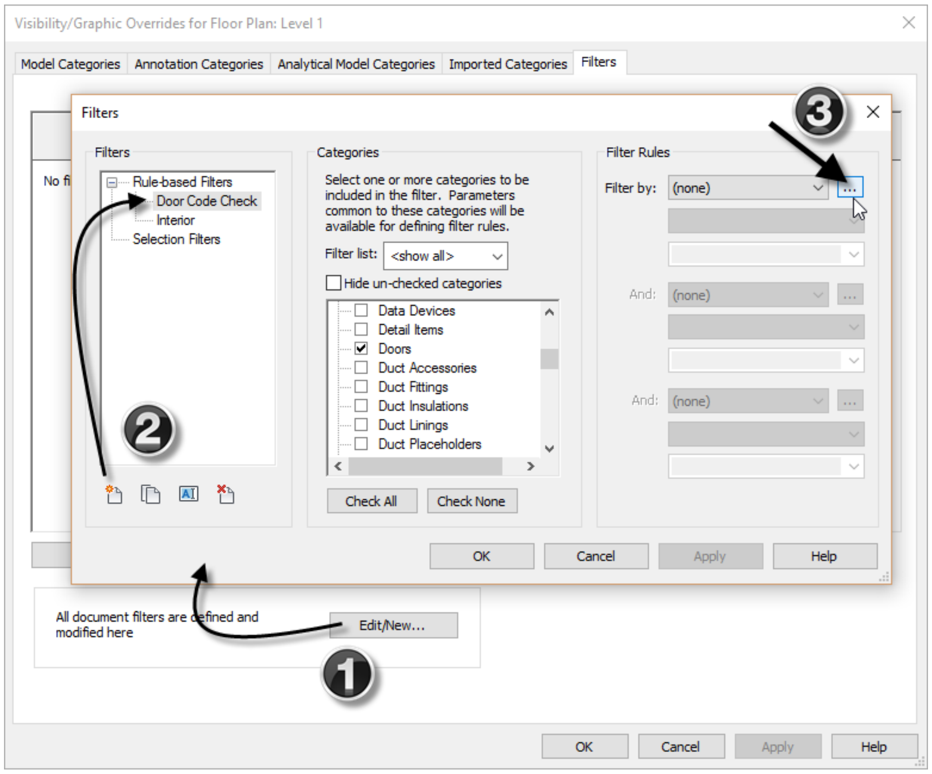
As you try values for *Typ Door Jamb* and apply them, the *Door 105 Check* dimension will adjust onscreen accordingly. (Sadly, in my testing, you have to click OK to update the reporting parameter in the dialog, it seems Apply will not do it).

8. Set the value of Type Door Jamb to: **6"** and then click OK

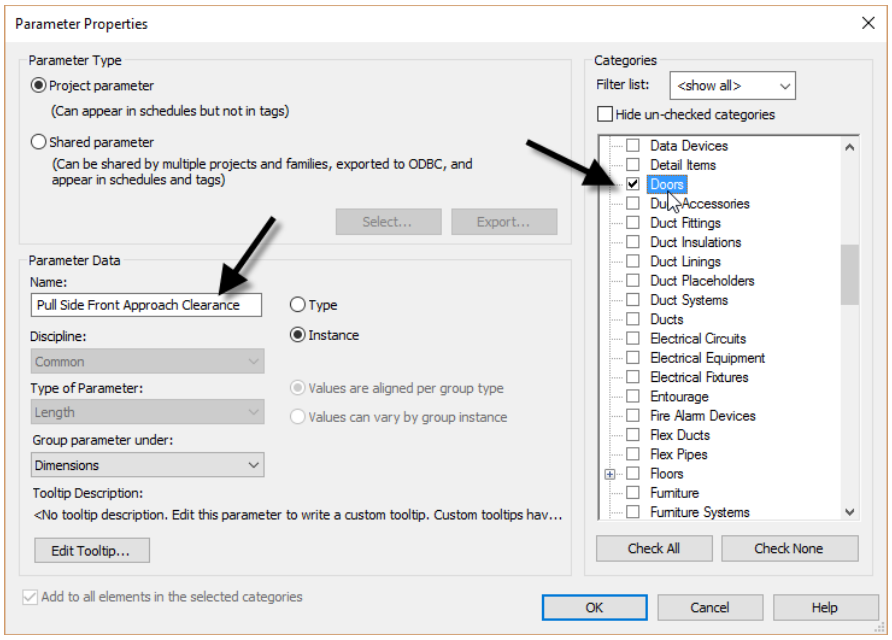
CATCH UP! You can open the file completed to this point named: ***02_Door Check_A.rvt***.

Now we want to have an onscreen alert when the value of the *Door 105 Check* parameter drops below that allowed by code. To do this, we need a new Project Parameter for the doors category and a filter in this view.

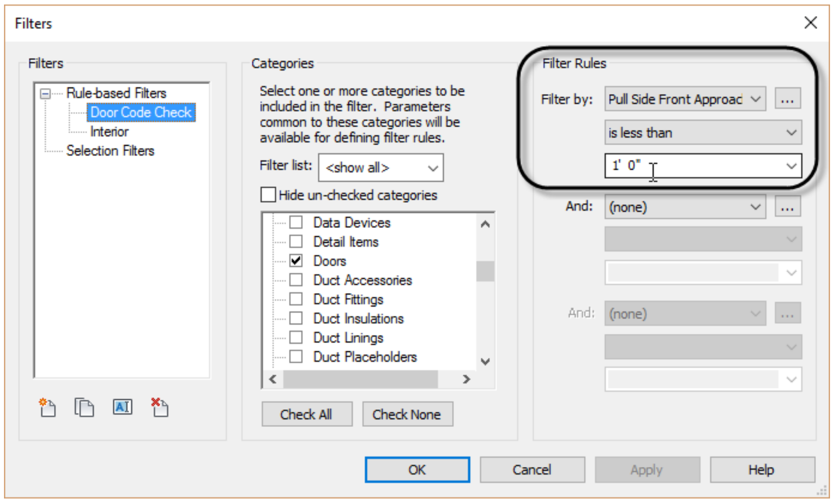
9. On the view tab, click the Visibility/Graphics button (or type: VG).
10. In the "Visibility/Graphic Overrides" dialog, click the Filters tab.
11. At the bottom, click the Edit/New button.
12. In the "Filters" dialog, click the new icon. Name it: **Door Code Check** and then click OK.
13. In the middle of the dialog, check the Doors category.
14. On the right, next to Filter by, click the small browse icon.



15. In the “Project Parameters” dialog that displays, click the Add button.
16. Name the new parameter: **Pull Side Front Approach Clearance**. On the right, assign it to Doors, accept the other defaults and then click OK twice.



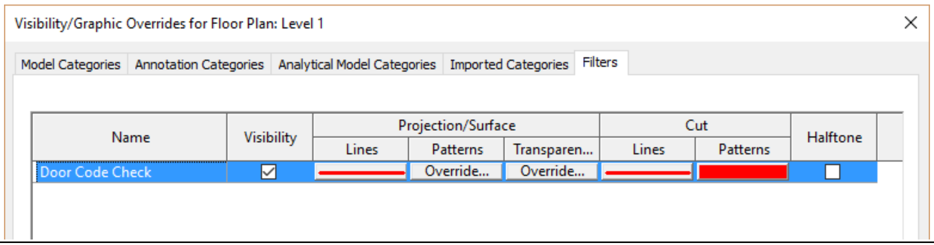
17. Back in the "Filters" dialog, from the Filter by list, choose: Pull Side Front Approach Clearance. Set the operator to: **is less than**, and then type: **1'-0"** in the text field.



18. Click OK to dismiss the "Filters" dialog.

19. Back in "Visibility/Graphics," click the Add button and add this new Door Code Check filter to the view.

20. Choose whichever overrides you like for the filter. (It is a good idea to apply something to both projection and cut).


Visibility/Graphic Overrides for Floor Plan: Level 1
Model Categories
Annotation Categories
Analytical Model Categories
Imported Categories
Filters

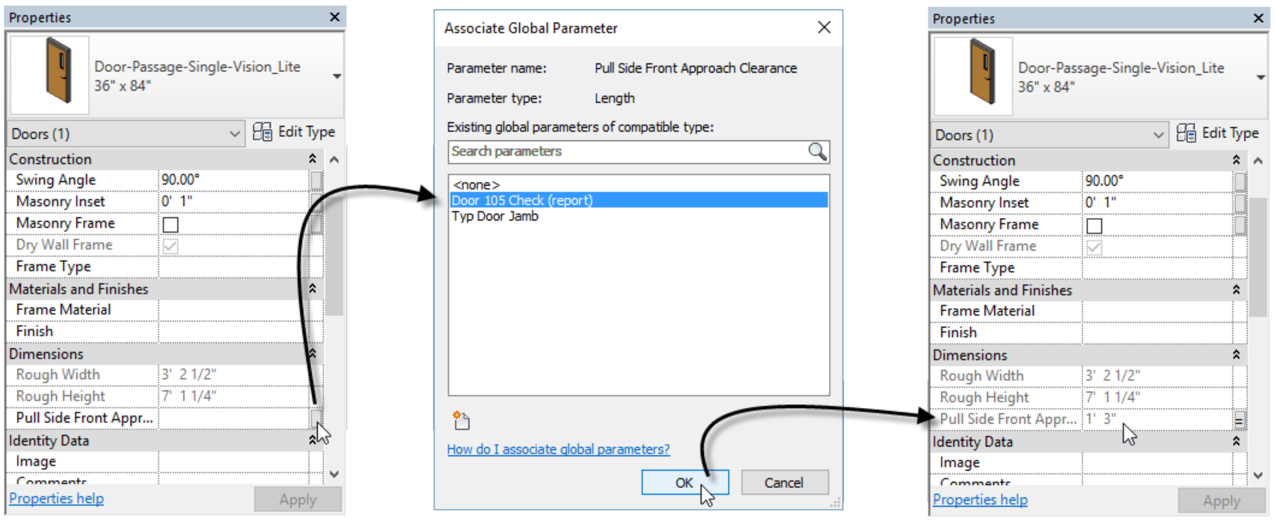
Name	Visibility	Projection/Surface			Cut		Halftone
		Lines	Patterns	Transparen...	Lines	Patterns	
Door Code Check	✓	🟥	Override...	Override...	🟥	🟥	☐

Catch Up! You can open the file completed to this point named: ***02_Door Check_B.rvt***.

That was a lot of work, and what is our pay off? Nothing! So we still have one more step to perform. We need to drive this new custom parameter on our door with our code checking global reporting parameter.

21. Select door 105 onscreen.

On the properties palette, locate Pull Side Front Approach Clearance. Notice the small associate global parameter button next to it.


Properties
Door-Passage-Single-Vision_Lite
36" x 84"
Doors (1)
Edit Type
Properties help
Apply
Associate Global Parameter
How do I associate global parameters?
OK
Cancel

Construction	
Swing Angle	90.00°
Masonry Inset	0' 1"
Masonry Frame	☐
Dry Wall Frame	✓
Frame Type	
Materials and Finishes	
Frame Material	
Finish	
Dimensions	
Rough Width	3' 2 1/2"
Rough Height	7' 1 1/4"
Pull Side Front Appr...	1' 3"
Identity Data	
Image	

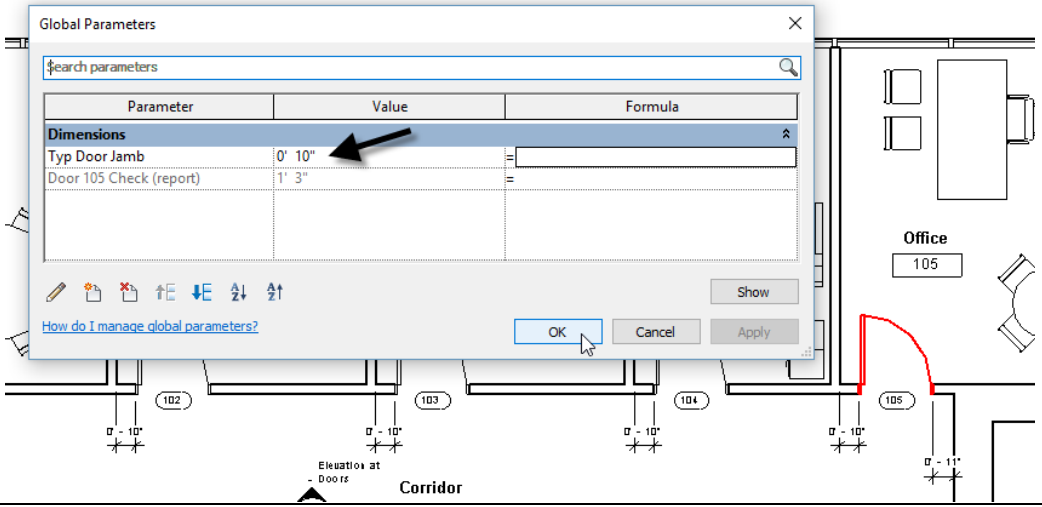
Parameter name: Pull Side Front Approach Clearance
Parameter type: Length
Existing global parameters of compatible type:
Search parameters
<none>
Door 105 Check (report)
Typ Door Jamb

22. Click the Associate Global Parameter button, choose: *Door 105 Check (report)* and then click OK.

Still nothing… Well, now we just need to flex.

23. Open the "Global Parameters" dialog and flex the *Typ Door Jamb* parameter.

As we saw above, the reporting parameter will update onscreen, but not in the dialog until you close and re-open it. But the doors should immediately change display to match your overrides in the filter above!



Parameter	Value	Formula
Dimensions		
Typ Door Jamb	0' 10"	=
Door 105 Check (report)	1' 3"	=

Catch Up! You can open the file completed to this point named: ***02_Door Check_C.rvt***.

How practical is this?

Now, is it worth it? Well this is a tricky one. The issue here is that since we are making an evaluation of the door's proximity to other model geometry (a wall in this case) we cannot simply build this into the door family. Now, we certainly could build in clearance geometry that we can toggle on and off, and many folks do this in their office standard content. But nothing about those clearance solids will react when they intersect walls or other geometry save when performing a clash detection. This approach leverages the global parameter to assess this very specific condition: is the door too close to the wall. It works well, but there is a lot of setup here and you would potentially need several such parameters per door to test each possible condition. So as interesting as this solution may be, I think you will agree that is it not very practical.

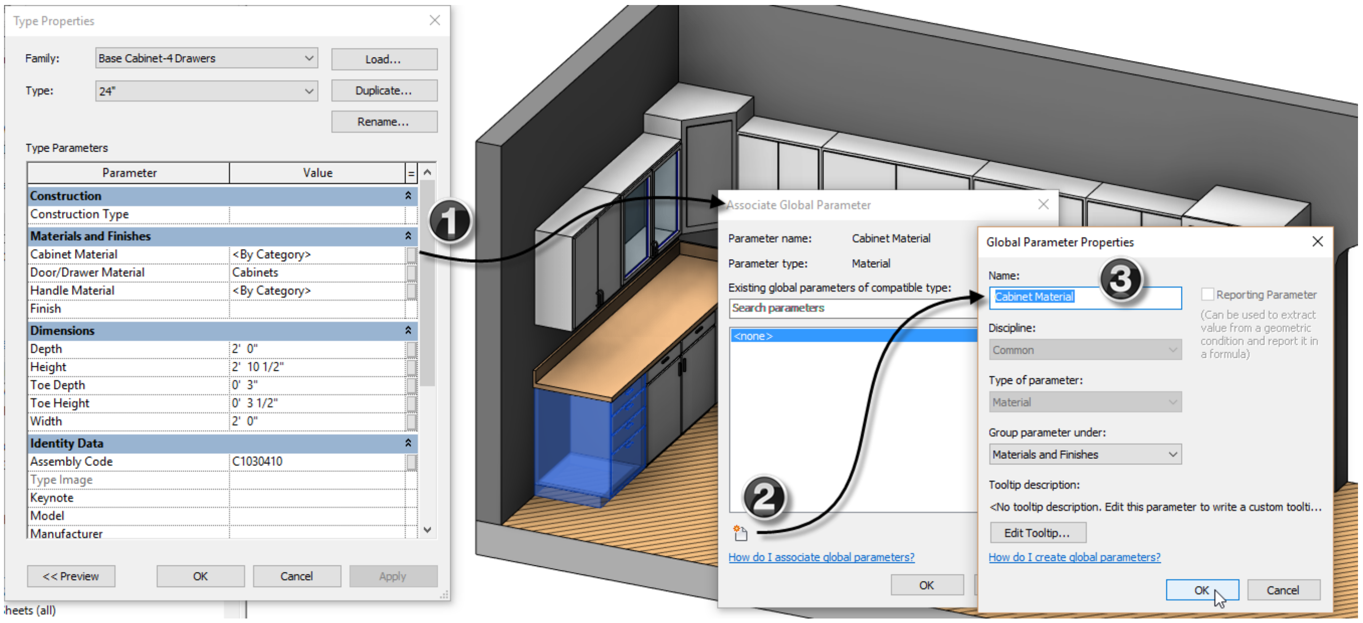
Driving Materials

Now let’s look at a much more practical example. In the family editor, when you have nested families, you can create parameters in the host family and have them drive the parameters of the nested families. Using global parameters, we can achieve similar behavior in a project allowing you to control several families at once without needing to embed the families into one another first. Driving a collection of materials across many similar families is one such example.

1. Open the file named: *03_Materials_!Start.rvt*.
2. Select one of the base cabinets onscreen and click the Edit Type button.

There are three material parameters. Notice that next to each one is an associate global parameter button.

3. Next to Cabinet Material, click the Associate Global Parameter button.
4. In the “Associate Global Parameter” dialog that appears, click the new icon at the bottom.
5. In the “Global Parameter Properties” dialog, name the new parameter: **Cabinet Material** and then click OK twice.


1
2
3
Type Properties
Load...
Duplicate...
Rename...
<< Preview
OK
Cancel
Apply
heets (all)
Associate Global Parameter
Global Parameter Properties
Edit Tooltip...
How do I associate global parameters?
How do I create global parameters?
OK
Cancel

Family: Base Cabinet-4 Drawers
Type: 24"

Type Parameters

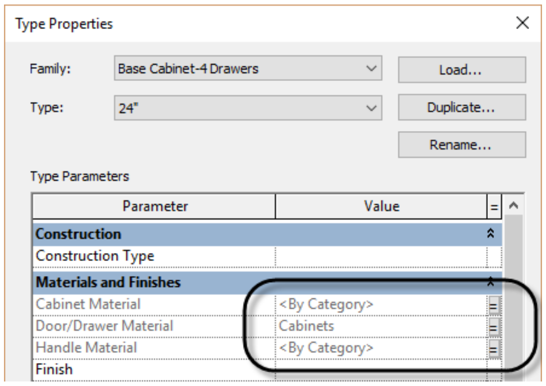
Parameter	Value
Construction	
Construction Type	
Materials and Finishes	
Cabinet Material	<By Category>
Door/Drawer Material	Cabinets
Handle Material	<By Category>
Finish	
Dimensions	
Depth	2' 0"
Height	2' 10 1/2"
Toe Depth	0' 3"
Toe Height	0' 3 1/2"
Width	2' 0"
Identity Data	
Assembly Code	C1030410
Type Image	
Keynote	
Model	
Manufacturer	

Parameter name: Cabinet Material
Parameter type: Material
Existing global parameters of compatible type:
Search parameters
<none>

Name: Cabinet Material
Reporting Parameter
(Can be used to extract value from a geometric condition and report it in a formula)
Discipline: Common
Type of parameter: Material
Group parameter under: Materials and Finishes
Tooltip description:
<No tooltip description. Edit this parameter to write a custom toolti...

6. Repeat for Door/Drawer Material and Handle Material.

7. Click OK in the “Type Properties” dialog to finish the first cabinet.



Type Properties

Family: Base Cabinet-4 Drawers

Type: 24"

Type Parameters

Parameter	Value
Construction	
Construction Type	
Materials and Finishes	
Cabinet Material	<By Category>
Door/Drawer Material	Cabinets
Handle Material	<By Category>
Finish	

8. Select the next base cabinet onscreen.
9. Edit Type and click the Associate Global Parameter button for the Cabinet Material.
10. Assign it to the Cabinet Material parameter you created above and then click OK.
11. Repeat for the other two materials and then click OK to complete this family.
12. Repeat the process for each of the remaining cabinet families in the project.

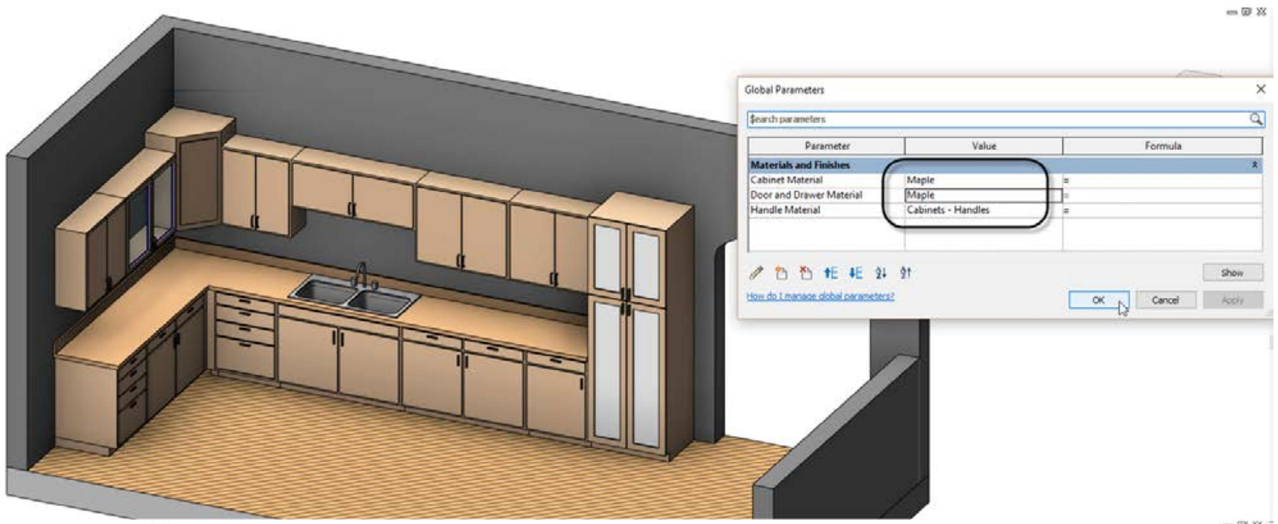
This is a little tedious and will take a bit of time, so if you prefer, you can skip to the catch-up file.

CATCH UP! You can open the file completed to this point named: ***03_Materials_A.rvt***.

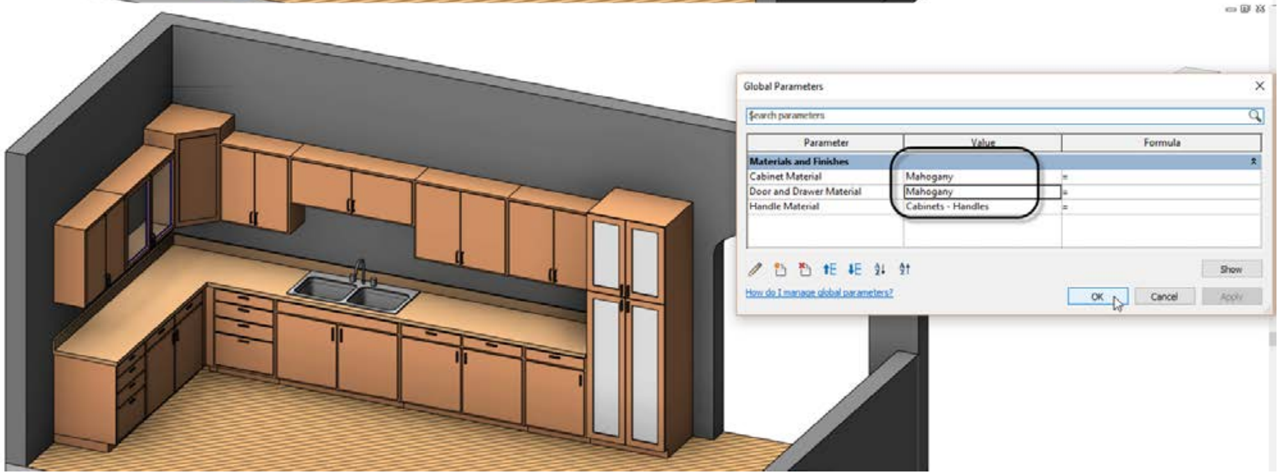
Also keep in mind that these are type parameters in these families. Therefore, if you intend to use any of the other types in this file later on, you will need to edit those types as well. This can be accomplished easily beneath the *Families>Casework* branch of the Project Browser.

To see the result, we now need to flex.

13. On the Manage tab, click the Global Parameters button.
14. Assign materials to each parameter and then click Apply.
15. Try different values and apply again.



Parameter	Value	Formula
Materials and Finishes		
Cabinet Material	Maple	=
Door and Drawer Material	Maple	=
Handle Material	Cabinets - Handles	=



Parameter	Value	Formula
Materials and Finishes		
Cabinet Material	Mahogany	=
Door and Drawer Material	Mahogany	=
Handle Material	Cabinets - Handles	=

Compared to the previous example, this one is much more practical. While it does take a little time to set everything up, once it is set up it becomes very easy to swap out an entirely new materials palette with just a few clicks.

Care to shoot some pool?

Here's another clearance example. Let's say we have a pool table that we want to ensure has enough space in the room that it will occupy. This information is readily available online. And we can easily do a quick measurement to check it. But if we put the values we want into a couple custom parameters, we can have the check performed live in the project and even display it in a custom color scheme.

1. Open the file named: *04_Pool Table_!Start.rvt*.

Build the parameters

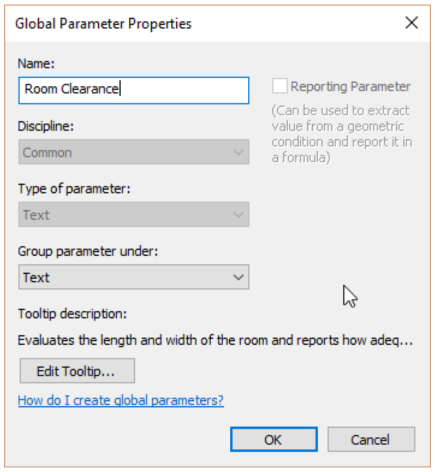
Let's start by labelling the dimensions in the Billiard Room.

2. Select each of the dimensions onscreen and label them with new parameters called: **Billiard Room Length** and **Billiard Room Width**.

Next we'll create a parameter to evaluate these dimensions compared to the recommended room size. There are a few ways to approach this. Assuming a 57" cue, the recommended size of the room for an 8-foot table is: 16'-9" x 13'-2". Naturally there would be a little give and take in these numbers and other factors might come into play like other room furnishings, circulation patterns and the precise placement and size of the table. To simplify this example, we will not factor in any of those extra variables.

For this example, we will consider three ranges: sizes less than the recommended, sizes within a foot or so of the recommended and sizes greater than the recommended. Furthermore, since our goal is to display the results graphically, we will combine the results of evaluating each dimension into a single formula.

3. On the Manage tab, click the Global Parameters button and create a new parameter.
4. Name the parameter: **Room Clearance**. Change the Type of parameter to: **Text** and for the tooltip, input: **Evaluates the length and width of the room and reports how adequate the size is compared to the recommended size**.


Global Parameter Properties

Name: Room Clearance

Reporting Parameter (Can be used to extract value from a geometric condition and report it in a formula)

Discipline: Common

Type of parameter: Text

Group parameter under: Text

Tooltip description:

Evaluates the length and width of the room and reports how adeq...

When building the formula, we will nest a few conditional statements together. You can build all of the variables directly into the formula, but it will be more flexible if we make some additional parameters. So let's make: Length Low (LL), Length High (LH), Width Low (WL) and Width High (WH).

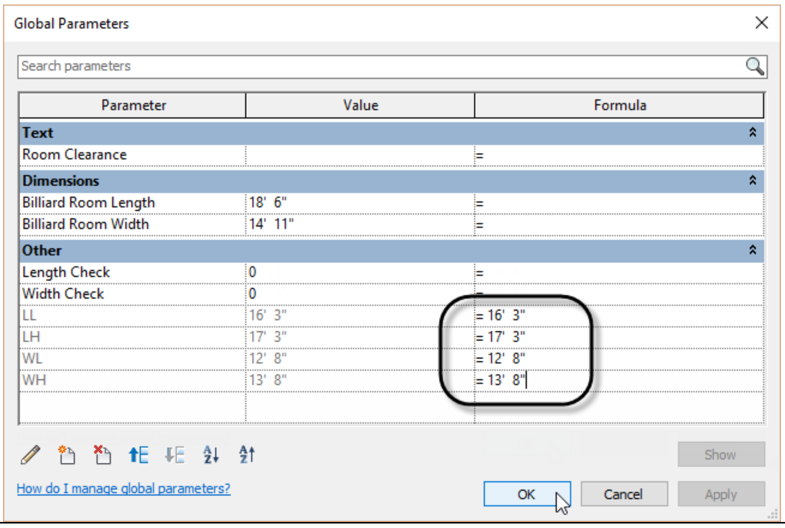
5. Create a new Length parameter grouped under: **Other** and name it: **LL**.
6. Repeat for **LH**, **WL** and **WH**.
7. In the Formula column, next to each of the four new parameters, input the following:

LL	**16'-3"**
LH	**17'-3"**
WL	**12'-8"**
WH	**13'-8"**

When you input the value into the formula column, it applies it to the Value column as a "read only" value. This is just a small failsafe to prevent accidental modification of these values. The formula can still be edited of course, but it just makes it "one step removed."

We need two more parameters to evaluate the length and width. We will make these integers and then use some simple formulas to combine everything.

8. Create two **Integer** parameters: **Length Check**, and **Width Check**.


Global Parameters

Search parameters

Parameter	Value	Formula
Text		
Room Clearance		=
Dimensions		
Billiard Room Length	18' 6"	=
Billiard Room Width	14' 11"	=
Other		
Length Check	0	=
Width Check	0	=
LL	16' 3"	= 16' 3"
LH	17' 3"	= 17' 3"
WL	12' 8"	= 12' 8"
WH	13' 8"	= 13' 8"

Catch Up! You can open the file completed to this point named: ***04_Pool Table_A.rvt***.

Assign the formulas

We will need three "IF" statements. One for each of the integer checks and one to combine them in the *Room Clearance* parameter. The concept we are using is this: evaluate the dimensions of the room and assign a number to the result. If it is below recommended values, we will assign a negative number. If it is within recommended values, we will assign zero. If it is above, we will assign positive one. When you add the results in the two parameters, it will yield a combined value that can be evaluated to give a final qualitative score.

9. In the formula field next to *Length Check*, input:

if(Billiard Room Length < LL,-2,if(and(not(Billiard Room Length < LL), Billiard Room Length < LH),0,1))

10. In the formula field next to *Width Check*, input:

if(Billiard Room Width < WL,-2,if(and(not(Billiard Room Width < WL), Billiard Room Width < WH),0,1))

11. In the formula field next to *Room Clearance*, input:

if(Length Check + Width Check <0,"Inadequate",if(Length Check + Width Check=0,"Acceptable","Generous"))

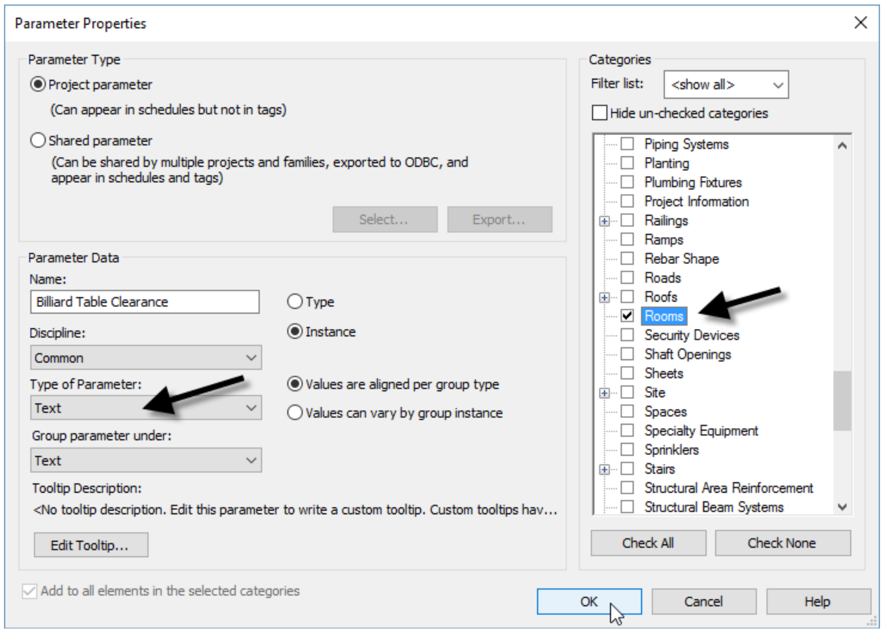
You can find these formulas in the file: *Pool Table Formula.txt*.

Catch Up! You can open the file completed to this point named: ***04_Pool Table_B.rvt***.

Create a Color Scheme

The value of the *Room Clearance* parameter should already be reporting the results and if you flex either the *Billiard Room Length* or the *Billiard Room Width* parameter, it will adjust to reflect the change. However, it might be nice to see the result graphically onscreen. Let's push the result of the *Room Clearance* parameter into a custom project parameter for the room.

12. On the Manage tab, click the Project Parameters button and then click Add.
13. Name the parameter: **Billiard Table Clearance**, for the Type of parameter, choose: **Text** and be sure to check the Rooms checkbox on the right.



Parameter Properties

Parameter Type

[✓ Project parameter] (Can appear in schedules but not in tags)

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

Parameter Data

Name: Billiard Table Clearance

Type [✓ Instance]

Discipline: Common

Type of Parameter: Text

Group parameter under: Text

[✓ Values are aligned per group type] Values can vary by group instance

Tooltip Description: <No tooltip description. Edit this parameter to write a custom tooltip. Custom tooltips hav...

Categories

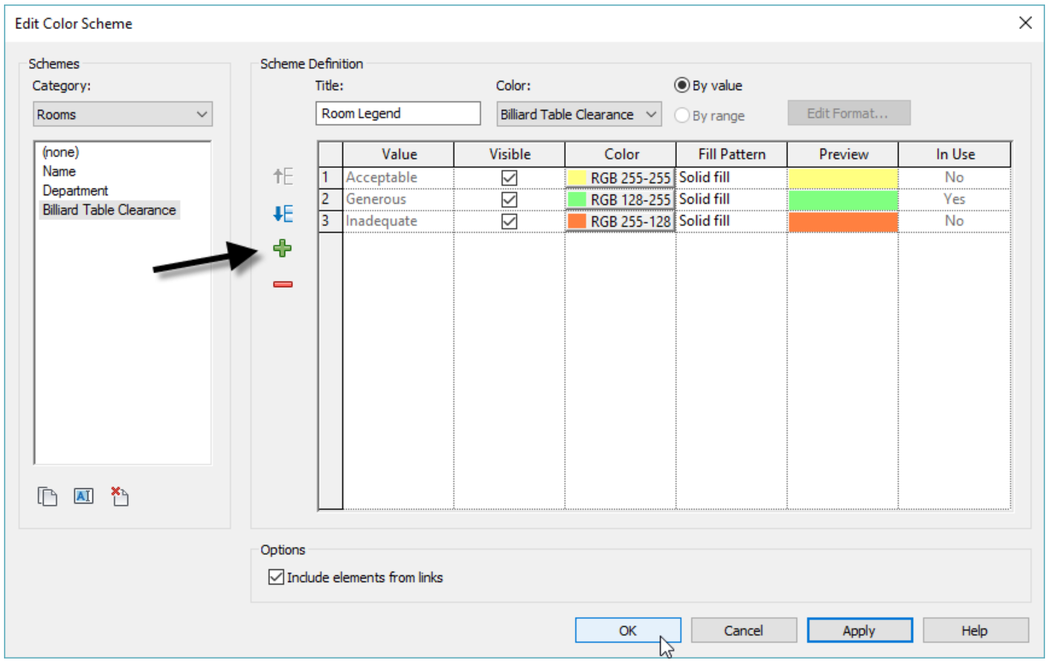
Filter list: <show all>

Hide un-checked categories

Piping Systems, Planting, Plumbing Fixtures, Project Information, Railings, Ramps, Rebar Shape, Roads, Roofs, [✓ Rooms], Security Devices, Shaft Openings, Sheets, Site, Spaces, Specialty Equipment, Sprinklers, Stairs, Structural Area Reinforcement, Structural Beam Systems

[✓ Add to all elements in the selected categories]

14. Click OK twice to finish.
15. Select the room onscreen, and then on the Properties palette, click the small associate global parameter icon next to Billiard Table Clearance.
16. Choose Room Clearance and then click OK.
17. Deselect the room, and on the Properties palette, click the <none> button for Color Scheme.
18. Change the Category to Rooms, and then Duplicate one of the schemes. Call it: **Billiard Table Clearance**.
19. In the Scheme Definition area, click the Add Value button and name it: **Generous**. Change the color to Green.
20. Add another two values: **Inadequate** set it to Orange and **Acceptable** set it to Yellow.
21. Change the Color option to: **Billiard Table Clearance** and click OK in the message that appears.


Edit Color Scheme

Schemes
Category: Rooms

(none)
Name
Department
Billiard Table Clearance

Scheme Definition
Title: Room Legend
Color: Billiard Table Clearance
[✓ By value] By range

	Value	Visible	Color	Fill Pattern	Preview	In Use
1	Acceptable	✓	🟨 RGB 255-255	Solid fill	🟨	No
2	Generous	✓	🟩 RGB 128-255	Solid fill	🟩	Yes
3	Inadequate	✓	🟧 RGB 255-128	Solid fill	🟧	No

Options
✓ Include elements from links

22. Click OK to see the result.

23. Open Global Parameters and flex the size of the room.

To keep the pool table centered in the room, create two more parameters with formulas that equal half of the Length and Width.

CATCH UP! You can open the file completed to this point named: ***04_Pool Table_C.rvt***.

As a proof of concept this example delivers, but I'm afraid that this one is also not super practical. For examples where you only have to check a small number of items, this could be useful. But given the quantity of parameters that might be required if you have to check many such items, it will become tedious to implement very quickly.

Something more practical

As interesting as the potential of that previous example may be, it can leave you wanting a more practical use for global parameters. Let's look at a simple yet useful example. Controlling the spacing of repetitive elements.

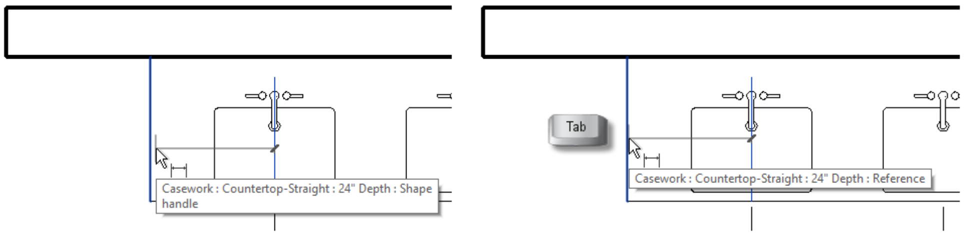
1. Open the file named: *05_Sinks_!Start.rvt*.

Setting equal spacing for elements like these sinks is easy. Just use an equality dimension.

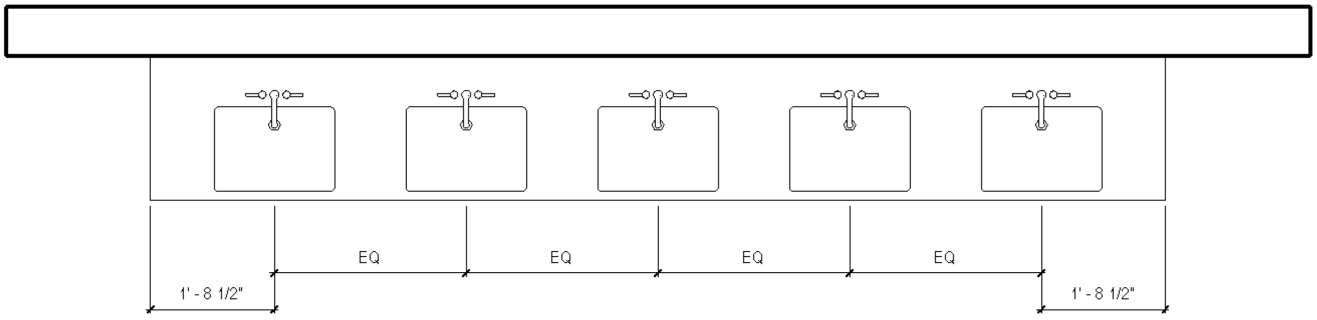
2. Add a dimension with a witness line at the center of each sink and then toggle on the equality.

3. Align the center sink to the center of the countertop.

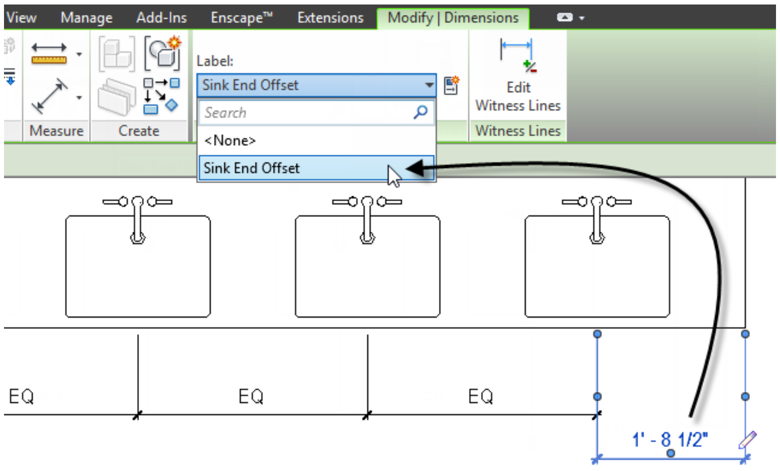
4. Add a dimension from the center of the end sink at the right to the end of the right end of the countertop.
5. Add a dimension from the center of the end sink at the left to the left end of the countertop. Use the TAB key to ensure that you are dimensioning to the "Reference" and not the "Shape Handle" for this end.



Casework : Countertop-Straight : 24" Depth : Shape handle
Tab
Casework : Countertop-Straight : 24" Depth : Reference



6. Select one of the side dimensions and label it with a new parameter called: **Sink End Offset**.
7. Select the other end dimension and label it with the same dimension.



8. Open the "Global Parameters" dialog and flex the *Sink End Offset* parameter.

CATCH UP! You can open the file completed to this point named: ***05_Sinks_A.rvt***.

With this arrangement, you can control the end distances equally and have all the sinks spaced equally as well.

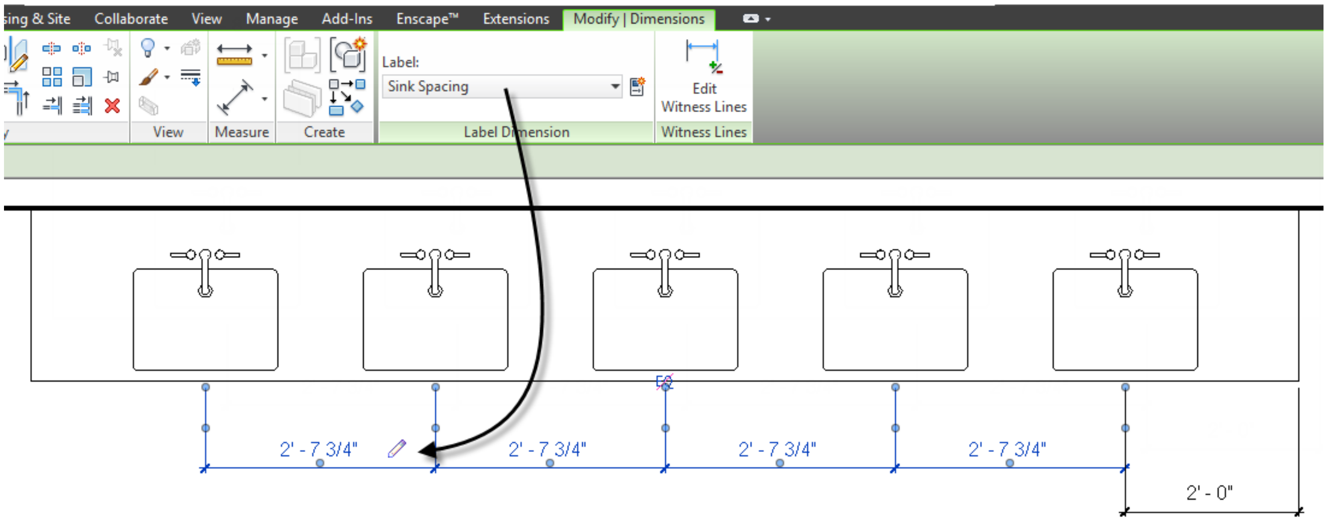
Now how much would you pay?

If you want the end distances equal to the space between the sinks, you need to do some math and configure things a bit differently. First, we will replace the equality dimension with a global parameter. Next, instead of the end offset parameter, we will calculate the distance from one end. Finally, we will use a reporting parameter to help us do the math and another length parameter to drive the length of the countertop and to form the basis of the calculation.

9. Delete the dimension at the left end. When the warning appears, click the Unconstrain button.

To be sure that the parameter is removed, you can use the Reveal Constraints icon on the View Control Bar.

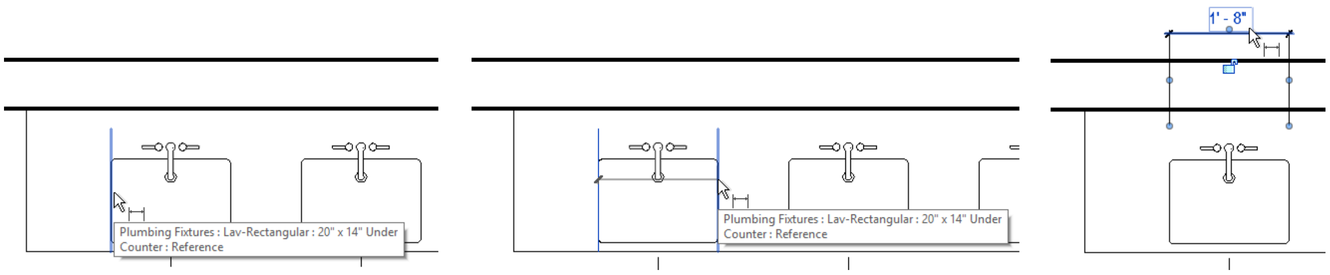
10. Select the equality dimension and toggle off the equality. Then label it with a new parameter called: **Sink Spacing**.



Notice that with global parameters, you can actually label a multi-segment dimension! This behaves identically to an equality dimension since the label is applied to all segments of the multi-segment dimension.

11. Dimension one of the sinks (with witness lines measuring the sink's width).

12. Label this with a new parameter called: **Sink Width** and then click OK.



Plumbing Fixtures : Lav-Rectangular : 20" x 14" Under Counter : Reference
Plumbing Fixtures : Lav-Rectangular : 20" x 14" Under Counter : Reference
1' - 8"

13. Select the countertop and then on the Properties palette, click the associate global parameter button next to the Length parameter.
14. In the "Associate Global Parameter" dialog, click the New Global Parameter icon and name the new parameter: **Countertop Length**. Click OK twice.

In this case, the quantity is five sinks. We won't be varying this. In my tests, I found no good way to have global parameters drive an array. So if you need a variable number of sinks, you cannot do this with a GP.

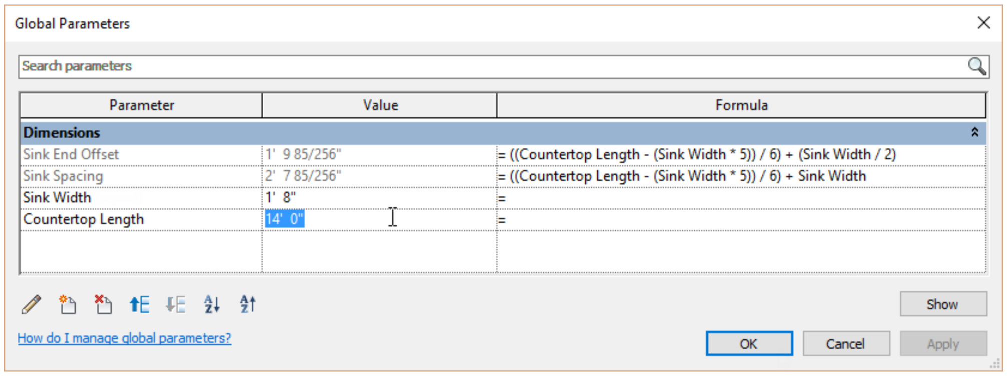
To calculate the spacing, we subtract the width of five sinks from the countertop length, divide this by six spaces (four between sinks, one at each end) and then add this value back to the width of one sink. This is the distance between sinks. To calculate the end offset, it is the same calculation but added to only half the sink width.

15. Open the "Global Parameters" dialog and in the formula field for the *Sink End Offset* parameter, input:

    ((Countertop Length - (Sink Width * 5)) / 6) + (Sink Width / 2)

16. For the formula of the *Sink Spacing* parameter, input:

    ((Countertop Length - (Sink Width * 5)) / 6) + Sink Width



Parameter	Value	Formula
Dimensions		
Sink End Offset	1' 9 85/256"	= ((Countertop Length - (Sink Width * 5)) / 6) + (Sink Width / 2)
Sink Spacing	2' 7 85/256"	= ((Countertop Length - (Sink Width * 5)) / 6) + Sink Width
Sink Width	1' 8"	=
Countertop Length	14' 0"	=

17. Flex the Countertop Length.

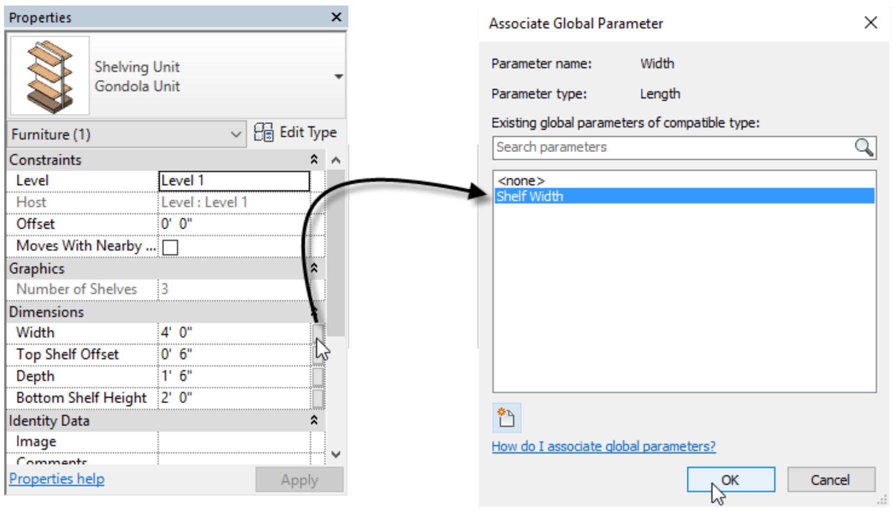
CATCH UP! You can open the file completed to this point named: ***05_Sinks_B.rvt***.

Create a collection of shelving units

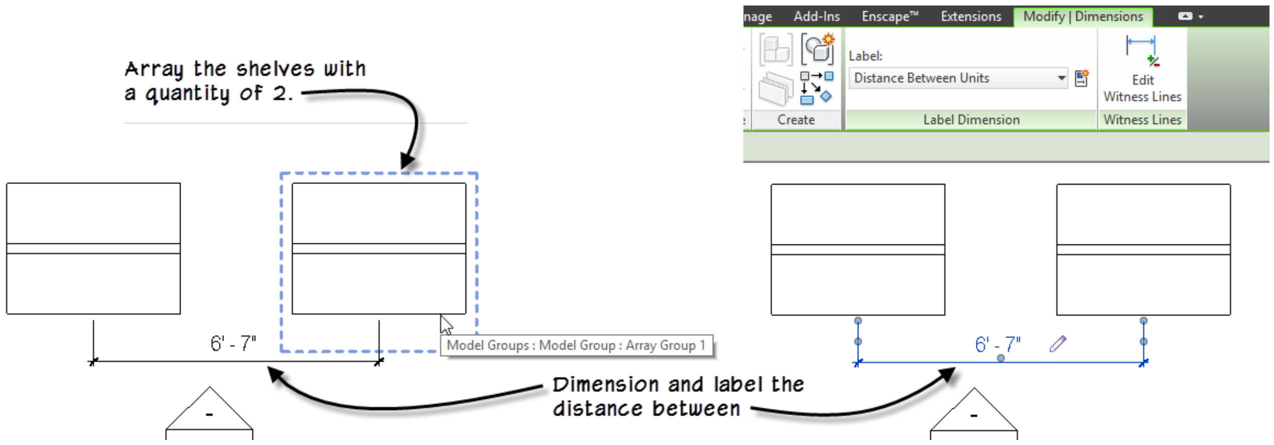
Let's look at another practical example. In this exercise, we will layout a collection of retail shelving units and set the spacing with global parameters. This example uses an array, but as I noted above, sadly we cannot drive the array quantity with a global parameter. So you have to flex the quantity onscreen and the spacing values in the "Global Parameters" dialog.

1. Open the file named: *06_Shelves_!Start.rvt*.

2. Select the free-standing shelving unit and then on the Properties palette, click the small Associate Global Parameter icon next to the Width parameter.
3. Click the Add Parameter icon and create a new parameter called: **Shelf Width**.

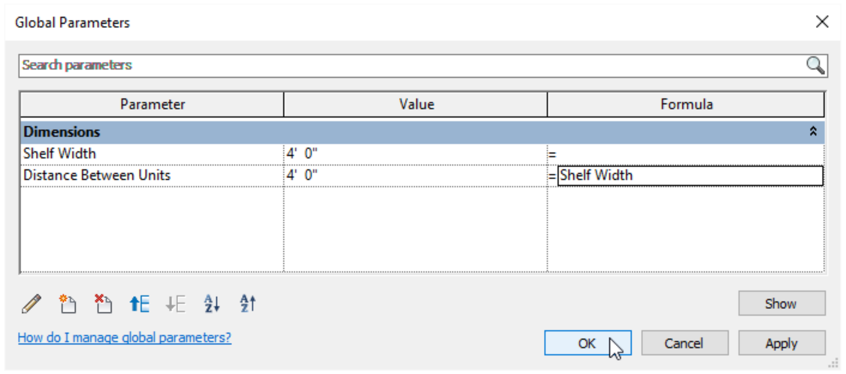


4. Array the shelving unit. Be sure Group and Associate is checked and the quantity is: **2**.
5. Add a dimension between the two resulting array group elements.
6. Select the dimension and then label it with a new parameter named: **Distance Between Units** and then click OK.



Now we can use the Global Parameters dialog to link these two global parameters together.

7. On the Manage tab, click the Global Parameters button.
8. In the Formula field next to *Distance Between Units*, type: **Shelf Width** (Case sensitive).


Global Parameters

Search parameters

Parameter	Value	Formula
Dimensions		
Shelf Width	4' 0"	=
Distance Between Units	4' 0"	= Shelf Width

9. Click OK to finish.

Your shelving units should now be flush to one another. Furthermore, you can now flex the Shelf Width (in the "Global Parameters" dialog) or adjust the quantity of arrayed elements (direclty onscreen) and the shelving units will all flex together and remain flush to one another.

CATCH UP! You can open the file completed to this point named: ***06_Shelves_A.rvt***.

Curtain Wall bay spacing

When global parameters first came out, this was one of the first things I thought of using them for. When designing curtain walls, you have three options: use a type-based design, lay it out manually or do a combination of the two.

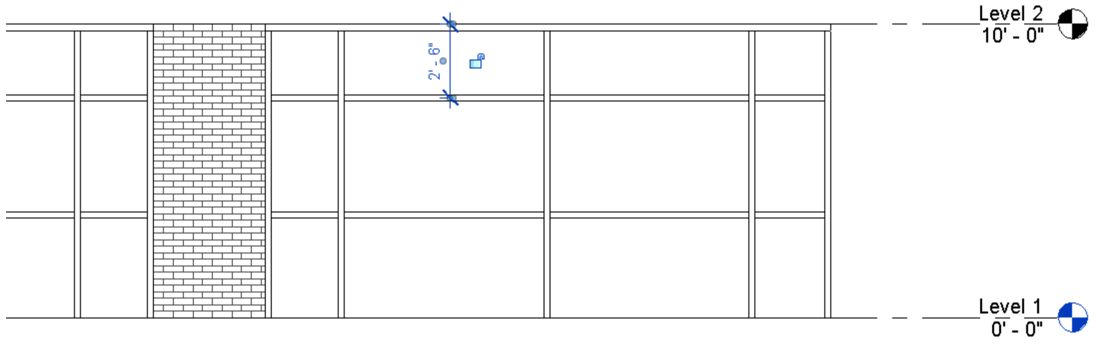
Type-based designs are nice because they automatically add curtain grids, mullions and panels based on the rules in the type. However, you only get one grid spacing in each direction. If you lay it out manually, you can vary the spacing in each direction in any way that you like, but you must layout each grid line manually. It is also possible to leverage type-based settings to establish the overall bay spacing and then add additional bays manually.

Equality dimensions and multiple copy can help with the manual bays, but it is still a tedious process; particularly when modifying the design. Well this is where global parameters can really help. You can now define parameters for each major bay spacing and then apply them directly to the grids in the curtain wall design. When it comes time to edit, simply modify the global parameters! Here's a simple example:

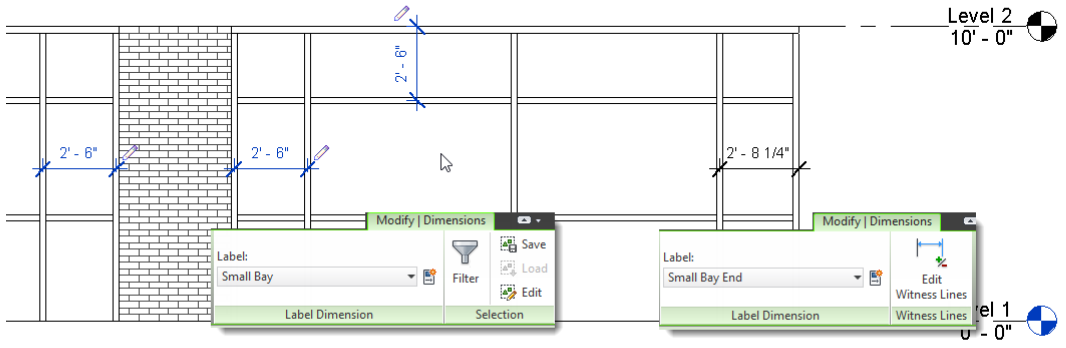
1. Open the file named: *07_Curtain Wall_!Start.rvt*.

This design is laid out manually, the grid spacing in each direction is set to: None in the type. Notice that the sides and top are narrower than the middle bays.

2. On the Project Browser, open the *South* elevation.
3. Add a dimension from Level 2 down to the horizontal curtain grid directly beneath it.



4. Repeat to dimension the two small left and right bay in each larger bay.
5. Select all of the dimensions except the rightmost and leftmost vertical bays.
6. Label them with a new parameter called: **Small Bay**.



7. Select the rightmost and leftmost dimensions and label them with a new parameter called: **Small Bay End**.
8. Open the “Global Parameters” dialog and in the formula field next to Small Bay End, input: **Small Bay + 1 1/4"**.
9. Flex the Small Bay parameter.

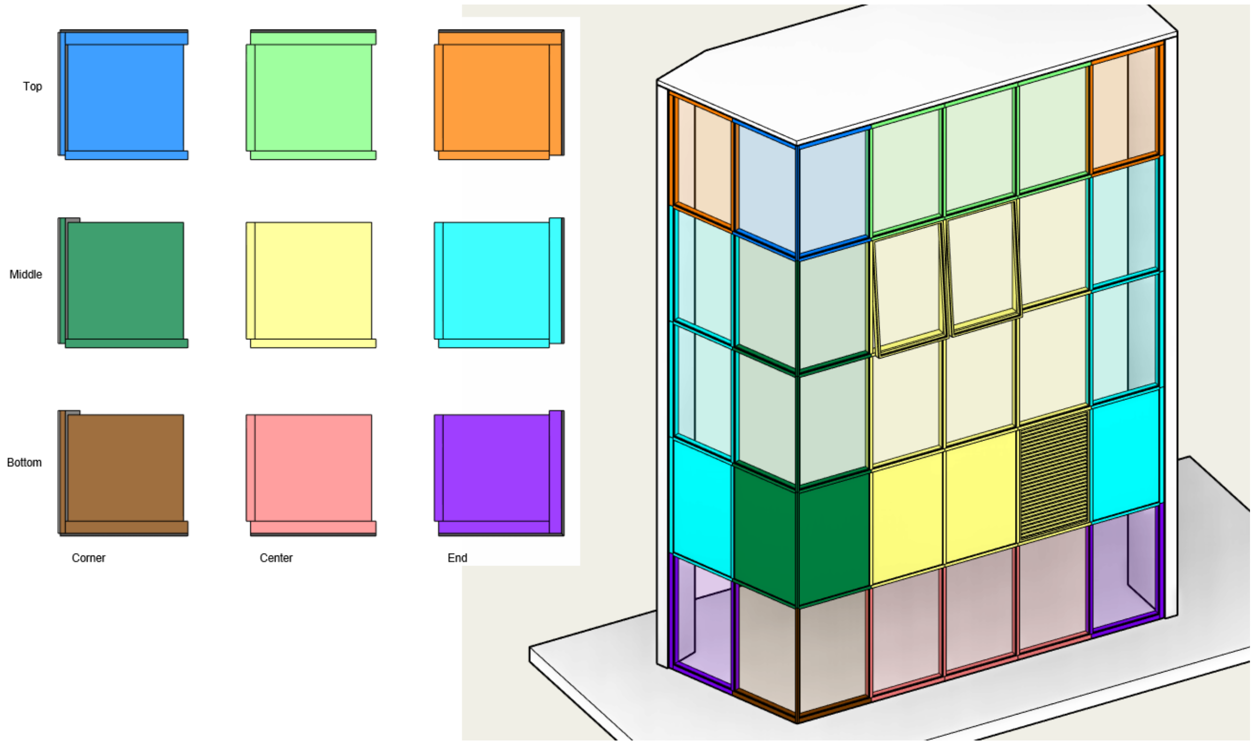
CATCH UP! You can open the file completed to this point named: ***07_Curtain Wall_A.rvt***.

This is just a simple example. Using the same strategy, you could build a much more robust curtain wall design. You are encouraged to experiment further on your own.

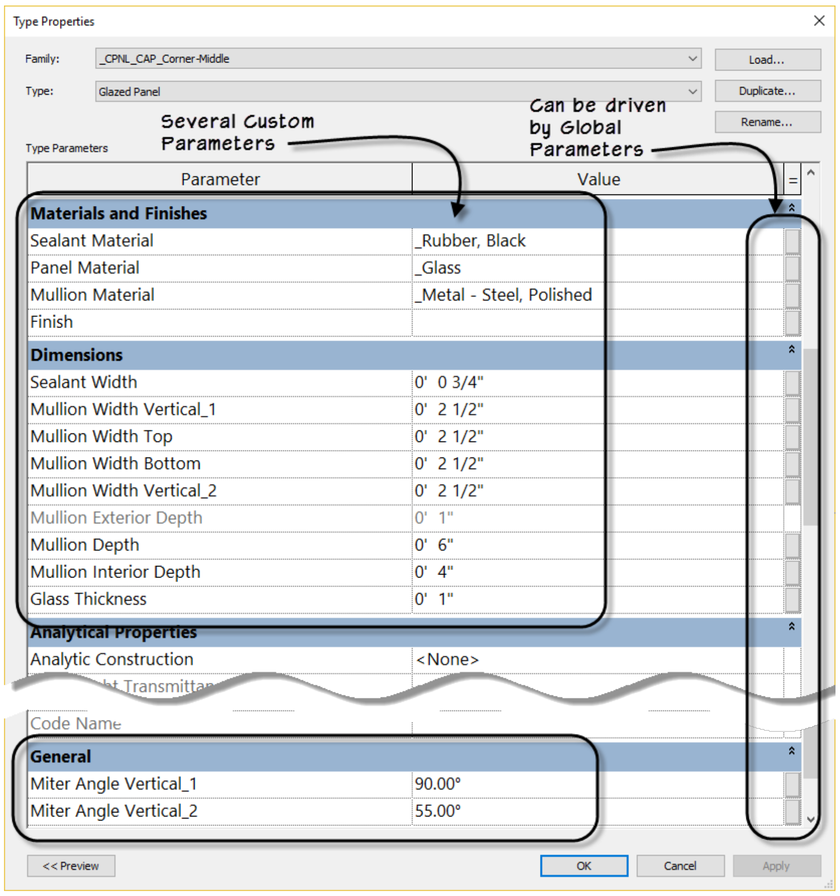
Driving Custom Curtain Panels

Here is another curtain wall example. This one uses custom created curtain panels that were created for a client of mine. This solution uses curtain panels only and does not make use of Revit’s mullion elements. Instead, the mullions are built into the panels. This approach allows for specialized mullion conditions that are not easily accomplished with the built-in mullions. For example, on corner panels, there is a rotational parameter that allows the corner mullions to be at nearly any angle and show proper cleanup of surrounding geometry like the glazing and other mullions running into them horizontally and vertically. On the edges, there is a sealant option, and other options are also baked in.

This functionality comes at a cost however. We had to create about a dozen separate families per panel design to accommodate all possible variations and avoid either having “half mullions” with seams where they touch or overlapping geometry due to doubled up mullions. There are nine basic conditions, with other options for operable windows and louvers.



The solution was built several Revit releases ago when global parameters were not yet available. Therefore, using these panels is quite tedious. The designer must edit up to a dozen (or more) separate families and types to make changes to things like the mullion size, angle, and sealant conditions. In order to make sure they all “fit together” properly, the values assigned to each parameter must be manually coordinated. Here is a look at the “Family Types” dialog for a typical family in this collection to give you some idea of the scope of this task:



Family: _CPNL_CAP_Corner-Middle

Type: Glazed Panel

Type Parameters

Parameter	Value
Materials and Finishes	
Sealant Material	_Rubber, Black
Panel Material	_Glass
Mullion Material	_Metal - Steel, Polished
Finish	
Dimensions	
Sealant Width	0' 0 3/4"
Mullion Width Vertical_1	0' 2 1/2"
Mullion Width Top	0' 2 1/2"
Mullion Width Bottom	0' 2 1/2"
Mullion Width Vertical_2	0' 2 1/2"
Mullion Exterior Depth	0' 1"
Mullion Depth	0' 6"
Mullion Interior Depth	0' 4"
Glass Thickness	0' 1"
Analytical Properties	
Analytic Construction	<None>
Code Name	
General	
Miter Angle Vertical_1	90.00°
Miter Angle Vertical_2	55.00°

As you can see, there are many parameters and panel conditions. So editing is quite a lot of work. Global parameters can greatly simplify this! Notice in the figure, that all of these custom parameters have the “Associate Global Parameter” button. There are examples here of Material parameters (like the “Driving Materials” example covered above), dimensions and angles.

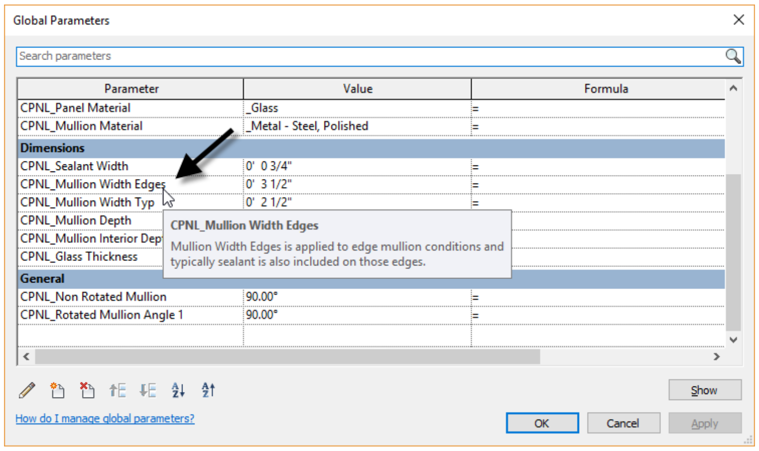
I will not be sharing the dataset for this example. I will describe the concept here in the paper and give an overview. Since these families are the property of my client, I am unable to provide them in the dataset for this class. However, this is really just a more complex example of the “Driving Materials” example covered above. You can review the concept with those files, or you can apply the concept to your own files.

Driving Many Custom Parameters

To make this solution work, all the effort is in the up-front setup. It is common for many firms to create “library” or “warehouse” files that contain examples of their firm’s custom families. Usually these items are inserted in the sample library file. To access those items, the end users can either use the Transfer Project Standards command or simply open the file, select the item(s) required and copy and paste them to their project file. Often the copy and paste method is easier and more surgical. Transfer Project Standards copies over all elements in a selected category, where with the copy and paste approach, you can focus on just the items you select. Further, when copying and pasting elements that have global parameters assigned to them, the global parameters also get copied over. This means that if the preliminary work is completed in the library or warehouse file, the experience for the end user is limited to a simple copy/paste operation.

Each of these curtain panels has four separate parameters controlling the mullion widths. These correspond to left, right, top and bottom. Left and right however are named Vertical_1 and Vertical_2 since orientation will vary. Each family required the four separate parameters for flexibility, but in most cases, you will only need two parameters to drive these: a "typical" mullion width and another mullion width at the edges.

So in this example, I have created just two Global Parameters: **CPNL_Mullion Width Typ** and **CPNL_Mullion Width Edges**. You can use the parameter tooltip to explain the use of these parameters. I am prefixing the names with "CNPL" to keep them separate from other similar parameters that might be used in the project.



Parameter	Value	Formula
CPNL_Panel Material	_Glass	=
CPNL_Mullion Material	_Metal - Steel, Polished	=
Dimensions		
CPNL_Sealant Width	0' 0 3/4"	=
CPNL_Mullion Width Edges	0' 3 1/2"	=
CPNL_Mullion Width Typ	0' 2 1/2"	=
CPNL_Mullion Depth		
CPNL_Mullion Interior Dep		
CPNL_Glass Thickness		
General		
CPNL_Non Rotated Mullion	90.00°	=
CPNL_Rotated Mullion Angle 1	90.00°	=

CPNL_Mullion Width Edges

Mullion Width Edges is applied to edge mullion conditions and typically sealant is also included on those edges.

I can use the same strategy with the angle parameters. One parameter for the non-rotated (90°) mullions and another for each rotation as appropriate. In this example, I have just one rotation of 55°. There are other parameters as well, like Glass Thickness, Shadow Box Depth and the materials. You can make as many parameters as required by your content.

Reusing the Configured Curtain Panels

The hardest part of this process is the setup. In this simple example, I have 15 custom parameters that need to be mapped and a total of 26 family types (spread over 15 families). That means that I had to click that little associate button **390** times in the setup process. (Not fun…) Once complete however, the designer's experience is much smoother:

1. Open the warehouse/library file.
2. Select the curtain walls/panels that you need and copy them to the clipboard.
3. Create or open another project.
4. Paste from clipboard and place the elements onscreen. (You can optionally delete them after placement, but be sure to place them and finish the paste first, then delete, otherwise they will not import completely).

You can now use these panels in your design. If you open the “Global Parameters” dialog, you will see that all the parameters have come across with the paste. Furthermore, if you edit the type of any curtain panel, you will see that these global parameters are already assigned. So all you need to do now is:

5. Flex the global parameters!

Type or Instance?

When associating global parameters, they can apply to either instance or type parameters in your families. This is interesting from the perspective of element hierarchy. If you apply a global parameter to instance properties, it behaves a little like making a section of elements and then changing the properties of the entire selection. It is almost like a saved selection for just that particular parameter.

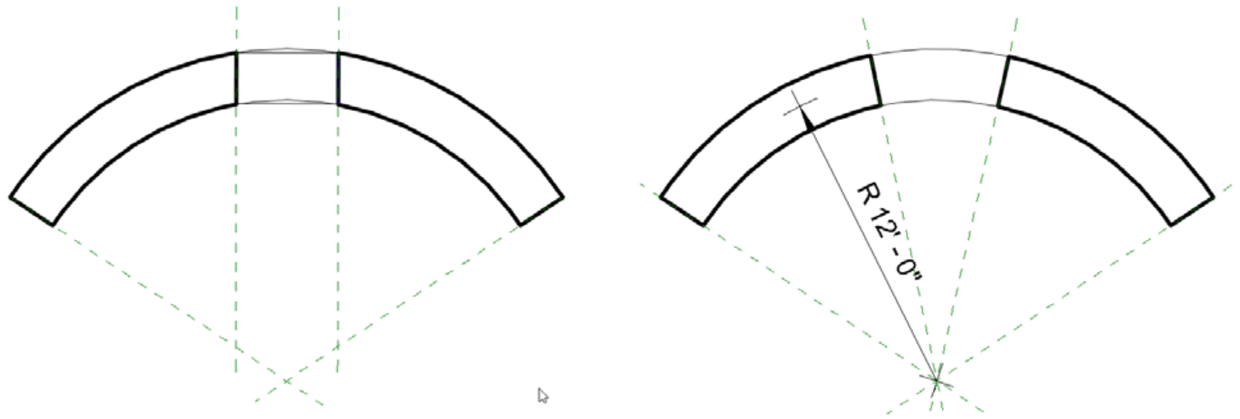
It is even more interesting when you associate a global parameter to a type parameter. In this case, you are taking advantage of the opportunity to create a common link between multiple types and properties. In this example we are taking a single global parameter (like Mullion Typ) and using it to drive several parameters across several types. However, it is important to realize that you *must* assign the relationships to *each* type. For example, if you associate global parameters to the parameters of one type in a family with three types, the other two types will *not* have the association. In other words, associations behave just like type values. Each type must be edited separately. And it is possible to have one type driven by global parameters and have another type that is not in the same family. This can actually be an advantage in situations where you need a special custom condition that you prefer to manually input values directly rather than using global parameters. You certainly could make another set of global parameters, but simply duplicating a new type might be all you need in such a circumstance.

For this curtain panel example, an argument could be made for using instance-based parameters. Certainly doing so would make the setup process easier. Instead of having to separately associate everything (with 309 clicks), I could have simply selected all the curtain panels onscreen and then on the properties palette, linked up the associations just one time (for each parameter – 15 clicks) for the whole selection. While tempting, consider that the goal here was to create a library or warehouse file with all the associations already connected. If they are instance-based properties, users will have to use the actual objects they copy and paste, or else they would have to make a new selection and link up the parameters again each time they add more panels. In some cases, that might be OK or even preferable, but in this case, I like the type-based approach because parameter associations come along with pasting. Once all the type-based properties are linked up, those associations remain even when new elements are created based on those pasted types. And as noted above, we can always make a new type and leave the parameters unassociated for special cases.

Labeling a Radial Dimension

New in Revit 2018, global parameters can be used to label radial and diameter dimensions. Let’s look at a simple example of that. Wall-hosted families like doors and windows allow you to create an opening in a wall. This is achieved in the default templates with an Opening Cut element. This element works well for most simple openings. An opening cut uses a simple sketch drawn on the elevation of the wall and creates a straight extrusion cutting all the way through the wall. If you want a more complex three-dimensional opening shape, or do not wish to cut all the way through the wall’s thickness, you need to use a void object instead.

Further complicating the issue, what if you want to create an opening in a curved wall? As mentioned, the default opening cut will create a simple extrusion that cuts all the way through the wall. And its sides will be parallel (left); they will not converge on the wall's radius (right):



If you want to create an opening element that cuts a curved wall and has its sides follow the wall's radius, then you must use a void instead of an opening cut in the door or window family and then use a global parameter to ensure that radius parameter of the opening always matches the host wall.

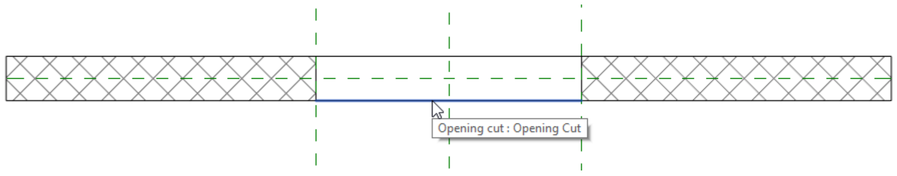
Create a New Family

Let's build this one from scratch. We'll make a simple wall opening only. You can add window geometry, glazing and trim later if you like.

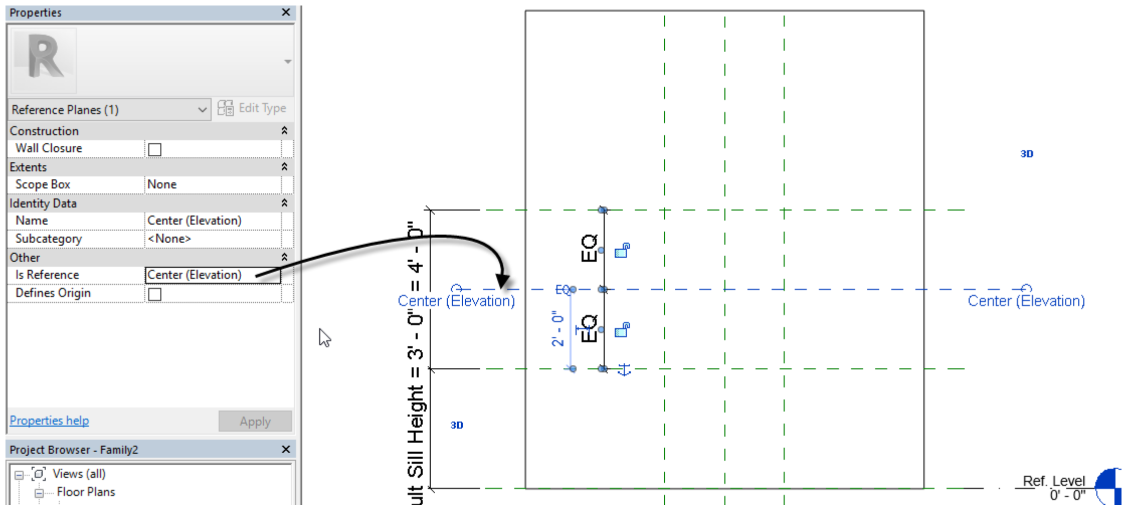
1. From the File menu, choose: **New** > **Family**.
2. Choose the template called: *Window.rft* and then click Open. (If you don't have this template, a copy has been included in the dataset).

This is the default window family template. It contains a stand-in host wall, an opening cut, indications of the exterior and interior sides and a few starting parameters. For this exercise, we will delete the opening cut and replace it with a void that will respond to a radius.

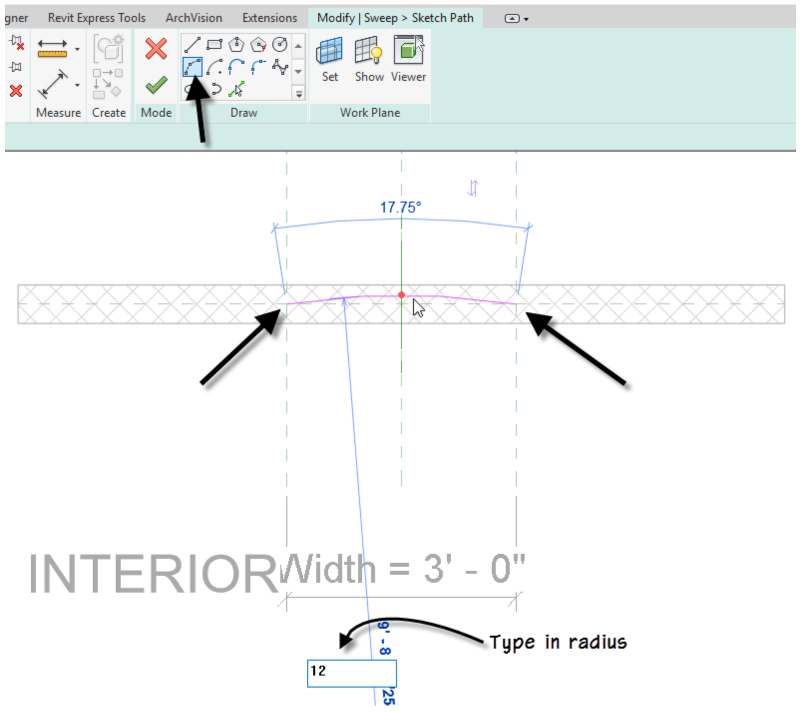
3. Select the Opening Cut onscreen and delete it.



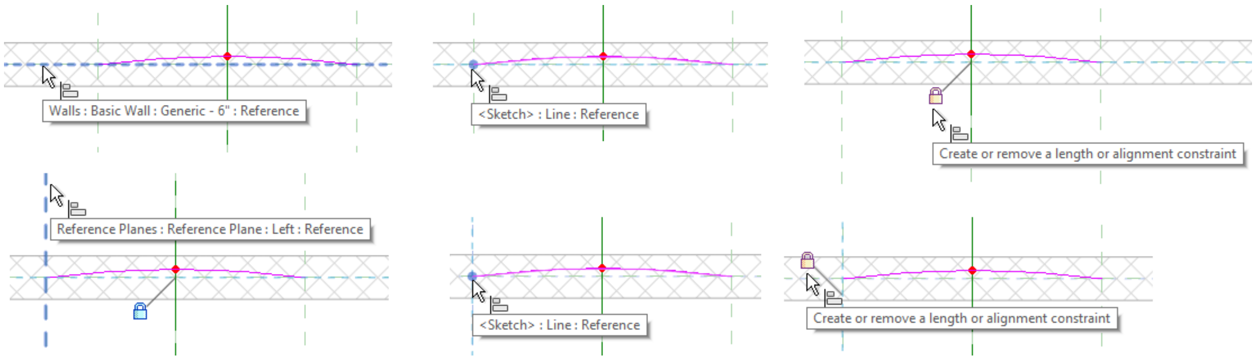
4. Open the *Exterior* elevation view and zoom to fit if required.
5. Add a reference plane horizontally between the Sill and Top reference planes.
6. Name it: **Center (Elevation)** and set the Is Reference property to: **Center (Elevation)**.
7. Add a dimension between this new reference plane and Top and Sill and toggle on the Equality.



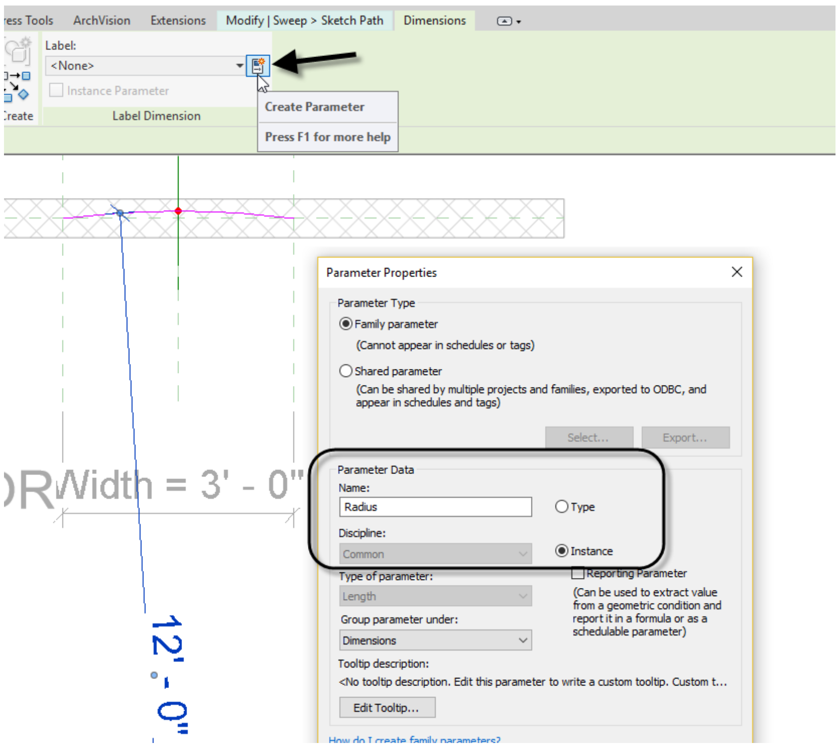
8. Return to the Ref. Level floor plan and then on the Create ribbon, click the Set Work Plane button.
9. From the Name list, choose: **Reference Plane: Center (Elevation)** and then click OK.
10. On the Create tab, click the Void Forms drop-down and then choose: **Void Sweep**.
11. Click the Sketch Path button.
12. Click the Start-End-Radius Arc icon.
13. Click the start point at the intersection of the Center (Front/Back) and Left reference planes.



14. Snap the other end to the intersection of the Center (Front/Back) and Right reference planes.
15. For the radius, type: **12'-0"** and then press ENTER. (Just start typing, it will automatically input to the radius).
16. Using the Align tool, click the center of the wall as the alignment and then the endpoint of the arc to align. Lock it. Repeat for the endpoint at the other side.
17. Align and lock the endpoints of the arc to the Left and Right reference planes. (Use the TAB key if necessary to assist in selection).


Walls : Basic Wall : Generic - 6" : Reference
<Sketch> : Line : Reference
Create or remove a length or alignment constraint
Reference Planes : Reference Plane : Left : Reference
<Sketch> : Line : Reference
Create or remove a length or alignment constraint

18. Click the Modify tool to finish aligning.
19. Select the arc, on the radius dimension that appears, click the “Make this temporary dimension permanent” icon.
20. Select the new dimension and then on the Label Dimension panel of the ribbon, click the Create Parameter icon.
21. In the “Parameter Properties” dialog, name the new parameter: **Radius**, choose Instance and then click OK.



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)

Select... Export...

Parameter Data

Name: Radius

○ 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

Tooltip description:

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

Edit Tooltip...

How do I create family parameters?

22. Deselect the dimension and then on the ribbon, click the large green Finish Edit Mode button.

23. Click the Modify | Sweep tab.

24. On the Profile panel, click the Load Profile button.

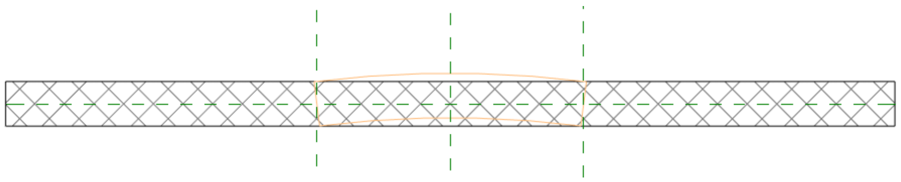
25. From the dataset folder, load the family named: *PRF_Opening Shape.rfa*.

Profile families are 2D shapes that can be used to create 3D forms. In this case, this family is a simple rectangle with flexible width and height. It will sweep along the curved path we just drew to create an arc-shaped void element.

26. Directly above Load Profile, from the drop-down list, choose the newly loaded profile.

27. On the ribbon, click the large green Finish Edit Mode button.

A tan colored void element will appear with a slight curved shape.



Notice that it is not thick enough to cut all the way through the wall and also that it is not yet cutting the wall. Let's adjust the size first.

28. On the Project Browser, expand the *Families* branch and then the *Profiles* branch.

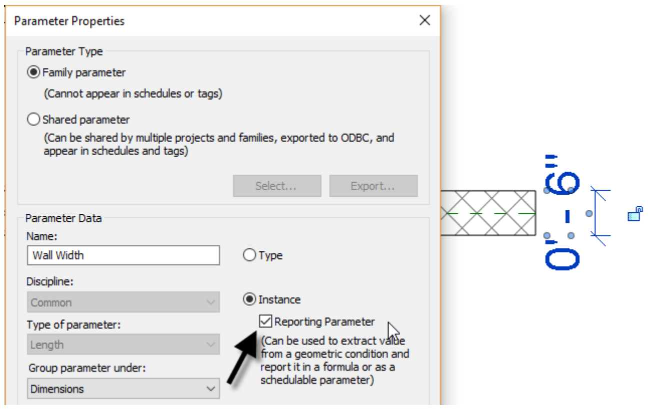
29. Expand *PRF_Opening Shape* to reveal a type named the same. Right click this type name and choose: **Type Properties**.

Notice the Associate Family Parameter buttons next to the parameters: H and W.

30. Click the Associate Family Parameter button next to H and then choose the Height parameter and click OK.
31. Click the Associate Family Parameter button next to W and then choose the Height parameter and then in the “Associate Family Parameter” dialog, click the New Parameter icon.
32. Name the new parameter: **Void W**, make it an Instance parameter and then click OK twice.

This creates a new parameter to drive the width of the void, but so far, its size is unchanged. Let’s make it equal to twice the width of the wall. To do that, we first need a reporting parameter to measure the size of the wall.

33. Add a dimension to the measure the width of the wall onscreen. Use the TAB key to assist in placing the witness lines.
34. Label this dimension with a new parameter called: **Wall Width**.
35. Set it to Instance and then check the “Reporting Parameter” checkbox. Click OK.


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)

Select... Export...

Parameter Data

Name: Wall Width

( ) Type

Discipline: Common

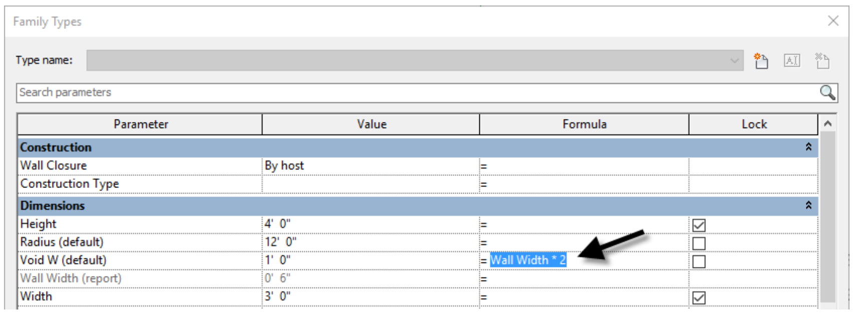
(●) Instance

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

36. On the Properties panel of the ribbon, click the Family Types button.
37. In the “Family Types” dialog, in the Formula column next to Void W, type: **Wall Width * 2** and then click OK.


Family Types

Type name:

Search parameters

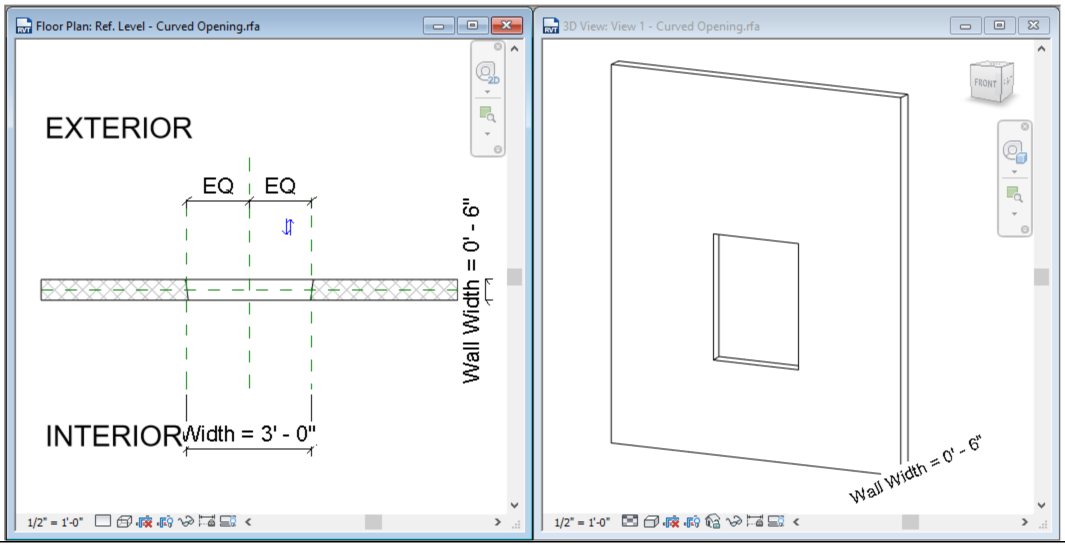
Parameter	Value	Formula	Lock
Construction			
Wall Closure	By host	=	
Construction Type		=	
Dimensions			
Height	4' 0"	=	☑
Radius (default)	12' 0"	=	☐
Void W (default)	1' 0"	= Wall Width * 2	☐
Wall Width (report)	0' 6"	=	
Width	3' 0"	=	☑

The void should now be twice the width of the wall.

38. On the Modify tab, click the Cut tool.
39. First click the wall, then click the void.

The void will now cut the wall.

40. Save the family and give it a name.



CATCH UP! You can open the file completed to this point named: ***Curved Opening.rfa***.

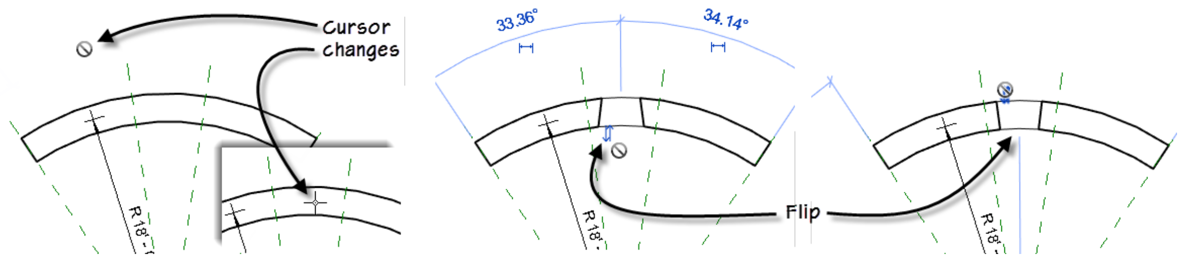
Associate a Global Parameter to a Radius Dimension

It's time to test out the family we just created. You can insert it into any wall. But for it to make sense, it needs to be inserted into a curved wall and the radius parameter we built into the family needs to match the radius of the host wall. This can certainly be accomplished by manually applying the radius to the opening element, but with a global reporting parameter, we can drive the value directly!

1. Open the file named: *08_Curved Openings_!Start.rvt*.
2. Open the *Level 1* floor plan.
3. On the Architecture tab, click the Window button and then on the Modify | Place Window tab, click the Load Family button.

4. Load your window family created above or the catch file: *Curved Opening.rfa*.
5. Click on one of the curved walls to place and instance.

Since we did not add any geometry to this family, there will be no preview other than a subtle change in the cursor when you get near an eligible host wall.

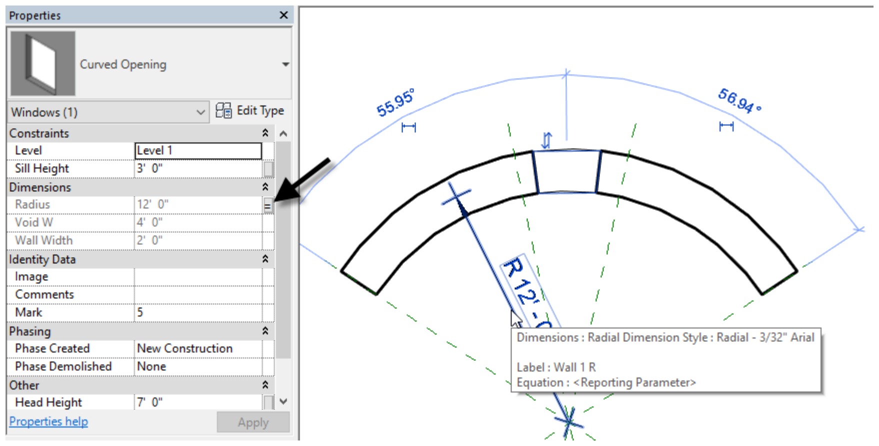


You might also need to flip the opening after placement to make it match the curve.

6. If required, flip the window. Place one in each of the three curved walls on the right.

Each of the three curved walls has a radius dimension. We can label these with a global reporting parameter to get the radius of the host wall. Then we can apply that to the openings.

7. Select the 12'-0" dimension.
8. On the Label Dimension panel, click the New Parameter icon.
9. Name the parameter: **Wall 1 R** and then check the "Reporting Parameter" checkbox.
10. Select the window instance in that wall. On the Properties palette, click the Associate Global Parameter button.
11. Choose: Wall 1 R and then click OK.



Constraints	
Level	Level 1
Sill Height	3' 0"
Dimensions	
Radius	12' 0"
Void W	4' 0"
Wall Width	2' 0"
Identity Data	
Image	
Comments	
Mark	5
Phasing	
Phase Created	New Construction
Phase Demolished	None
Other	
Head Height	7' 0"

Dimensions : Radial Dimension Style : Radial - 3/32" Arial
Label : Wall 1 R
Equation : <Reporting Parameter>

The other two dimensions are already labeled with global parameters: **Wall 2 R** and **Wall 3 R**.

12. Associate these parameters to the Radius parameter on each of the other two windows.

As you do this, you should see the windows adjust slightly as they conform to the new radius.

CATCH UP! You can open the file completed to this point named: ***08_Curved Openings_A.rvt***.

Now let's flex it.

13. Select the first wall. On the Options Bar, uncheck the "Keep Concentric" checkbox and then drag the middle control handle (open dot) to change its radius.

You should see the dimension update to a new radius value and if you look at the window, it has kept pace. Select it and look at the Radius on the Properties palette, and you will see it matches.

14. Select one of the windows (in any of the walls).
15. On the Properties palette, click the Edit Type button, increase the Width to 5'-0" and then click OK.

If you look at the other two windows (in the walls whose radii we have not edited) the width of the window should match the reference planes in the view. You can use the move or align tool to get them to line up precisely and thereby verify that the sides of the window are following the radius of the host walls. Feel free to experiment further and add geometry to the window if you wish. For example, if you want curved glass, you can copy and paste the same sweep in the family, change it to a solid. And then duplicate the profile type, and assign its width to the thickness you want for the glass. You will need to re-associate the parameters and align and lock the path of this copied sweep again. You should also assign a glass material to this new piece of geometry. You can also add 2D geometry in the plan view if desired and moldings and trim around the outside of the window if you like.

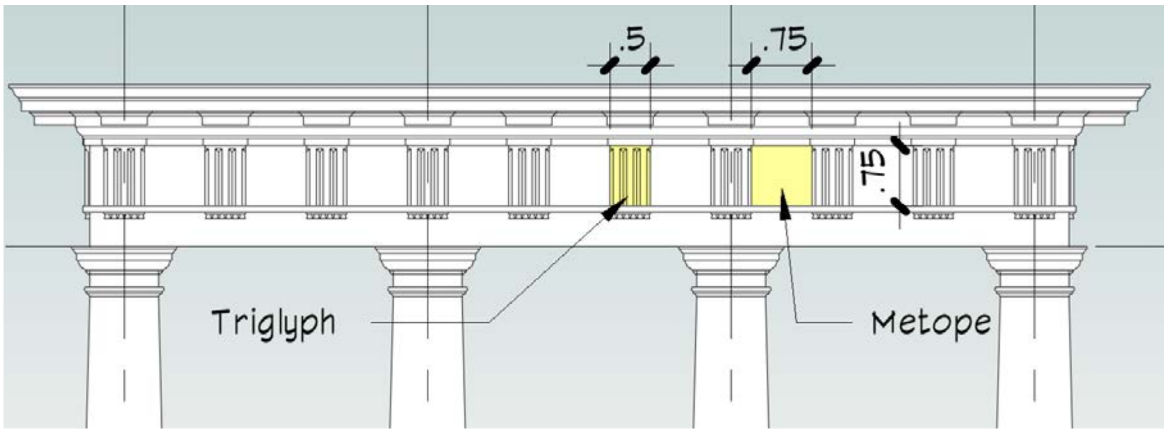
CATCH UP! You can open a sample file described here named: ***Curved Window.rfa***.

Column Spacing – intercolumniation

When I wrote Renaissance Revit, a book that takes a "deep dive" into the family editor and explores the creation of the classical orders of architecture as fully parametric families, we did not have global parameters. One of the challenges that was therefore left unsolved in that book was the connecting the proportions of various families to one another. For example, the orders utilize a series of interconnected proportions that are based on the diameter of the column measured at its base. This proportion is applied not just to the column and its base, capital and shaft, but also to the entablature and even the spacing of the columns relative to one another. In other words, the system of proportions will inform the entire façade, not just its columns. Without global parameters, changing the value of the base diameter in the column will not change the entablature or the column spacing. But with global parameters, it can! The problem and solution is similar to the "Driving Custom Curtain Panels" topic above.

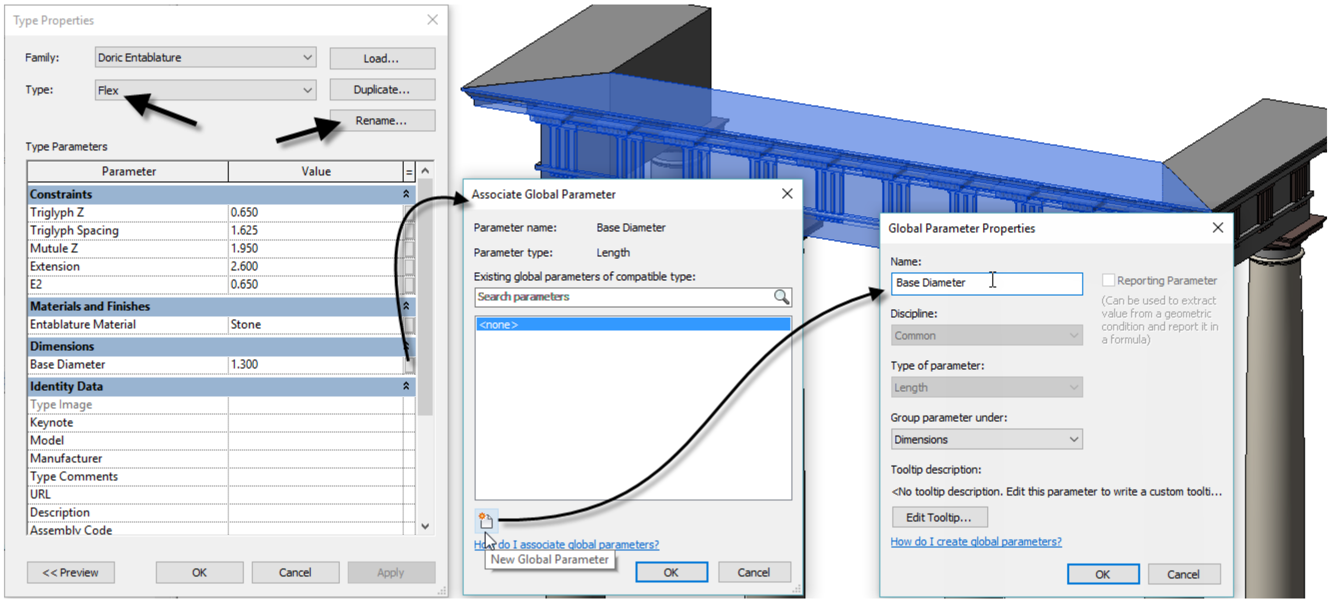
1. Open the file named: *09_Colonnade_!Start.rvt*.

This file uses the Doric order. The Doric order is perhaps the most stringent when it comes to it proportions. The most characteristic feature of the Doric order is its triglyphs and metopes. A triglyph is an adornment that is meant to represent the ends of a beam as they would appear projecting through the frieze of the entablature. A metope is the portion of the frieze between each triglyph. are precisely square in shape at 0.750 diameters per side. A triglyph is 0.750 diameters high by 0.500 diameters wide. In addition, you must have a triglyph over the center of each column. This means that only certain column spacings are allowable. Specifically: 1.25, 2.5, 3.75 and 5 diameters; measured center to center.



For this example, we'll start with a value of 3.75 diameters and use this to drive all of the other proportions.

2. Select an instance of the entablature family onscreen.
3. On the Properties palette, click the Edit Type button, click the Rename button and name it: **Flex**.
4. Click the Associate Global Parameter button next to Base Diameter. Click the New parameter icon and name it: **Base Diameter**.



Type Properties

Family: Doric Entablature

Type: Flex

Type Parameters

Parameter	Value
Constraints	
Triglyph Z	0.650
Triglyph Spacing	1.625
Mutule Z	1.950
Extension	2.600
E2	0.650
Materials and Finishes	
Entablature Material	Stone
Dimensions	
Base Diameter	1.300
Identity Data	
Type Image	
Keynote	
Model	
Manufacturer	
Type Comments	
URL	
Description	
Assembly Code	

Associate Global Parameter

Parameter name: Base Diameter

Parameter type: Length

Existing global parameters of compatible type:

Search parameters

<none>

How do I associate global parameters?

Global Parameter Properties

Name: Base Diameter

Reporting Parameter

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

Discipline: Common

Type of parameter: Length

Group parameter under: Dimensions

Tooltip description:

<No tooltip description. Edit this parameter to write a custom toolti...

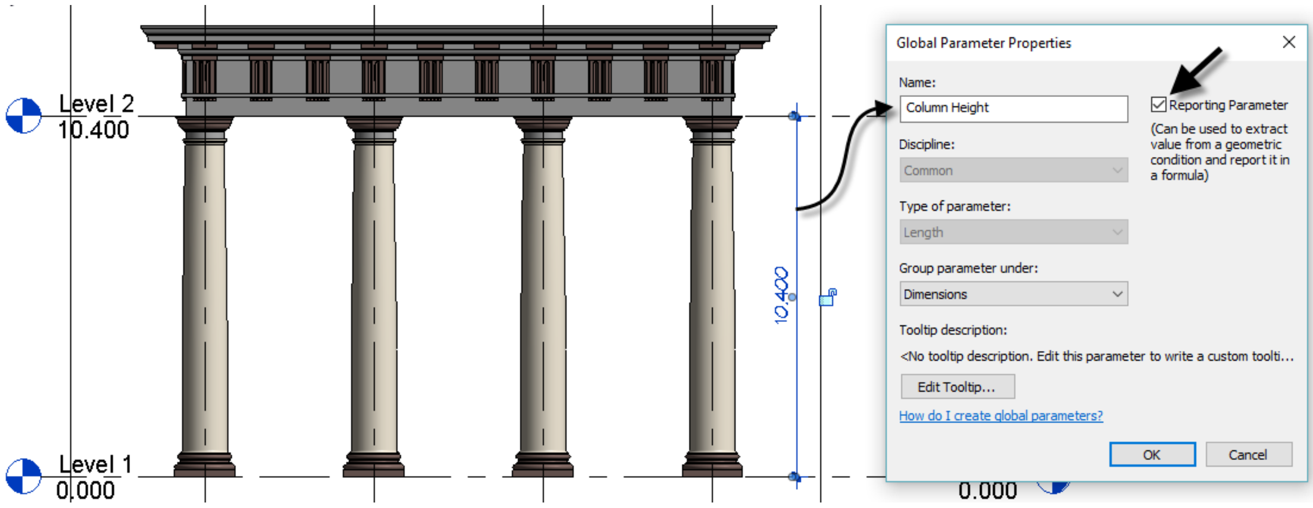
How do I create global parameters?

5. Click OK three times.

The Base Diameter of the columns is calculated automatically from the distance between the two levels. So we need a reporting global parameter for the distance between Level 1 and 2.

6. Open the *South* elevation.

7. Add a dimension between Level 1 and Level 2. Label it with a new global parameter.
8. Name the parameter: **Column Height** and then check the Reporting Parameter checkbox.



Global Parameter Properties

Name: Column Height

☑ Reporting Parameter (Can be used to extract value from a geometric condition and report it in a formula)

Discipline: Common

Type of parameter: Length

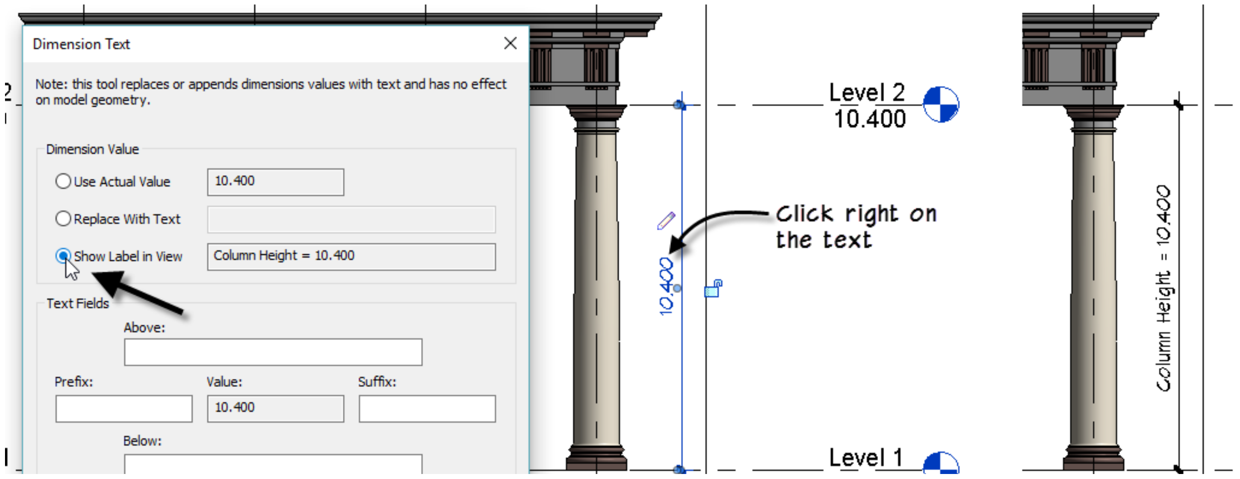
Group parameter under: Dimensions

Tooltip description: <No tooltip description. Edit this parameter to write a custom toolti...

9. Click OK to complete the parameter.

In the family editor, when you label a dimension it shows the name of the parameter on the dimension. Global parameters do not this by default, but you can override them to display this way if you choose.

10. Click directly on the text value of the dimension.
11. In the “Dimension Text” dialog, choose the “Show Label in View” option and then click OK. (You can also do this on the Properties palette).


Dimension Text

Note: this tool replaces or appends dimensions values with text and has no effect on model geometry.

Dimension Value

○ Use Actual Value 10.400

○ Replace With Text

● Show Label in View Column Height = 10.400

Text Fields

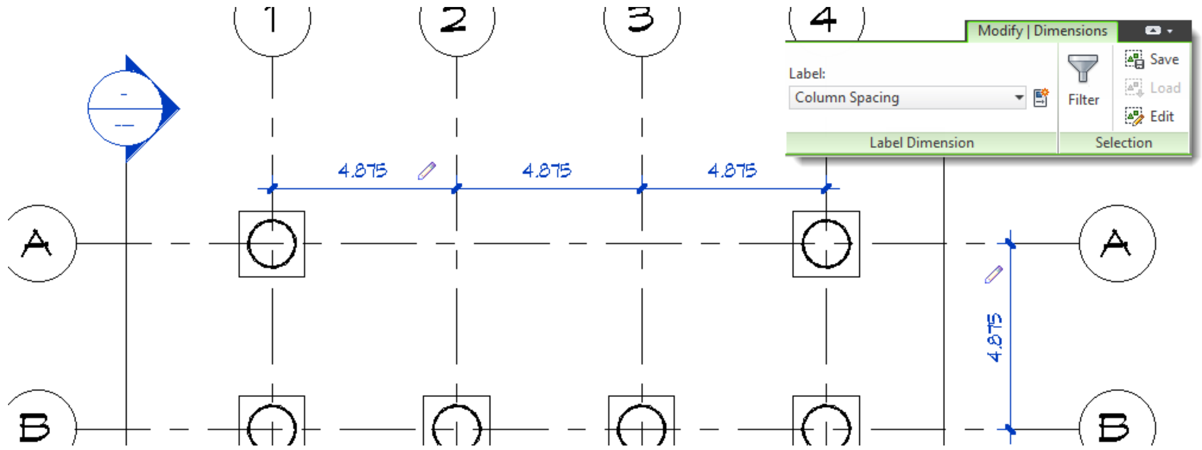
Above:

Prefix: Value: 10.400 Suffix:

Below:

12. Open the *Level 1* floor plan view.
13. Select the equality dimension and then toggle off the equality.

14. Hold down the ctrl key and add the other dimension (between column A and B) to the selection. (2 dimensions).
15. Create a new global parameter to label the selected dimensions and name it: **Column Spacing**.

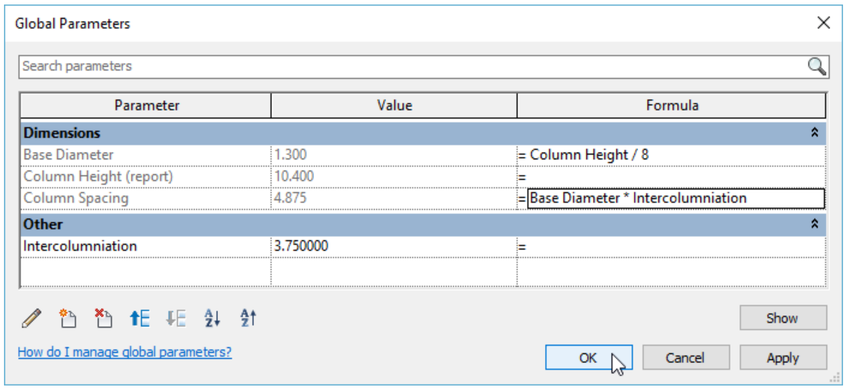


We need one more parameter.

16. Open the "Global Parameters" dialog and click the New Global Parameter icon.
17. Name the new parameter: **Intercolumniation** and change the Type of parameter to: **Number**.
18. Click OK to create the parameter and set its value to: **3.75**.

That's all the parameters we need. Now we just need a few formulas. The Doric order is eight diameters tall.

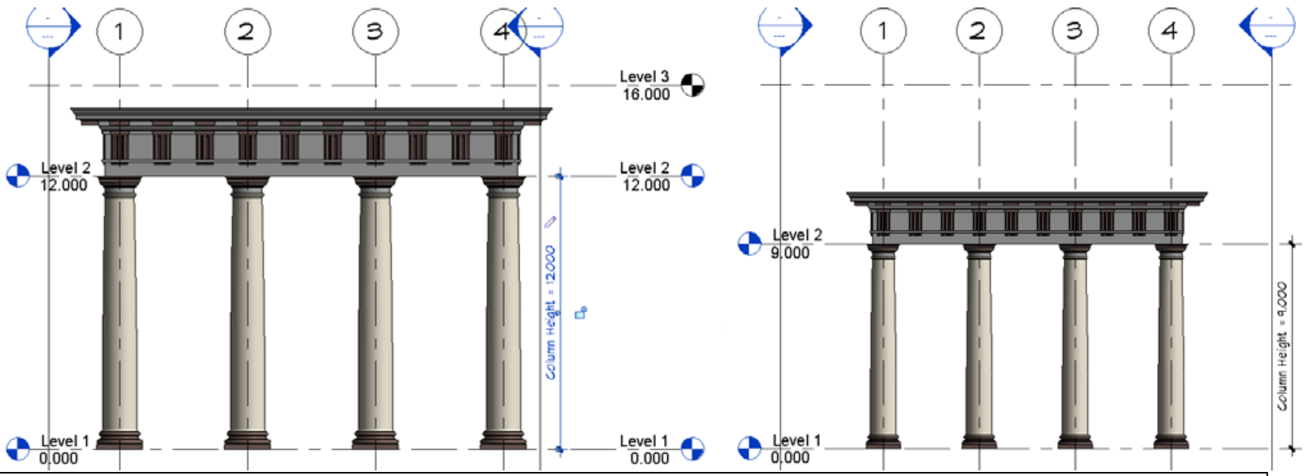
19. For the Base Diameter, in the formula field type: **Column Height / 8**
20. For Column Spacing, input a formula of: **Base Diameter * Intercolumniation**



Parameter	Value	Formula
Dimensions		
Base Diameter	1.300	= Column Height / 8
Column Height (report)	10.400	=
Column Spacing	4.875	= Base Diameter * Intercolumniation
Other		
Intercolumniation	3.750000	=

That's it! To flex we just adjust the location of Level 2.

21. Edit the height of Level 2 and change it to: **12**. Try another value, such as: **9**.



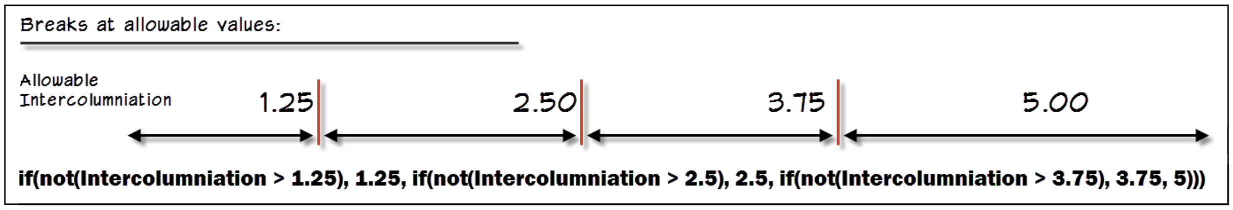
CATCH UP! You can open the file completed to this point named: ***09_Colonnade_A.rvt***.

Bonus

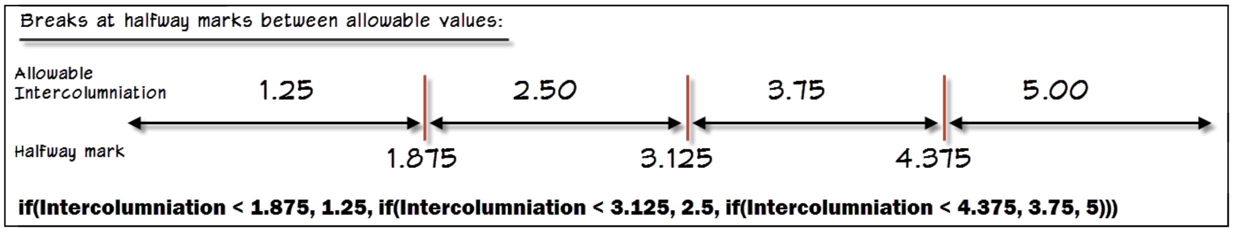
At the start of the exercise, I noted that only four spacings were allowed in the Doric order. While everything here is working nicely, there is nothing to prevent your inputting an Intercolumniation value that is not allowed. Using some additional formulas, we can prevent this from happening and limit the choices to those four that are allowed. This will require a nested "if" formula. First, we need to create a new parameter that will evaluate the input of the Intercolumniation.

22. Create a new Global Parameter set to Number and called: **Intercolumniation Check**.

23. Input one of the two following formulas:



if(not(Intercolumniation > 1.25), 1.25, if(not(Intercolumniation > 2.5), 2.5, if(not(Intercolumniation > 3.75), 3.75, 5)))



Breaks at halfway marks between allowable values:
Allowable Intercolumniation
1.25
2.50
3.75
5.00
Halfway mark
1.875
3.125
4.375

if(Intercolumniation < 1.875, 1.25, if(Intercolumniation < 3.125, 2.5, if(Intercolumniation < 4.375, 3.75, 5)))

if(not(Intercolumniation > 1.25), 1.25, if(not(Intercolumniation > 2.5), 2.5, if(not(Intercolumniation > 3.75), 3.75, 5)))

if(Intercolumniation < 1.875, 1.25, if(Intercolumniation < 3.125, 2.5, if(Intercolumniation < 4.375, 3.75, 5)))

The first option makes the breaks at the allowable values. Revit does not have a formula for "less than or equal" so to achieve that, you nest a Not formula into the If formula. So by saying "not(Intercolumniation > 1.25) it is the same as saying less than or equal to 1.25. So in the first example, any value higher than this, even 1.26, would go to the next allowable value: 2.5 in this case.

If you prefer to make the break in the middle, you can figure out the halfway marks (noted in the figure) and then use those values in the formula instead. You could use the nested Not formula again here, but since we are halfway between values it seems less necessary.

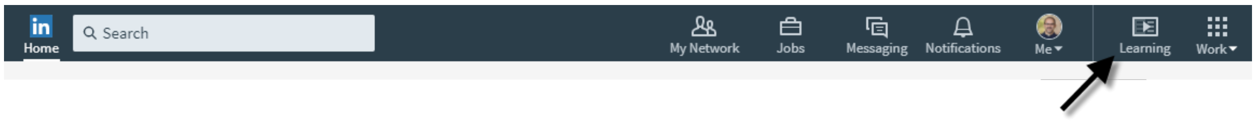
CATCH UP! You can open the file completed to this point named: ***09_Colonnade_B.rvt***.

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.

Lynda.com has recently been rebranded as LinkedIn Learning, so if you are Professional member of LinkedIn, you already have access to the entire training library. You can access it directly from LinkedIn.



I have several Revit courses including: Revit Essential Training, Revit Family Editor and Revit Architecture Rendering, Advanced Modelling in Revit Architecture, Formulas and Curves and many more. Recently we released a course on Revit View Range, a three-part series on Worksharing including Collaboration for Revit, a course on Revit Areas and 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.