Insanely Great Stairs and Railings with Autodesk® Revit®
Phil Read – HNTB Companies

Class Code: AB114-1

Class Description
Creating wonderful, extraordinary stairs and railings in Revit is only limited to your imagination! I'll be presenting approachable solutions to challenging, real world examples of stairs and railings. From "old-school historic" to "tricked-out modern"—all of this is possible in Revit using both in and out-of-the-box techniques that will help you create elegant solutions that are beautiful, easy to implement, and functional. This is not a class just for "super-advanced" Revit users! If you only understand the basic functionality of stairs and railings and are looking for solutions to take your stair and railing designs to the next level in Revit, this class is for you!

Key Learning
• Best practices for stairs and railings
• Compelling in-the-box techniques
• Exceptions to the rules of stairs and railings
• Thinking outside of the box
• Predictable and elegant tips and tricks

About the Speaker:
Phil Read is Vice President and the firm wide Director of Practice Integration for HNTB and is focused on extending the use of BIM for visualization, analysis and integrated project delivery (IPD). Prior to his role at HNTB, Phil was a Key Account Project Manager at Autodesk, involved in the successful training, implementation, and support of the Autodesk® Revit® suite of products. Phil has contributed to numerous books and presentations on BIM, best practices and the cultural impacts of disruptive technology. Phil holds a B.S. in Communications and a B.A in Architecture as well as a M.A. in Architecture.

Contact:
Google Voice: (980) 428-9567
Email: pread@hntb.com
LinkedIn: http://www.linkedin.com/pub/phil-read/3/92b/587
Blog: http://architecture.blogspot.com/
Twitter: http://twitter.com/philread
Sample Files (RVT): http://dl.dropbox.com/u/125962/AB114-1.zip
Introduction
Making Complex Stairs and Railings in any software application can be difficult. First, you have to understand the rules and constraints of the application – in effect you’re learning another *language*. Second, you have to contend with complex design issues. Stairs and railings may be straightforward and functional (think of a steel or concrete egress stair) or they may be extraordinarily complex and sculptural:

*Titanic Grand Staircase*

These two issues of “design idea” meeting “software” (meta: art vs. science) can create a lot of frustration for a design team! Even with an elegant “generic” modeling application (SketchUp, Bonzai, 3DMax, Maya, Rhino, etc.) stair creation requires greater application “fluency” compared with modeling other objects. And even once you understand the rules for creating and manipulating geometry in a generic modeling application – what still remains difficult is the design idea and managing design iteration.

In Revit, stair and railing creation is complex on yet another level: Revit is purpose built for designing building elements and relating them to the rules of constructed relationships. So rather than being just another “generic” modeler, Revit is biased toward relationships specific to designing a building and maintaining those relationships as the design changes. And to top it off, there’s one more layer still. Within the broader language of Revit – there’s a very specific language to creating stairs and railings.
In Revit, it’s important that your building elements understand relationships to other elements. They need to view, schedule and maintain change appropriately.

To this end, the “out of the box” stair and railing functionality can be used in some interesting, innovative ways and we’ll cover some great techniques in a bit. But in many cases, out of the box functionality will **not** suffice for at least one of two reasons.

**First** – the functionality you require may not yet exist. Simple enough. This will require you to use other out of the box functionality to create a reasonable facsimile of what you need. The upside is there may actually be some other desirable benefit to using this functionality in an unintended manner. But the downside is that you’ll have to be careful where the meta-data ends up - particularly with regard to scheduling. It’s very likely that you’ll have to use some filtering in your schedules.

**Second** – the functionality that you require may be so indefinable that you might as well stop waiting for an out of the box solution and start looking for elegant hacks. This is when the system you’re trying to create doesn’t contain a well-defined set of rules, or if the exceptions to the rules of your system are so unpredictable that attempting to define the overall system isn’t worth the effort.
My approach has always been to try and find the solution that is technically correct – the ‘best practice’. But at the same time, I think you need to take Implementability into account. I’ve got two simple approaches to determining when to use an in the box solution – and when to think out of the box.

First: Spreadsheet or Sculpture? If what you’re designing can be described within a spreadsheet then it’s likely that you’ll be able to use the stair tool out of the box (like an egress stair). But if what you’re trying to design cannot easily be defined within the confines of a spreadsheet (i.e., stair tool dialog) then it’s likely you’ll be working outside the box of the default stair tool. This is because there are just too many exceptions and peculiarities to make using the stair alone tool worthwhile.

Second: Just because your design might not fit within the confines of the stair tool – it doesn’t mean that you can’t maintain a proper balance of project part/whole relationships when using other tools like the Family Editor. Remember that the Family Editor will allow you to maintain many relationships in a project by editing single elements and then reloading to maintain design iteration.

Overall, it’s important that whatever your approach – inside or outside of the stair and railing toolbox – that your solution should maintain a balance of “efficient
predictability”. Design efforts are distributed across people and teams and it’s important that your approach is not so unique that anyone else would not be able to understand how to modify your design when it changes (and it will).

Finally – here’s a couple of resources to get you thinking about stairs and railings as elegant, artistic expressions. The first is in print and the other is an online resource.

**Staircase Design** (2002) by Ana G. Canizares (ISBN 3-8238-5572-7) is a really great art book that organizes stairs along Types, Materials and Details (definitely light on the details). Please keep in mind that this book is not necessarily a great technical resource – so it isn’t about resolving the specifics of constructability. Yet this book was formative back in 2002 as I flipped through the well-composed photographs while trying to wrap the language of Revit around some wonderfully challenging designs.

[http://www.stairporn.org](http://www.stairporn.org): Most are functional. Some would seem rather dangerous to pets, kids and while holding a glass in one hand and a bottle in the other. And a few are not even meant to be functional at all! In spite of the title of the website - all are SFW.

**Bottom Line: What Justifies Workarounds / Hacks / Etc?**
Be careful when reaching into your bag of Tips and Tricks! You need to weigh the cost of implementing a new process against the cost of doing what’s familiar. You want to make sure that you’re maintaining a balance between *Predictability* and *Efficiency*:

![Predictability and Efficiency Balance](image)

Just remember these 4 simple rules:
- **Beneficial**: Not just to you – but to the project team
- **Efficient**: Implementation and changes are fast and predictable
- **Elegant**: Understood by the team and any last minutes new members
- **Repetitive**: Can be used on many projects
Component Parts of Railings

- Rail Structure
  - Profiles – You can have many profiles per railing. But each profile family (*.rfa) may only have one loop. And that one loop needs to be closed with no overlapping lines.

- Baluster Placement
  - Start/End/Cornet – Defines the beginning, end and corner conditions of a railing sketch.
  - Internal Patterns – Balusters between the beginning, corners and end balusters.

Baluster Placement Dialog

- Different Types of Balusters
  - Baluster Post – Parameters to extend the top vertically
  - Baluster – Same as above – but top and bottom can extend vertically as well as be cut with a void to match the angle of a sloped Rail structure
  - Baluster Panel – Same as above – but contains parameters to control a panel width.
Railings as Repetitive System
It’s unfortunate that Revit doesn’t yet allow components to be quickly and easily distributed along a user-defined path. There are “line” based families – but these don’t work in curved conditions. In the meantime, you can use the railing tool to distribute elements along paths that are straight, curved or both within the same sketch. This is really helpful when you’re trying to accurately distribute components!

Couple of Rules:
1. Nest the desired component family in Baluster Post family
2. FYI - Don’t expect it to schedule or tag. Don’t “Share” parameters of nested Family in an attempt to schedule. It won’t schedule. It’ll break.
3. Railings don’t have to contain handrail profiles! It’s easy:
   a. Set the Top of the railing to Host
   b. Give the baluster a positive Top Offset value
   c. Delete the railing profile. This hosts the baluster by the “Host” rather than the railing profile. Picking any of the “balusters” will allow you to edit the sketch.

Uses – Any repetitive elements that must be placed on center and evenly distributed along a path:

- Outdoor Lighting
- Outdoor Planting
- Pipe Bollards
- Light Rail Sections
Railing as Repetitive Component
Creating small and highly repetitive railing systems and then Grouping and Copying takes considerably longer to update than modeling a railing-shaped Family Component and copying the family throughout a project.

For example, hotel balconies contain small railing segments. And a modest project of this type can contain hundreds of railing segments (one for each room’s balcony). But the benefits of updating a Family Component compared to a Group is significant:
  - Reload 1000 Families: <5 seconds
  - Update 1000 Groups: >5 minutes

Railing as Walls and Curtain Walls – Advantages
- Arbitrarily Placed / Relocated Balusters
- Arbitrarily Curved Panels
- New grids may be added, unpinned and easily exchanged when adding or modifying panels

Curtain Wall as Railing – Additional Panels Placed and Unlocked
“Seuss Railing” – The Curved Handrails are not part of the sketch. Note the 3D pattern file applied to the panels to simulate wire mesh.

Wall as Railing – The wall is being intersected by a wall hosted “baluster” that “splits” the glazing on placement of the baluster creating individual “panels”

Overall

- Using Curtain Panels as a process for making chain link fences is really great and in many cases is superior to the OOTB railing tool. Be sure to modify the panel with a 3D model pattern file.
- Ability to “unpin” and modify the baluster and panel location and sizes graphically a fantastic advantage of using this process. Otherwise, you have to go back and forth between a spreadsheet-like dialogs with little indication of the results until you finish. And often making it work in one condition – only to discover the rule ‘breaks’ somewhere else.
- Being able to contain space is another significant advantage of using walls and curtain walls as a railing. Otherwise you need to draw (and manually coordinate) Room Separation lines concurrent with railing locations.
Component Parts of Stairs

- Nosing Structure
  - Profile – One loop. Closed.
- Stringers
  - Left / Right / Middle – Only rectilinear shapes are allowed and can be located to the left, right and center(s).
- Treads
  - Depth / Thickness – Important to keep in mind that when the thickness changes, the top of the tread maintains elevation.
- Baluster / Railings
  - Specific Placement Rules for Stairs: Creates specific and unique conditions for the behavior of railings when used with stairs.

![Type Properties](image-url)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Treads</strong></td>
<td></td>
</tr>
<tr>
<td>Minimum Tread Depth</td>
<td>0' 11&quot;</td>
</tr>
<tr>
<td>Tread Thickness</td>
<td>0' 2&quot;</td>
</tr>
<tr>
<td>Nosing Length</td>
<td>0' 1&quot;</td>
</tr>
<tr>
<td>Nosing Profile</td>
<td>Stair Nosing - Pan : Stair Nosing - P</td>
</tr>
<tr>
<td>Apply Nosing Profile</td>
<td>Front Only</td>
</tr>
<tr>
<td><strong>Risers</strong></td>
<td></td>
</tr>
<tr>
<td>Maximum Riser Height</td>
<td>0' 7&quot;</td>
</tr>
<tr>
<td>Begin with Riser</td>
<td>✔</td>
</tr>
<tr>
<td>End with Riser</td>
<td>✔</td>
</tr>
<tr>
<td>Riser Type</td>
<td>Straight</td>
</tr>
<tr>
<td>Riser Thickness</td>
<td>0' 0 1/4&quot;</td>
</tr>
<tr>
<td>Riser to Tread Connection</td>
<td>Extend Riser Behind Tread</td>
</tr>
<tr>
<td><strong>Stringers</strong></td>
<td></td>
</tr>
<tr>
<td>Trim Stringers at Top</td>
<td>Do not trim</td>
</tr>
<tr>
<td>Right Stringer</td>
<td>Closed</td>
</tr>
<tr>
<td>Left Stringer</td>
<td>Closed</td>
</tr>
<tr>
<td>Middle Stringers</td>
<td>0</td>
</tr>
<tr>
<td>Stringer Thickness</td>
<td>0' 2&quot;</td>
</tr>
<tr>
<td>Stringer Height</td>
<td>1' 0&quot;</td>
</tr>
<tr>
<td>Open Stringer Offset</td>
<td>0' 0&quot;</td>
</tr>
<tr>
<td>Stringer Carriage Height</td>
<td>0' 2 1/2&quot;</td>
</tr>
<tr>
<td>Landing Carriage Height</td>
<td>0' 7&quot;</td>
</tr>
</tbody>
</table>
Nosing

The shape of the nosing is not limited to traditional nosing profiles. Any shape that needs to extend beyond the face of the tread is okay. For example, here’s a stair where the treads are welded to the tread above. Note how the nosing incorporates the back of the tread below:
**Note:** Make sure that if you’re using a complex profile to represent both the nosing and the riser, to set the Riser Type to “None”. Otherwise it won’t assign the profile correctly since it’s overlapping with the default riser.

Finally - keep in mind that the complex nosing profiles can be set to wrap all sides of the treads:
Stringers
The default stringers in Revit can only be rectilinear and may be positioned to either side of the stair (left and right) and middle. In order to overcome this limitation – you’re going to use the Railing tool and assign a profile to the railing that acts as a custom stringer.
Keep in mind that stairs may host railings that are not necessarily just to the left or the right of the tread. Here’s a stair with a central “railing” that’s being hosted by the stair:

**Treads**
The default tread in Revit can be pretty boring at times – but it’s very useful to let Revit figure out the tread locations! In situations where you need to be a bit more inventive – you’re going to create a custom *baluster*. But instead of the baluster being *vertical* it’s going to be *horizontal*. This horizontal baluster will be used in conjunction with the default tread (in some cases even *completely enveloping* the tread). In this way the baluster acts as either a support to the tread – or as the tread itself.
First – the baluster can be used to create an interesting tread support for the default tread. A couple of rules:

- **Not every baluster needs a railing** - If your baluster support isn’t going to be a part of the “real” railing, just create another “railing” that is hosted by the stairs.
- **Baluster** – Component Family
- **Nesting** – if you have a really complex support element – it may be helpful to model as a generic element and nest into the baluster family. This is because the hard-wired reference planes and parameters may cause your baluster to fail when the parameters attempt to “flex” in the project. By modeling the geometry elsewhere and nesting it, you avoid this hassle.
- **Level of Detail** - Assigning Course/Medium/Fine and Orientation is key to faster graphic regeneration in Revit. This can be done at the deepest level and is remembered through nesting.

![Original Component](image1)

![Nested in Baluster Family](image2)

![Final Placement](image3)
Second – the baluster can be used to create the **actual tread**. Some more rules:

- **Nesting** – Don’t forget that you can nest materials and length parameters in your generic element and then control the values from the project environment.

- **Level of Detail** - Assigning Course/Medium/Fine and Orientation is **key** to faster graphic regeneration. This can be done at the deepest level and is remembered through nesting.

- **The Actual Tread**
  - Can either be a “surface” treatment to the baluster as tread sit on top of the support element
  - Inlaid into a carefully constructed baluster tread
  - Entirely enveloped by the baluster tread
Finally – if a custom center stringer and support element is supporting the stair and handrail then it may even be possible to nest both the left and right balusters together. This can simplify and shorten modeling time.
Start and End Posts are really useful and can help complete the structure of your custom railing and baluster system. This example uses only start and end posts to anchor the custom railings and stair structure.
This example uses only the end post to anchor the entire stair structure.
This example uses the Start Post to create an entire railing system that would not (no matter how hard you try) be possible with the out-of-the-box rules. And the nice this about using the Start Post is that whenever you create the stair – the “railings” are automatically added.

First – model what you can with the out of the box tools. Even if you can only model the treads, this will help give you context to the rest of the system. Then export this and import into a Generic Model family.
This is a great technique for creating really complex and difficult to predict systems - like a woven cable net around a stair. The important part of this technique is to use the out of the box stair tool to model whatever is possible as context. Then:

- Export the stair in 3D
- Import the exported file into a Generic Model Family
- Model the cable network in context (Reference Planes are helpful)
- Nest the finished family into a Baluster Post family.
- Load the Baluster Post into the project and associate it with the Start Post of the railing. Whenever you create a stair of the same configuration the cable net will follow. Even for a multi-story stair!
Stair As Object / Sculpture / Feature – Apple Stair

Stair in 5th Avenue NY Apple Store (the center cylinder is the elevator core)

Actually, there’s very little about the “Apple” stairs that can’t be modeled out of the box! Even though the stairs are very sculptural there are quite a few repetitive relationships that are easily identified and defined. Of course, what doesn’t strictly conform to the rules of the stair tool will have to be modeled elsewhere. So what can be modeled out of the box?

- Treads
  - 2” thick
  - 6’ wide
  - 4’ interior radius

- Tread Support
  - A small support “pin” is used to support the treads. One pin per tread supports the inner portion of the tread.

---

1 A bit of a disclaimer: I’m basing the measurements for modeling these stairs on the many photographs that have been posted online of the 5th Avenue Apple Store (as well as other Apple stores). But the process described should in principle be useful for modeling the stairs if more exact measurements were available.
Two pins per tread support the outer portion. With this in mind, we’re able to place the pin as a ‘baluster’ after modeling as a generic model and nesting into a baluster family (not a baluster post), which will allow it to conform to the elevation of the landing. The pin is also given a “fine” level of detail assignment so it doesn’t show up unless the view is set to fine.

- **Handrail**
  - Two handrails types are needed, since they will host different baluster definitions (one or two balusters support pins per tread).

- **Start and End posts**
  - Two families are needed – one for the start configuration and another for the end. They include the horizontal extension beyond the end of the railing sketch.

*Out of the Box Context: Treads, Supports, Railings and Posts*

This portion of the stair that can be modeled out-of-the-box is then exported as 3D context for the remainder of the stair, which will be modeled as a Generic Family and then nested into a start post. There’s actually a third railing in this
stair (which contains no railing profile). This 'invisible' railing is hosted by the stair and creates the remaining elements:

- **Glass Panels**
  - Two swept blends form the upper and lower curved panels
  - A sweep forms the panels at the landing
  - A single extrusion forms the center cylinder
  - A single void extrusion is used to create all the discreet panels separations at once

- **Panel Supports**
  - The railing will be used to host the pins that support the treads, so the actual balusters supports for the railing will be nested into this family.

- **Handrail Supports**
  - Since the handrails will be used to host the baluster ‘pins’ (one or two per tread) the actual handrail supports will be nested into this family. Each panel has a single baluster element.

*Completed Support Start Post with Elaborate Panels and Connections*
Completed Stair and “Start Post”
Overall Best Practices

- Nest Families (Hardware)
  - Create elements once and nesting will save time during iteration
  - Nesting a family as a single component keeps it from ‘breaking’ or ‘flying apart’ when the parameters change in the hosting family.
- Study the Online Samples at http://seek.autodesk.com/
- Download sample files for this document (link on cover page).
- Build / Test in Separate Project
  - Keep This Project of Custom Stairs / Railings
  - Copy / Paste or Transfer Project Standards between your working and actual project.
- Level of Detail / Visibility is Important (Coarse / Med / Fine)
  - Don’t forget to assign appropriate detail levels and orientation to smaller elements in order to automatically control visibility.
- Careful to Filter Schedules
  - Using a wall or curtain wall as a railing may have certain advantages, but be sure to filter out the name of the Type so it doesn’t improperly schedule.

Conclusion

Hope this session gave you some ideas on innovative uses of the stair tool in Revit!

Please don't hesitate to send email and descriptions of how these techniques have helped you overcome complex design and modeling situations. And if you send and image or two please let me know if it’s okay for me to post your efforts in the Arch | Tech blog online at http://architechure.blogspot.com/.

Enjoy every sandwich!

Phil