



## Revisiting Project Templates: Maintaining Consistency and Company Standards

Robert Manna – Stantec Consulting Ltd.

David Spehar - Stantec Consulting Ltd.

**AB2064** In the last 2 years, Autodesk® Revit®-based software has added a number of features that make it even more important to maintain a solid company template that can help to ensure high-quality, consistent documents and data. Yet at the same time projects are constantly evolving along with your standards, or your firm is so large that a single template can't fit all that you need. This class examines some of the key feature changes in Revit, their impact on how your project teams work, workflows for how to make sure project teams can get the latest standards, and a review of some simple API utilities that can make it even easier for users to fully use the benefits of the project template you've worked so hard to build.

### Learning Objectives

At the end of this class, you will be able to:

- Describe the latest features in Revit that impact project template design
- Develop a process for updating standards in active projects
- Determine how best to break up your standards to make them more modular and adaptable
- Consider custom tool development to help users better employ company standards

### About the Speaker

*Robert works for Stantec in the Boston office, he has been a key team member on multiple projects, and he now serves as the BIM R&D leader for the firm as well as providing business consulting services for clients implementing BIM. He has taught internally, provides high level support as well as planning and implementation of new tools. In 2013 he led the organization of the first annual RTC Design Technology Summit, he has spoken at: RTC NA, Autodesk University, has been a guest lecturer at the BAC, and has presented at BIM events hosted by the AIA, ACEC, Autodesk and resellers. He has written two articles about Revit for the AUGI AEC Edge magazine, and has written assessment questions for KnowledgeSmart. He maintains a personal blog dedicated to Revit & BIM. When he has time he hangs out with his wife and three year old daughter, and enjoys skiing, swimming and biking*

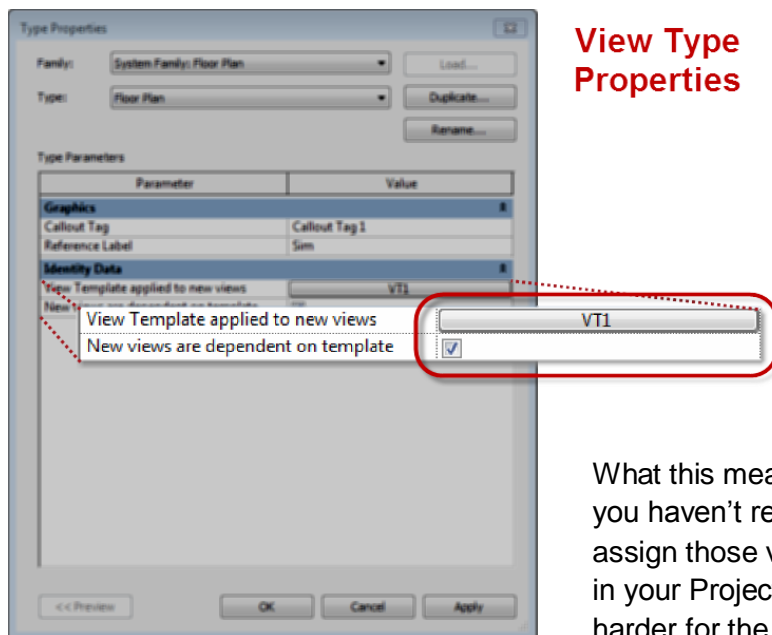
[robert.manna@stantec.com](mailto:robert.manna@stantec.com)

[david.spehar@stantec.com](mailto:david.spehar@stantec.com)

## New Features

### View Templates

When we last talked about Revit Project templates we were big fans of View Templates. The problem was while you could put them in the project there was no guarantee that your users would use them, and second you had to manually update changes in the templates to the views themselves. Fortunately, since then Autodesk came through and made some improvements which we'll go through.



First, view templates can not only be assigned to Views, but Views can also be set to be dependent on the assigned View Template. The view template acts as a live reference, options controlled by the View Template cannot be modified in the view and changes to the View Template will be reflected in any views with the assigned view template.

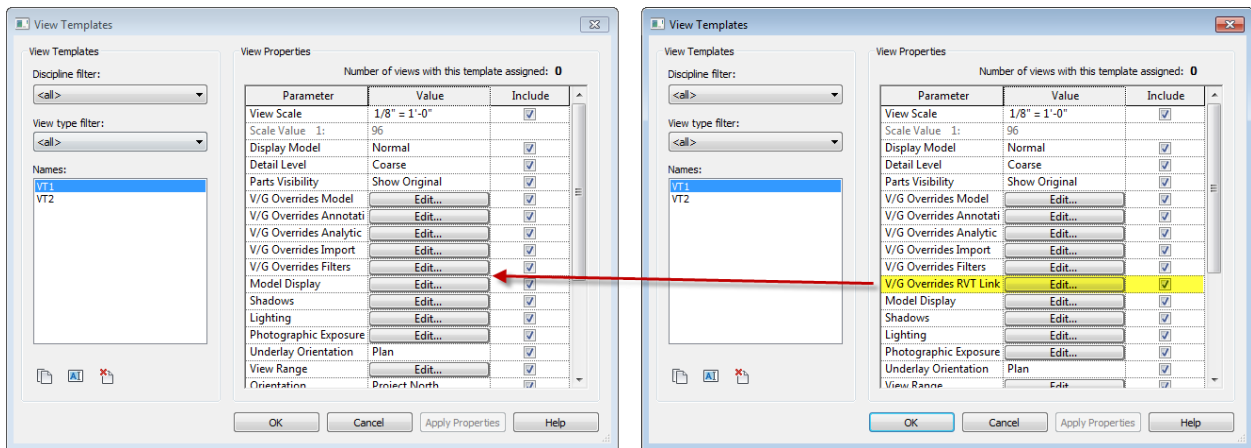
What this means to your project template. First if you haven't realized this already, you can now assign those view templates to the proper view(s) in your Project Template. Now we're making it harder for the day to day user to screw up your drawing set graphics and you're making it easier for your Model Manager\Project Architect to make sure the drawing set is consistent.

What have we done? We went View Template crazy! We've got something for everything, even view templates for our generic modeling views. The real beauty is the allowed granularity of the View Templates. So in some cases for instance our Modeling Views the only thing that the View Template controls is the View Classification Parameter which helps with the sorting of the Project Browser. Everything else the user is welcome to (and encouraged) to modify.

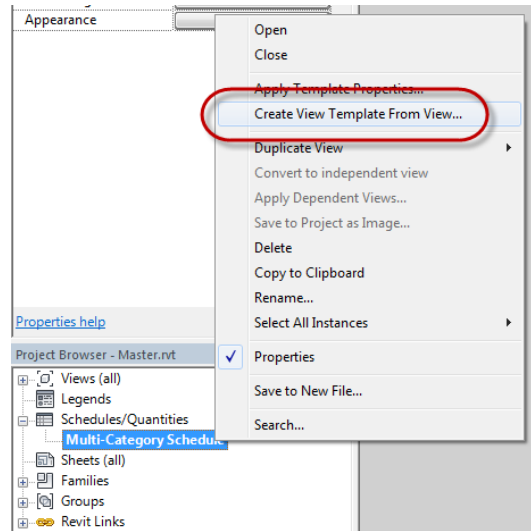
Naming of View templates requires a little upfront planning. Ever since the introduction of the ability to include (or exclude) specific settings, care needed to be taking in naming convention. This has become more important now that the default View Template assignments lock settings. It is helpful to your users if you establish a consistent system such as prefixes to define discipline (A\_floor plan, I\_floor plan, etc.) or to define use (doc\_floor plan, pres\_floor plan, etc.).

One critical piece to keep in mind (that we'll discuss more later) if you're setting up View Filters as part of your View templates you need to be very aware of the changes made and double-check to make sure Filters have updated properly.

Worksharing and Linked Files can also both impact View Templates. When either operation occurs (a file is linked or worksharing is enabled) for all existing View Templates the associated Graphic override tabs are automatically included as part of the View Template. Settings will be the defaults, however if you intend to not control the visibility of links or worksets via your View Templates users will need to make sure to go back and un-check those items. This workflow issue is also a great example of why some firms have chosen to use a Worksharing enabled file with stub links as their starting point for a project template, this is particularly popular with engineering companies.

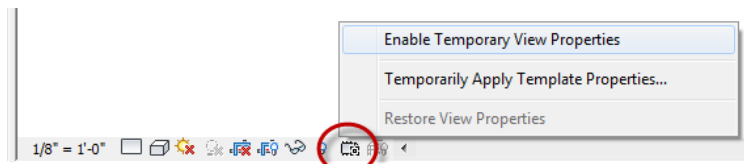


You can use view templates with schedules too! The trick is you need to create a View Template from an existing schedule. Once you've done that you can customize, duplicate and create as many as you want. Options for Schedule View Templates are limited, it only allows you to control the "Appearance" tab of the schedule properties dialog.

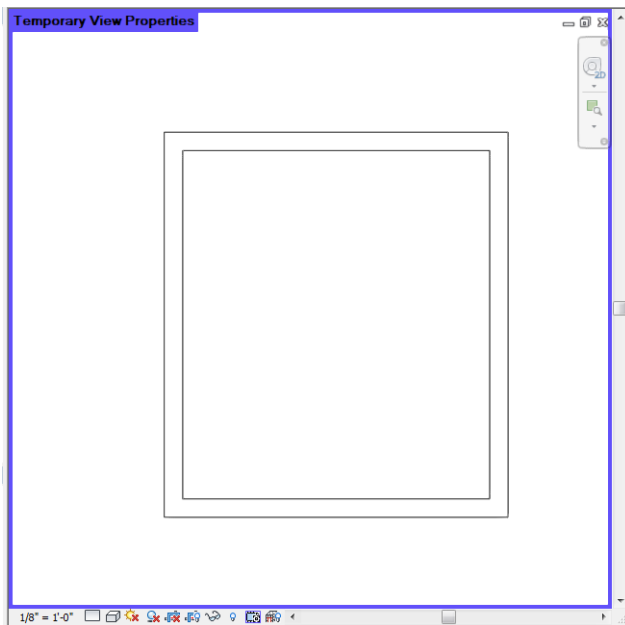


Where this is particularly helpful is with the new Schedule formatting features in 2014, you can now set the Text styles for the Title, Header and Body. Remember, unlike previous schedules this is no longer just the font and font size, but the ability to select any of the existing Text Types defined in the project, thereby exactly aligning schedule appearance to the Text Types you've already created for your Project Template.

**Temporary View Properties**



To compliment View Templates, in the 2014 release Autodesk provided full blown temporary view overrides. Now when a user is in a View controlled by a View Template if they need to temporarily change something that is controlled by the View Template they can enter the Temporary View Properties override mode and change what they need to. This is a temporary view state, so the view will revert back to the settings that are applied by the View Template, however it will retain the temporary state even if you close the view. Further, since it is a temporary view state, you get a nice bright purple border to let the user know they are in a temporary view state. While in the temporary view properties state users can either directly override settings, or they can temporarily apply an alternate view template.



What this means to your project template? You can feel less guilty about locking stuff down! If your users do complain, just remind them that they now have the ability to temporarily override the view template to see what they need, without risk of compromising the proper graphic settings of the view. You may also want to consider configuring some alternate view templates that your users can quickly switch to. The menu button develops a history of recently applies alternate view templates so that users can easily flip back and forth. This feature was originally developed because structural engineers wanted quick ways to visually toggle between seeing the geometric model versus the analytical. MEP users could use it to toggle

quickly between line diagrams and colored model views or architects could use it to quickly turn on visual fire ratings.

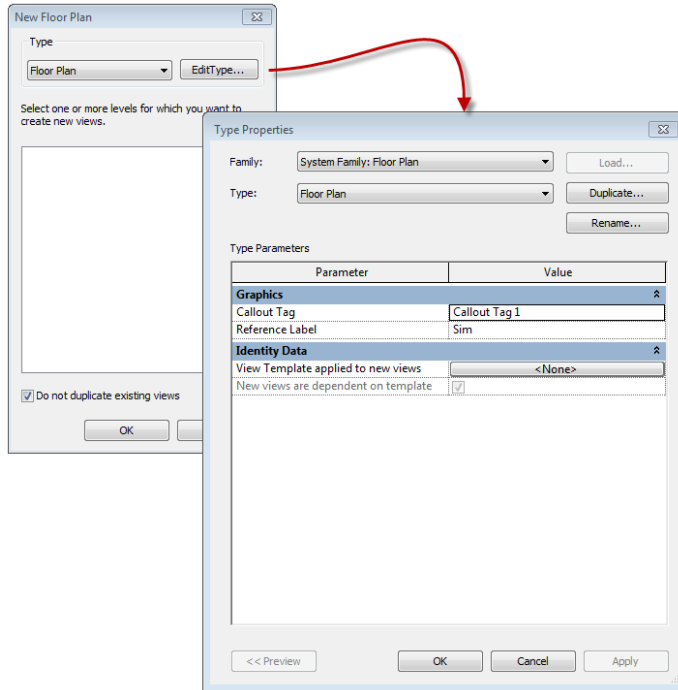
What have we done? We were already committed to the View Template approach, now we're even more committed and feel justified in our decision.

### View Types

In the 2013 release Autodesk finally (it only took over a decade) gave us the ability to create View Types for not just sections, elevations and details, they also added Plan View Types (yay!). From the create Plan View dialog, you can duplicate the default type and create new types. More importantly a View Type can also have a default (linked) View Template assigned. Therefore when you create a new view of a specific type, you'll get all the proper settings.

Why are so excited about this? Well, we had all the View Templates and now we can create unique View Types to match the majority of the View Templates. Now we're making it easy for users to follow the standards (we're not sure we can make it any

easier). With view types that match our templates users need only select the proper view type when creating a new view.



There are a few “gotchas” though. The View Template parameter behaves like an instance parameter, so if the user creates a new View with the wrong View Type, just changing the View Type will do no good, you need to also update the View Template to be the correct one.

Second, View Types do not transfer via Transfer Project Standards (sad trombone). We’ll talk more about this when we talk about container files but suffice it to say, if you want View Types to automatically set View Templates, you’ll be able to transfer the View Templates, but

you’ll need to create the View Types and assign the templates to them after you transfer the templates.

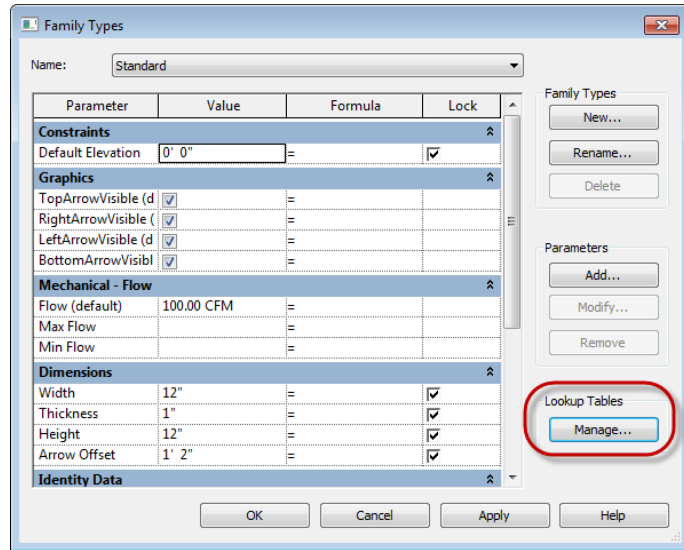
With so many View Types available, it also becomes important to choose carefully what your default View Type will be when a user does create a new view. This applies both to creating new Plan Views and the “default” view types generated when creating a new Level. In both cases be sure to make selections based on what you desire to be the defaults for your users and then delete them. This will set these selections as the defaults in the template.

### Alternate Dimension Units

2014 finally gives us the ability to have dimensions report their values in two sets of units for example Meters and Feet/Inches. Why mention this here? The functionality has been implemented as a Type Property of the Dimension (Style) Type. Therefore if you’re a firm that often does work where you are required to have dimensions reporting multiple units, you’re likely going to want to add the necessary dual unit Dimension types to your template so that teams don’t have to create them each time. Not to mention you don’t always need dimensions to report two values, so setting up the different types ahead of time helps prevent confusion and drawing error later.

### Look-up Tables

Look-up tables, in a course about Project Templates, what..? When Look-up tables were little text files that sat in a lonely folder on your server we would've talked about them if we were doing a course on Revit deployment strategies. With 2014 though, Look-up tables have moved "inside" the associated family (thank goodness!). We wanted to mention that here, because if you're maintaining a project template for MEP, 2014 is probably a good time to revisit your template.



If you haven't already upgraded your template, when you do, it's going to want to know where the Look-up Tables for any loaded content already exists. If it doesn't find the tables you'll get a warning message that you'll have to pick through each unique family and upgrade and re-point the table for, so that it can be imported. Our recommendation set-up 2014 to look back at your 2013 look-up table directory and with luck that will solve most of your problems.

### Other things to update in your template (if you haven't already)

If it has been awhile since you took a serious look at your template here are a few other items that deserve consideration thank to the changes in the last two years:

- Stair types – the increased flexibility of the stair tool means you should consider new stair types and possibly more varieties.
- Railing types – the addition of the Continuous Handrail and Toprail options means it is completely worth revisiting your pre-define rail types as well as making sure you have the proper handrail bracket types loaded for your users to use.
- Specific Angles for Pipe, Duct and Conduit routing – for your system types you can define any angle, increments or specific angles, helpful to assure your modelers are meeting engineering standards/requirements.
- Multi-rebar annotations – this is a new tagging tool that allows you to tag multiple rebar instances with a single tag. You must create tag families to be loaded and assigned to types created in the project. Tags support several new label options to report quantity of rebar and lengths.
- Rebar Type Hook Lengths – you can now define the behavior of rebar hooks for your rebar types.

## Maintaining Consistency

### Techniques for Maintaining Consistency (lessons learned):

We've spent quite a bit of time in the last two years learning all about maintaining consistency between Revit files, particularly active project files when "standards" or "settings" change and the projects need to be updated. While you may not always be concerned about updating active projects, if you're dealing with multiple templates or source files, the same challenges may still apply.

### Transferring View Types, View Templates & View Filters:

View types cannot be transferred through Transfer Project Standards, so this means that you need to manually create any View Types. It also means that you have to manually assign the correct View Template to the proper View Type once the View Template is imported into the project/template. View types is also a good example of where a little bit of API development could help. A small API macro or tool could easily be written that will create or recreate view types in one file based upon another document.

View Templates in combination with View Filters have some unique issues to be aware of in conjunction with Transfer Project Standards (TPS). We've learned, through a bit of trial and error that:

- An initial transfer will work fine. If you are using TPS to create View Templates and View Filters in a file that previously did not have those Templates and Filters the transfer will work as expected.
- In the case where View Templates already exist, but they need to be updated so that you have identical settings in two different files you must be aware of the following.
  - Two files "Master" and "Link".
  - Both files contain a View Template named "VT1"
  - VT1 is modified in the Master, specifically View Filters are removed or added.
  - When TPS is used to transfer VT1 from the Master to the Link, VT1 **will not** update if the **only** change is to View Filters.
  - In order for VT1 to update properly, in addition to the changes to View Filters settings for either **Model Overrides** or **Annotation Overrides must have been updated.**<sup>1</sup>

From what we can tell, changes to in the View Filters tab of a View Template simply do not qualify as a change when Transfer Project Standards is used to bring in updated versions of a View Template. When there are other changes Revit prompts you to overwrite the View Template, when it is only the Filters, no warning dialog ever prompts you to overwrite the existing template and its settings.

---

<sup>1</sup> We've tested Model and Annotations, other tabs may also qualify.

View Filters can do odd things if you are working with them in a workshared project. The issue that have isolated is that apparently due to how the workset for a View Filter is handled;

- User A modifies View Filter (VT1).
- At the same time user B uses Transfer Project Standards to update all of the View Filter.
- When the users Sync With Central two versions of the View Template will be generated and the name of one of them will be followed by an incremental number (VT1 and VT1(1) ).
- This can also happen if two users simply modify the same View Filter in their own local files.

We suspect that the View Filter is being treated similar to when two users have created two new types with the same name and one is automatically renamed during Sync to Central, the behavior suggests that a user cannot actually “check-out” a View Filter the same way other “objects” are checked out to a user when they make changes or modifications. Our best recommendation to avoid this issue is that in particular when you want to transfer view filters from one project to another, that you assure all users have SWC’ed and no one is actively working in the models except the user performing the transfer.

### **Legends as Detail Groups**

Another item we’ve struggled with quite a bit is legends on sheets, particularly sheet notes (sometimes called Keynotes, we like to refer to them as Keyed Notes), but also legends that include graphics. MEP disciplines often like to use these types of “legends” on their sheets, but in our case we working with an architectural and interior design group that relies heavily on these types of legends in their drawing sets which are very consistent.

As we all know “Legend” views cannot be transferred between projects. The approach that we finally settled on was to use Detail Groups to maintain consistent legend content among projects. For the group we were working with this was particularly important because even though the legends are in their project template, the legends could be updated at any time, and since these are “standard” sheet notes they would like their active projects to also reflect those changes therefore it needed to be as easy as possible to propagate those updates. We took all content 2D content, except for legend components and grouped those items. Legend components still need to be updated in each unique project, but this is substantially more manageable than having to update everything. Another advantage to Detail Groups is that they don’t have to be placed on Legend Views, they can also simply be placed on drafting views, or even sheets if you wanted.

If you do want to use Detail Groups to manage Legend content here are some key points we learned about the process:

1. Create the detail groups in a blank file based on your project template.
  - a. Create a blank view and place all of your annotation elements. If you need to integrate with Legend Components, lay everything out in a legend, and then copy/paste everything (except the Legend Components) to your blank drafting view.



- b. With all the elements in the blank view, group the elements and name the group.
- c. Save the group out from the file.

### Stantec philosophy (modularizing standards)

Because we are such a large organization spanning multiple disciplines, market sectors and geographic regions (not to mention that we frequently acquire firms) we needed to take a long, hard look at our approach to template development. Right or wrong (the jury is still out) we settled on a multi-tiered approach. Part of which was making a clear distinction between what we defined as “Standards” and “Best Practices” and then deciding the right places for these to live.

For our Standards definition, think (dare I say) CAD standards. These are the things that apply to everyone and every project and are mostly geared towards graphical standards such as text styles, dimension types, object visibility settings, etc. We also defined some Model Management standards such as parameter types/names for defining view and sheets and browser organization. For us, “Standards” do not include things such as sheet naming/numbering convention or schedule formats for specific object categories.

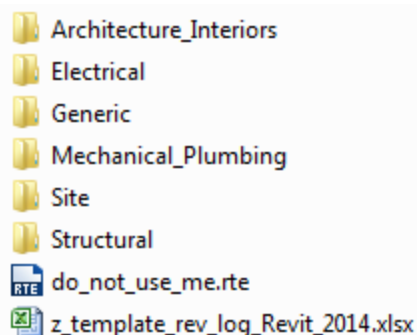
For our Best Practices definition, this is a little trickier. These are things that make a lot of sense for a region, discipline or market sector but are not the same for everyone. Perfect examples are the sheet naming/numbering convention and schedule formats listed above. Many people like to predefine the sheet format in a template. We found this to cause more confusion and angst than defining any type of standard. What works for a team in British Columbia may not work for a team in South Florida. Same goes for schedules such as door schedules or mechanical equipment schedules. Just because I like to document my doors one way does not make it a standard for someone else. As much as you like to think that your way is a standard, you have to ask yourself if it is the only way.

As you can imagine, this approach was the cause for much debate amongst our template development team but after some gnashing of teeth and a lot of name calling we finally came to a consensus. The end result was a multi-step process.

#### *STEP 1 – Build a Generic Template*

We developed a native Revit 2014 baseline, or “generic”, template that is intended to be the starting point for all subsequent templates. This is where we established all of our Standards. It is not intended to be the starting point for projects but rather the building block for more detailed template development. It was a bit driven by one-box by keeping the standards that pertain to all disciplines (such as view types for Architecture, Structure, Mechanical, etc.) but not defining discipline specifics such as system families.

#### *STEP 2 – Build discipline specific templates*



Once we established our Generic Template, it was time to move on to discipline specific templates. These became a bit more difficult because they start to blur the lines between our definitions of Standards and Best Practices. Because of this, we took an approach that included gathering all known templates from around the firm in an attempt to winnow down to as few per discipline as possible. In doing so we were able to get broader buy-in to our templates; identify some solid Best Practices; and eliminate “variations on a theme” by finding that there were regional approaches to modifying old templates that people were happy to consolidate. The end results were not (nor do we have delusions that there will ever be) just one template per discipline. Rather we consolidated as much as possible to gather the best of the best.

### *STEP 3 – Build Container Files*

This is where we really focus our energy on Best Practices. You have a really cool way to do something such as standard partition types or life safety plans? Great, consolidate everything you need into a Revit file (typically an .rvt and not an .rte) that is easily transported into other project files. Some judgment calls need to be made regarding Container Files containing content, legend views or other elements that are not easily transferable or are managed elsewhere. We just recommend you pick the method that will make managing/updating the information the easiest.

### **General Considerations**

As we ventured down this road, we encountered many “gotcha’s” that you may want to consider.

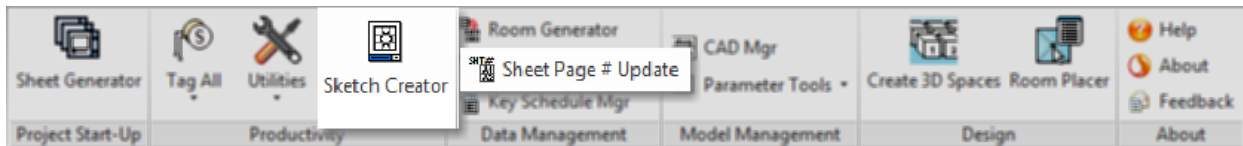
- Building a template that contains very little is a lot more work than you think. Plan on spending a good amount of time planning, talking and testing, especially when it comes to settings such as object styles and line styles / weights.
- Don’t even try to “finish” templates. Give that dream up now. Just create realistic milestones to get production ready versions out the door. Remember – something is better than nothing so the sooner you can get users something, the better. Once a team starts a project with some random template, the harder it is to implement the good one. So if you wait “just one more month” think of how many projects you’ll lose touch with.
- Maintain a good Template Log that identifies all template names, Revit versions they started from, general descriptions and tracking of modifications from production version to production version. This way your users can see the enhancements that have been made.
- Plan on maintaining independent Metric and Imperial templates that are both built from the ground up using the correct units templates. Settings don’t just convert such as line weights, etc. If you have to maintain both, try to create shared standards for things such as text and dimensions. For example, a single text style named for both units like “2.38mm(3/32in)\_Tr”.
- Centralized Content versus Template Based Content. Some decisions need to be made regarding the best place for content. For instance, architecturally you can get away with very little content (even system families) in your template and rely heavily on a centralized content

library and Container Files to load or Transfer Project Standards. However, engineers tend to rely much heavier on preloaded content. Just remember, the more content that lives in a template, the more overhead you'll have maintaining it in multiple locations.

- If you're going to manage a large amount of templates spanning multiple disciplines, consider creating a "default" template to point your users to in the Options – File Locations and place the real templates in a subfolder structure under the default. We named our default template "Do Not Use Me" and for the File Open shortcut we named it "Do not use – click browse". This way, users click browse and they're directed to the folder structure housing all of our templates. They can pick and choose from there.
- As mentioned above, gather the valid templates from across your organization and work with the originators to update them using your new generic template. But make sure the templates are still accessible. The last thing you want to do is take templates offline that users have been relying on because they don't meet your new standards. Talk about not getting user buy-in.

### Using Custom Tools to Encourage the Adoption of Standards

While we can't make anyone follow our best practices and standards, or even use the templates or content that we provide we can take steps to make it more attractive. Our collective experience is that the "standards" or "best practices" that are adopted and followed are the ones that are easier to use than what the tool does on its own. Therefore one place in particular where we've started to try and encourage this behavior is in the development of our own custom API tools. Several of the tools we've developed to date effectively require that our users be using our Project Template(s), or at the very least the shared parameters we embed into the templates, or they need to use specific content, for instance the official Stantec titleblocks and the parameters we've embedded into them.



For example we have created a tool to help automate the creation of drawing sketches for construction administration. This tool depends heavily on a number of Project Parameters to properly generate the sketch sheet and view. A second tool we are working on that will encourage the use of our "standard" titleblocks is a Drawing Package Manager, that tracks when drawings were issued. A third, very simple tool is on that applies page numbers to the sheets in a drawing set (or updates the page numbers). These are all relatively simple tools, but they automate either complex tasks, tasks prone to error, or tasks that are otherwise manually intensive. The key is, these are things that users appreciate being automated, yet to take advantage of the automation, they have to employ our best practices and standards.

2014 introduced some additional API functionality that makes it easier for developers to write tools that make it easier to help maintain standards. Critically developers can now copy/paste elements between two open documents (models). Among the simple tools that you might

consider developing to help maintain consistency and standards would be a “Copy/Paste” for View Templates. Currently Transfer Project Standards is the only effective way to move View Templates between files, however the problem is that you get all View Templates, what if you only wanted one or two? Boost Your BIM has already done a proof of concept by way of a simple [Macro for Revit](#).