



AS10688

Revit Master Class: Building Terrific Real-World Door Families in Autodesk Revit

Matthew Stachoni
Microsol Resources Corporation

Learning Objectives

- Learn how to efficiently build real-world door components in the Family Editor
- Combine components into assemblies that are flexible and information rich
- Discover the secrets to hosting components to various references to effectively position them
- Be able to best capitalize on family, shared, and project parameters to report door components in tags, schedules, and reports

Description

Even for the best Revit family modelers, door families are some of the toughest to create effectively. With all of the moving parts, data requirements, and end-user demands, building out a comprehensive door library is always a challenge. In this class we take a thorough step-by-step approach to building a series of door components and combining them into assemblies for use in design, documentation, and construction administration. By focusing on these 3 use cases, we explore how to capitalize on all of the tools in the Family Editor to create doors that are easy for the end user to use, look terrific in rendering and animations, and document themselves efficiently in tags, schedules, and reports.

Your AU Experts

***Matt Stachoni** has over 25 years of experience as a BIM, CAD, and IT manager for a variety of architectural and engineering firms, and has been using Autodesk, Inc., software professionally since 1987. Matt is currently a BIM Specialist with Microsol Resources, an Autodesk Premier Partner serving New York City, Philadelphia, and Boston. He provides training, BIM implementation, specialized consultation services, and technical support across a wide array of Autodesk AEC applications.*

Previously, Matt was the BIM and IT Manager for Erdy McHenry Architecture LLC, responsible for handling all digital design application efforts and IT support. As a BIM specialist for CADapult Ltd. (now Applied Software) he provided on-site construction BIM modeling and coordination services for construction managers, HVAC, and MEP trade contractors for a number of projects. He is a contributing writer for AUGIWorld Magazine, and this is his 12th year speaking at Autodesk University.



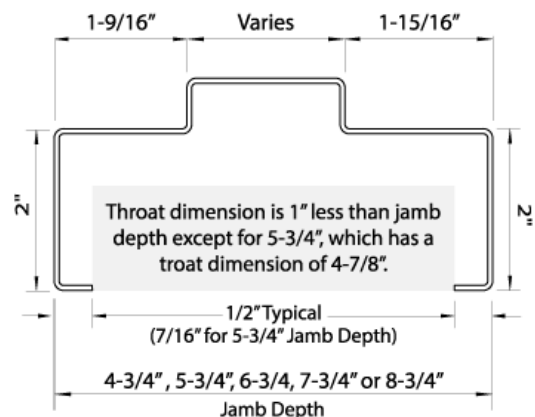
I. Introduction

This class is designed to show you how to design and build door families which are effective, easy to use, and contain the information required for your schedules to function properly. There are a lot of steps to building families of any kind, but this class is not meant to be a start to finish narrow-focused series of tutorials. Instead, it is best considered to be a rundown of the ideas and specific door use cases you are likely to run into in commercial construction, and how to best address them from a Revit family point of view. It is assumed that you understand the basics of building Revit families, but need some guidance in how to get started building out what can be a very complex company door family library.

Doors are really best thought of as complete assemblies of interdependent elements, and this class will use the term assembly to refer to the complete door family package you would place in a wall. A door assembly taken as a whole is defined by the elements in the door; a solid-core wood door with a hollow-metal frame, for example, could be one assembly. A double aluminum storefront entrance with exit hardware would be another. Door assemblies are what get scheduled, not the parts inside.

The door assembly defines the materials and sizes of the dependent door panel(s), the frame, the hardware package, the glazing, and other construction specifics. In turn each subcomponent has its own parameters, some of which are driven by the host assembly, others by the manufacturer, and others independently of either.

The nature of how doors function as a collected assembly of parts will drive your logic in creating the required Revit sub-component families. This underpins much of the resultant door library to make them either easy or hard for end users to use throughout many projects.



For example, a hollow metal door frame is commonly specified by manufacturers to be in several preset sizes, from 4-3/4" to 8-3/4" jamb depths in 1" increments. The throat dimension (the back opening) is 1" less than this jamb depth and has a backbend dimension of 1/2". Except for 5-3/4", which has a throat dimension of 4-7/8", or a backbend dimension of 7/16".

Because the HM frame is in (5) specific sizes with geometry for the rabbets, backbends, and throat, this automatically triggers the logic that infers that the frame is best implemented as a family with 5 types, and that the choice of frame is governed by a Family Type parameter in the assembly. Otherwise there really isn't any way to maintain the frame size integrity – you could just specify any frame size.

Our goal in this class is to deliver door families that heed the following specifications:

1. They should ultimately be easy to use for the end user. Families that have a convoluted set of parameters, are bloated with numerous types, or are inelegant will not gather many fans. Users may be tempted to visit Revit content warehouse sites that they should not *cough revitcity*cough for content.
2. They need to look great in 3D. Your goal is to be able to drop in any door from your library and show it off in a rendering or animation with pride. Doors that are missing push/pull hardware, do not swing open, or have the wood grain running in the wrong direction just look wrong, even to unsophisticated owners. And the number of unsophisticated owners and end users is shrinking every day. Everyone



is getting used to seeing photorealistic renderings that are indistinguishable from a photograph. That level of realism is only achieved when the eye doesn't catch something it knows is out of place.

3. They need to follow your company's 2D graphics standards for lineweight, linetype, and detail level. Set standards in place for what should and should not appear in 2D views.
4. A 2D door in plan is strictly symbolic in nature, even if it does provide clues to details. No 3D elements should be visible in your floor plan. 3D objects are visible in elevations and sections.
5. All door components should be of the Doors category and 3D elements assigned to Door subcategories. New non-default subcategories should be defined and standardized early on to allow door libraries to establish themselves.
6. There need to be standards in place for what can and what cannot be changed within a single family, and the standards should apply to all doors used within your organization. Door assembly families need to be flexible on an instance basis in certain respects, but be standardized at the type level where required. Going back to the hollow metal frame example, you may specify that the door frame is an assembly level standard, and have an assembly (family) that specifies a hollow metal, double rabbet frame. You could easily have several assemblies with hollow metal frames, with varying door panel configurations, hardware, security devices, and so on.

Another assembly might have a hollow metal single rabbet frame, and another would have a wood frame. Thus there would be three assemblies – families – each with their specific style frames. However, the depth of the frame would be driven by the family instance, e.g. there is a hollow metal frame but the frame type could be one of five size choices.

To create door families properly, one needs to have a modicum of moderate to advanced skills in working in the Family Editor. To that end it is important to first review the “rules of the road” in building solid families of any kind.



II. A 13-step program for building families that work

It is no secret that building families correctly is an essential Revit skill. These 13 steps are the secrets to successfully building component families that just work. Not all steps will be required for all kinds of families; in particular simpler families such as annotation, detail components, and profiles only require some of them. But when you get into complex 3D components that need to look good, accurately report information, and behave in all views, you need to take all of these steps into account.

1. First and foremost, **Plan** the family's required and requested behaviors. Think about how flexible it needs to be. Determine if and how it needs to be hosted, if it is to be nested, what constraints and parameters it needs, and so on. Sketch it out on paper if need be.
2. Choose the family template which determines the category / hosting condition and initial setup (scale, units, etc.). Remember that you can choose the Generic Model category first then reassign the category using the Family Categories and Parameters dialog. Build your own family templates.
3. Draw Reference Planes that will block out the geometry and define constraints. Do this in plan, front, and either left / right elevations. Try to place as many related reference planes as you can in one view to make them easy to find.
4. Dimension the Reference Planes and Label them to associate the dimension with an existing or newly created family parameter. "**Flex**" the parameter, ensuring that by changing the parameter the reference planes move as predicted. Locking a dimension ensures the geometry does not move.
5. Model your geometry and constrain it to the Reference Planes using align / lock. Flex the behavior by entering values into the parameters to ensure it does not break in real world use.
6. Create additional parameters for other properties of the family, such as materials. Give them a name, data type, and assign them to a parameter group.
7. Use formulas to generate parameter values and/or relate one parameter to another, using Revit's formula syntax for logical and mathematical operations. Pay attention to parameter capitalization.
8. Create Family Types, and assign Type and initial Instance parameter values. Tip: when you have an Instance parameter with a family of several types, set the parameter to a valid default value in the Formula column and hit Apply. That will write it to all types at the same time. Otherwise you may end up with a type with an invalid value for the instance parameter (a zero length, for example) and generate an error. Once tested, delete it from the Formula column which makes the values across all types now independent.
9. Create and assign Subcategories to geometry. Note: This only works on raw geometry, not on nested families. Solution: Create the nested family of the same category as the parent family. Subcategory assignments follow the family into host families and into the host project, like a layer inside of a block in AutoCAD.
10. For nested families, make their parameters Instance based, which makes it easy to link to parameters in the host family. For Profile families that may be nested and used to generate a Sweep, use Type Parameters to define the profile size. Once loaded into the host family, you can link that family type parameters to the host's parameters.



11. Leverage 2D symbolic linework, filled and masking regions, and detail components to remove 3D geometry from 2D views wherever possible. Make heavy use of detail component families (which can be parameterized) and nest them.
12. Set Detail Level visibility of objects to Coarse / Medium / Fine. This is especially important for mullion profile detailing.
13. Always **Flex** the parameters in the Family Editor and test in the project context. **Do not test your families in your active project.** Use a temporary dummy project to test things out.

III. Door families we will create in this class

We are going to cover a broad range of commercial, hospitality, and institutional door families that should cover quite a lot of the kinds of doors you use on a daily basis. These are designed to demonstrate various advanced family building techniques that could be applied in other kinds of families as well. The doors we will create are:

- Typical commercial interior solid core wood door panel with an HM frame. Options include lights in the door and sidelight frame with glazing;
- Exterior curtain wall / storefront entrance door with extruded aluminum door panel and rail construction, with a sub-frame or applied stops;
- High-end hospitality paneled door with a HM frame with an elaborate cornice and jamb trim;
- Frame and panel solid wood door with wood frame;
- Frame and panel solid wood louvered door with wood frame

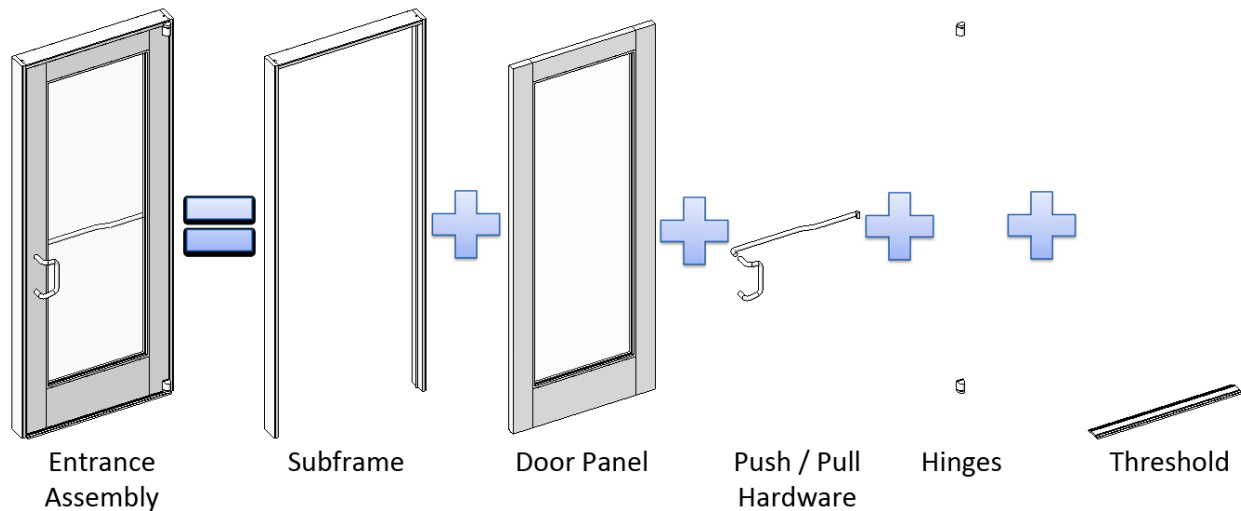
IV. Practical considerations for designing door families

Before we begin creating any family components, it is important to map out how the door is to function in our projects and what information is required to be scheduled and mapped. These standards will drive future development as well. But what makes a good, practical standard for door families? The first thing is that it has to take into account, within reason, the most complex conditions you will come across, yet remain accessible for more typical projects. For doors, this includes being able to handle the following conditions:

- Double doors with unequal panel widths
- Double doors with different panel types
- Door assemblies that can vary their placement within the wall from exterior to interior, and even be placed completely outside of the host wall to work with multiple walls layered together
- Drywall frames that wrap the wall construction vs. frames that lie completely inside the R.O.
- Masonry frames with 2" and 4" head face widths
- Hollow metal sidelights with variable glazing widths
- 2D doors swings which have separate graphics for coarse detail level as well as variable swing angle
- 3D door panels which have a variable opening angle
- Schedule material, finish, and type for panels and frames separate from the Revit material assignments that affect shading and rendering
- Assignment of various hardware types, such as locksets, hinges, and thresholds



Taken together this list of design requirements drives the family construction in a number of ways. First, because we need to be able to swap out door and panels within a single family, it makes sense to build our door assemblies as a collated group of nested families for everything. Panels, frame, push/pull hardware, hinges, glazing, and even 2D swings should be made as separate Door families, nested into the main assembly family, and have their parameters linked. The main assembly family will have no native geometry of its own.



Note: These sub-families need to all be created as Door families, not Generic Models or something else. This is because only by creating the geometry in the Doors category can we assign it to a Door Subcategory. Subcategories allow us to manage visibility, material assignments, and detail level in a highly granular fashion which can be rolled into View Templates to help manage views inside and across projects.

Building “SuperDoor” families

One technique that is sadly somewhat common is to build “SuperDoor” families that are composed of every type of door imaginable in one family. Wood / metal frames, solid core and frame /panel construction, frames with sidelights vs. no sidelights, can theoretically be encapsulated in a single family.

While this cuts down on the families, other things suffer. The complexity of the parameters required across all of those different constructions can be mind-boggling. The weight of the door families themselves can be an issue, as a small change may need to propagate across all doors in the project.

The other issue is with Family Types. With a SuperDoor family you end up with a family type for each construction type which can get to be very long. Families with many types don’t operate as efficiently as those with fewer types. That also means that traditional door type parameters such as Width and Height now need to be Instance parameters. That is not bad, per se, but it depends on how much you would use things like selecting all doors of a certain size – easy when sizes are types and using Select Similar.

For the purposes of this class, we will eschew building multi-construction families in lieu of having a single family be of a particular construction, and using Type parameters for height and width.

V. Shared parameters for door assemblies

The incorporation of Shared Parameters for door assembly families is critical because, by default, Revit does not provide all of the data necessary for scheduling and tracking of information. My approach is to create all parameters as Shared Parameters, so that everything about the door can be scheduled and managed in the project environment. Not all families will use all parameters.

NAME	DATATYPE	DESCRIPTION
Detail Head	Text	Head detail #/A-###
Detail Jamb	Text	Jamb detail #/A-###
Detail Sill	Text	Sill detail #/A-###
Frame Centered In Host	Yes / No	Checked = center frame in host. Unchecked = use "Frame Offset" value
Frame Depth	Length	Depth of frame
Frame Drywall Slipon	Yes / No	If checked, frame uses drywall slipon dimensions and decreases the rough opening to half of the frame face width
Frame Family Type	<Family Type...>	Frame family type
Frame Head Face Width	Length	Face width of frame at the head
Frame Jamb Face Width	Length	Face width of frame at jambs
Frame Overall Height	Length	Outside height dimension of frame
Frame Overall Width	Length	Outside width dimension of frame
Frame Pull Side Offset	Length	Offset distance of frame from pull side of wall. Positive values push frame into the wall, negative values pull frame outside of wall. Unused if 'Frame Centered on Host' is checked.
Frame Revit Material	Material	Revit material assigned to frame
Frame Scheduled Finish	Text	Scheduled finish of frame
Frame Scheduled Material	Text	Scheduled material of frame (HM, WD, ALUM, etc.)
Frame Scheduled Type	Text	Scheduled frame type
Frame Sidelight Sill Face Width	Length	Sidelight frame sill face width
Glazing Frame Revit Material	Material	Revit material assigned to the door panel glazing frame
Glazing Revit Material	Material	Revit material assigned to the glazing
Glazing Scheduled Material	Text	Scheduled glazing material
Glazing Scheduled Type	Text	Scheduled glazing type
Hardware BHMA Finish Number	Text	BHMA standard number for hardware finish
Hardware Hinges Family Type	<Family Type...>	Hinges family type
Hardware Revit Material	Material	Revit material assigned to all hardware excluding lockset
Hardware Scheduled Devices And Sets	Text	Scheduled device and hardware designations
Hardware Threshold Family Type	<Family Type...>	Hardware threshold family type
Lockset Backset	Length	Lockset backset from latch side edge
Lockset Elevation	Length	Elevation of lockset center from floor
Lockset Family Type	<Family Type...>	Lockset family type

NAME	DATATYPE	DESCRIPTION
Lockset Revit Material	Material	Revit material assigned to lockset
Panel 1 2d Swing Angle	Angle	2D swing angle for panel 1
Panel 1 3d Swing Angle	Angle	3D swing angle for panel 1
Panel 1 Family Type	<Family Type...>	Panel 1 family type
Panel 1 Width	Length	Nominal width of panel 1
Panel 2 2d Swing Angle	Angle	2D swing angle for panel 2
Panel 2 3d Swing Angle	Angle	3D swing angle for panel 2
Panel 2 Family Type	<Family Type...>	Panel 1 family type
Panel 2 Width	Length	Nominal width of panel 2 if different from panel 1. Unused if 'panels equal' is checked.
Panels Equal	Yes / No	If checked, width = 2 * panel 1 width. If unchecked, width = panel 1 width + panel 2 width
Panels Louver Revit Material	Material	Revit material assigned to panel louvers
Panels Rails Bottom Width	Length	Panel bottom rail width
Panels Rails Cross Width	Length	Panel cross rail width
Panels Rails Lock Width	Length	Panel lock rail width
Panels Rails Mullion Width	Length	Panel mullion rail width
Panels Rails Revit Material	Material	Revit material assigned to panel rails
Panels Rails Top Width	Length	Panel top rail width
Panels Revit Material	Material	Revit material assigned to panel
Panels Scheduled Finish	Text	Scheduled panel finish (STAIN, PAINT, ANOD, etc.)
Panels Scheduled Material	Text	Scheduled panel material (WD, HM, ALUM, etc.)
Panels Scheduled Type	Text	Scheduled panel type
Panels Standard Thickness	Yes / No	Checked, panels are 1-3/4" thick. Unchecked, panels are 1-3/8" thick.
Panels Stiles Revit Material	Material	Revit material assigned to panel stiles
Panels Stiles Width	Length	Panel stile width
Panels Undercut	Length	Undercut distance below panels. Included in nominal panel height.
Walls Exterior Wrap	Length	Wrap distance for exterior layers of wall
Walls Interior Wrap	Length	Wrap distance for interior layers of wall

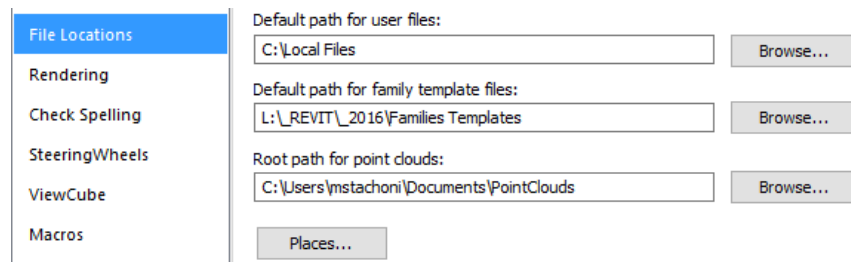
For frames, panels, and hardware, we will assign each as a <Family Type> parameter. That way we can swap them out for different family types. Note that most of the families are going to be driven by Instance parameters which are linked to these Family parameters. In most cases the family parameters drive the nested families.

VI. Building door components

In this section we start to build out the door families and the subcomponents that make them up. This is not an exhaustive tutorial; I will only summarize the family and explain any special thinking behind the family geometry, parameters, issues, configuration, etc. All of these families are available as part of the dataset additional materials download, so you can open the families up for yourselves to check out, fold, spindle, modify, etc. Feel free to create your own variations based on these.

Set up your Families Template Library

To make it easier to build families, you are going to need some custom templates. This will save a lot of time in setting up initial scale, lineweights, units precision, line styles, and so on. To do this set aside a folder on your server called Family Templates and point to it under Options > File Locations > “Default path for family templates files.”



Building Workable Door Family Templates

The first thing to do is build some door templates that will work for our needs. For the subcomponents of the door assembly, we need to use a family of the category Doors in order to assign the proper Subcategories. However, we do not want the wall host that is in the Door template, so we need to make one from scratch.

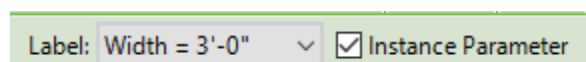
To do this, start with the Generic Model.rft template. This is an unhosted template and is as simple as it gets. Go to Family Category and Parameters and change the category to Doors and hit OK. Create a default family type called “Standard.” I recommend you have a good default name in all families so you don’t end up with things called Family Name:Family Name in the project, so it makes sense to put them in the family templates.

Save it to your Families Templates library as “Door – Unhosted Start.rfa” and in Windows Explorer, change the extension to .rft

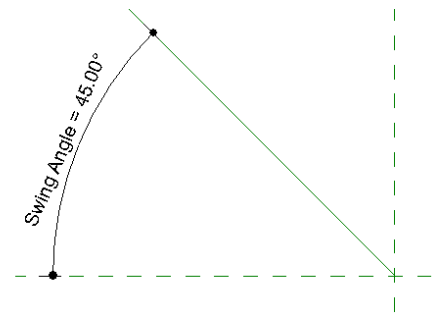
Door Swing 2D.rfa

The first family to build looks like the simplest but is actually one of the more perplexing ones to create. This family is the 2D door swing plan symbol. While deceptively simple, you have to pay attention and really follow the instructions to the letter to make it work, because Revit doesn’t get along with angles as well as it should. The first thing to do is create a Door family using the “Door Unhosted – Start” template. Even though it uses 2D graphics it is actually a 3D door family.

Start by creating a line 3’ off of a reference plan. Dimension it and label it with the Width parameter. In the options bar, change it to Instance.

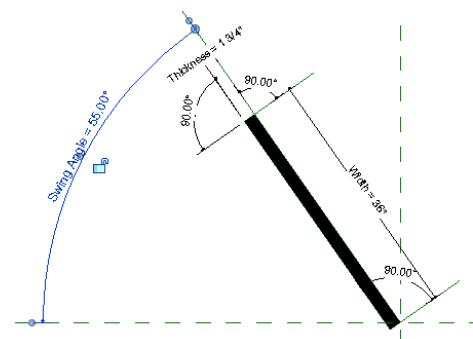


Lay out a reference plane from the origin to some point to the upper left of the screen. This will be the plane of the outer side of our door swing, Lock its end point to both reference planes. Create an angular dimension specifying its angle from the vertical plane. Test it to see that it does not break, especially at 90°.



Now we add the geometry for the door panel and swing linework. For a coarse level of detail, we want a filled region of solid black as the panel; in medium and fine it will be an open rectangle. In coarse we will use a straight line to the curve endpoint. In medium and fine we will use an arc.

Create a new Detail Item family, give it a new type called "Standard" and save it as "plan swing solid fill.rfa." Create a new reference line from the origin to the top left as you did before, at least 50" long. Lock the endpoint to the reference planes. Set the active working plane to this reference line's flat plane.



Offset it 2" down, and connect the top and bottom edges to form a rectangle of reference lines. Dimension the 90° angles on 3 sides. Dimension the angle from the horizontal to the first angled reference line and label it as a "Plan Swing" parameter. Text the parameter so the whole assembly should rotate orderly.

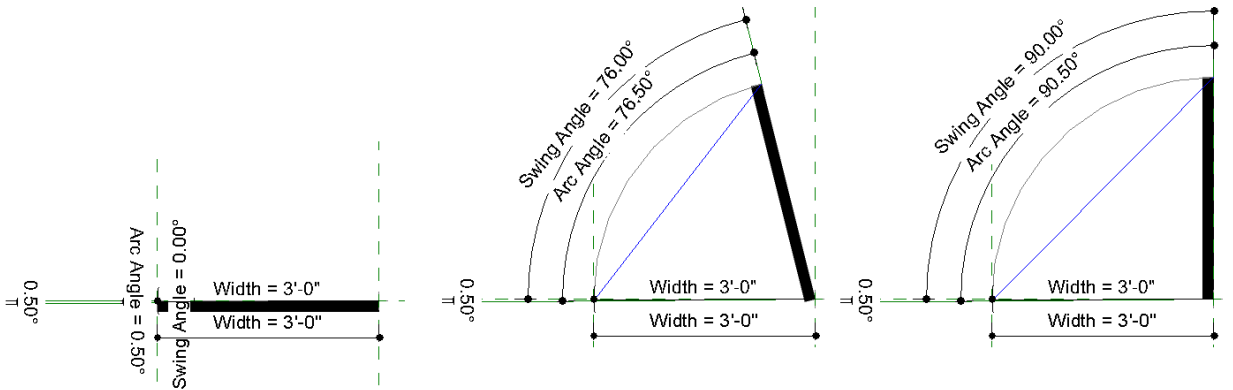
Draw a filled region using the four reference lines as the boundary. Align and lock the edges to the reference lines. Test it again. Load it into the Door Swing 2D family. Align and lock it to the central reference planes, and link the angle, thickness, and width parameters to those in the Door Swing 2D family. Set its Visibility Settings to show in a Coarse detail level only. Flex the parameters in the family to make sure everything works.

Draw a masking region by aligning it and locking it to the reference planes inside of the detail item. Flex the family again to make sure everything works. Set its visibility settings to Medium and Fine.

Draw a reference line from the origin to the left but at an angle south of the horizontal plane. Dimension the angle, select the reference line and set the angle to 0.5°. Create a reference plane left and parallel to the vertical plane. Dimension it and label it as the Width parameter.

Draw an arc from the center origin to act as the swing arc. Turn its center marks on and align / lock it to the center origin. Align and lock the left point to the new reference line and drag the end grip to meet the corner of the black solid fill. Select it and make the radius dimension permanent, and tie to the Width parameter. Create a new Angle parameter called "Arc Angle" under Other and put in the formula " $=0.5^\circ + \text{Swing Angle}$." Test all swing angles from 0 - 180°

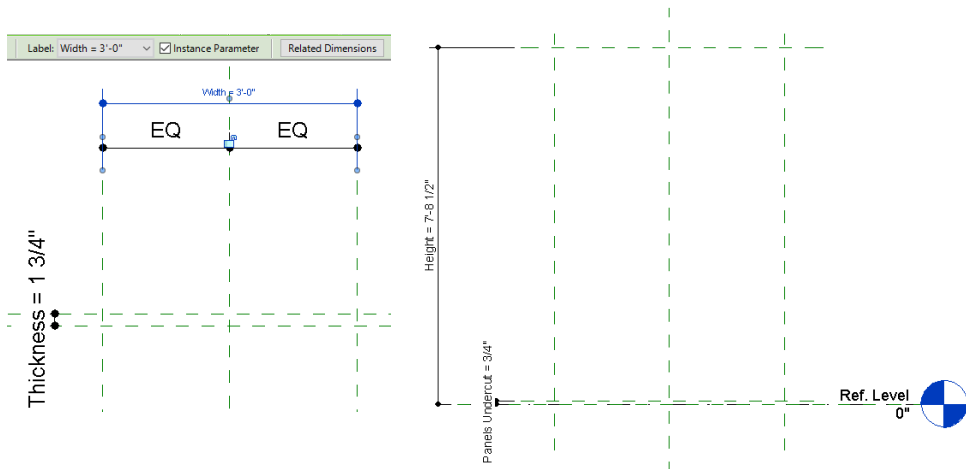
Make the swing angle 90° again and isolate the swing arc. Draw a straight line from one end to the other end. Unisolate it and they should stick together through all angles. Set its visibility settings to Coarse only.



Now you have a door swing which functions well and looks good at all angles.

Building a Door Panel

Compared to a door swing, a panel is pretty easy. Start with the Door Unhosted-Start.rfa and create some reference planes on either side of the center vertical, a plane underneath the horizontal one, and in elevation create two planes above the reference level:



In plan, equalize the dimensions and dimension / label across with the Width parameter. Dimension and label to the bottom reference plane as Thickness. Make them both Instance Parameters in the Options Bar. In elevation, make a reference plane close the ground and label it with the Shared Parameter “Panels Undercut”. Dimension / label the Height parameter that goes from the reference level to the top reference plane. Name all reference planes as Left, Right, Top, and Undercut. Name the upper horizontal plane as Panel Exterior and the lower one as Panel Interior. Make the Left, Right, and Top reference planes Left, Right, and Top Strong References.

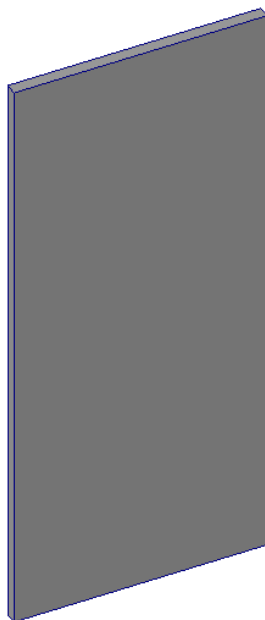
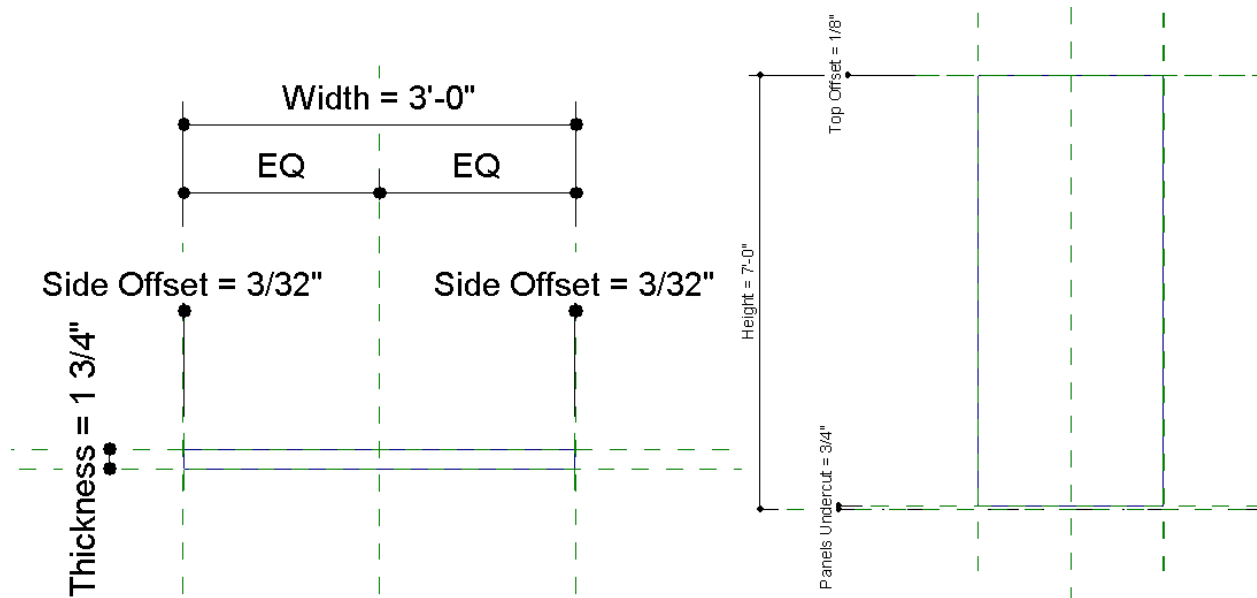
The Width and Height parameters come in as Type parameters but can be changed to Instance in the Options Bar. You want to do this so they can flex with the host assembly family.

Create three offset reference planes, inboard from the Top, Left, and Right reference planes. Label the right and left offsets as a Type Parameter called “Side Offset” with a 3/32” gap. From the front elevation, make a “Top Offset” gap 1/8”, also a new Type parameter.

In plan, create a new extrusion. Lock the sketch to the Side Offset planes and the front/back planes. In the front elevation align and lock it to the Undercut and Top Offset planes.

Select the Panel and set it to the Panel subcategory. Click on the Material button and create a new family Shared Parameter called "Panels – Revit Material." Also create the Shared Parameter called "Panels Scheduled Material" and group it under Materials as well.

Create a new family type called Standard. Save it as a new family prototype (not a template!) called Door Panel Unhosted – Start.rfa. Do another SaveAs as "Panel – Single Wood Flush" so you don't lose your starting point.



Type name: Standard

Search Parameters

Parameter	Value
Construction	
Construction Type	=
Function	=
Materials and Finishes	
Panels Revit Material	<By Category> =
Panels Scheduled Finish	=
Dimensions	
Side Offset	0' 0 3/32" =
Top Offset	0' 0 1/8" =
Width (default)	3' 0" =
Height (default)	7' 0" =
Thickness (default)	0' 1 3/4" =
Panels Undercut (default)	0' 0 3/4" =
Rough Height	=
Rough Width	=
Analytical Properties	
Analytic Construction	=
Visual Light Transmittance	=
Solar Heat Gain Coefficient	=
Thermal Resistance (R)	=
Heat Transfer Coefficient (U)	=
IFC Parameters	
Operation	=
Identity Data	
Fire Rating	=