

## Presentation Helper Overview

This Python script for Rhino provides a number of tools in order to make marketing presentation drawings production easier.

The design logistic is that the script takes information provided by Revit models and decide how it should be organized in a way for presentation drawing production.

While human judgment and intervention are still needed, the script is able to take over some repetitive work and speed up the process.

For it to work,

1. The drawing has to be exported to a DWG with ZGF Standard Layers from a Revit model
2. The 2d drawing in DWG format has to be brought into Rhino where the script can run
3. After edited in Rhino, the drawing has to be exported to Illustrator to meet final PDF output requirements

See the following pages for a step-by-step instruction from Revit to Illustrator.

The script is under development and has not been tested for a plenty of drawings. There is zero user interface at the moment and the instruction might be confusing sometimes for those who do not know how the script is set up.

For questions, bug report, suggestions, either on the script's functions or on the instruction, send emails to me ([vera.tian@zgf.com](mailto:vera.tian@zgf.com)). I am very happy to help and update the script. We can make presentation drawing production in the future easier together than it is.

### Current Version (20190424) Updates

- Added a function to find out the keyword(s) in a block's name with a given block instance.

### Version 20190109 Functions

- Select block instances by keywords in their definitions' names.
- Replace multiple block definitions of the same family generated by exporting DWG from Revit with a single one (with limitations).
- Explode selected block instances to groups.
- Remove unused block definitions and layers without objects.
- Provide a set of standard layers and move lines, hatches and text to these layers.
- Extend lines to form an enclosed outline for CurveBoolean (with limitations).
- Calculate scale and add a frame for sizing the artboard in Illustrator.
- Add north arrow, scale and legends to a drawing.
- Set up standard marketing line weights for plans at the scale of 1/64" = 1' - 0".

## Editing In Revit

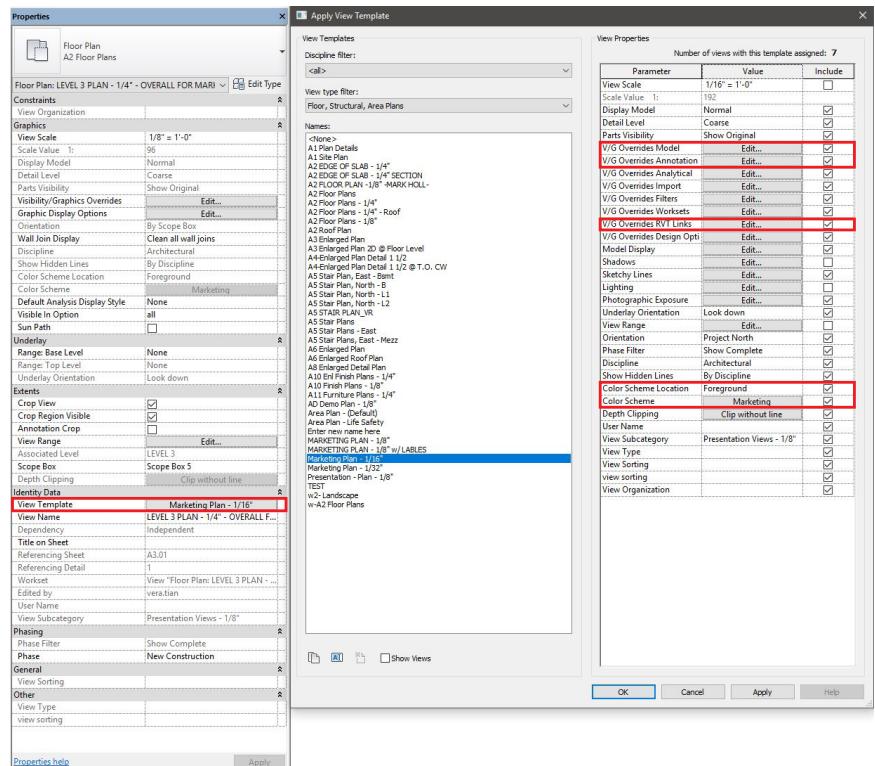
If the presentation drawing is produced after a project has been completed, there should be presentation views already set up in the Revit model.

At such stage, there will not be further changes in the model and therefore it is recommended to save a copy of the model or detach it from the central, so as to avoid any unintentional changes made.

Then, find the already created presentation drawings in the model and make sure of several things through editing View Template:

1. All major furniture and equipment show up and are transparent (i.e. not in white or in any patterns.) This can be set in V/G Overrides Model , or V/G Overrides RVT Links for linked-in models, in the current View Template.
2. All the rooms are filled with marketing colors (Can be set with Color Scheme in current View Template and Marketing Color under Room Properties -- General when a room is selected)
3. All line hatches are removed. They will become lines and hard to clean up when brought into Rhino/Illustrator.
4. All room tags, including names but not room numbers, are visible (Can be set in V/G Overrides Annotation in the current View Template and the family of the room tags)
5. Room colors are not "flooded" to the center of glazing (Can be avoided by adding Room Separations at the edge of glazing)

All of these can be done later in Rhino too but they are easier to do in Revit in comparison.



## Exporting DWG from Revit

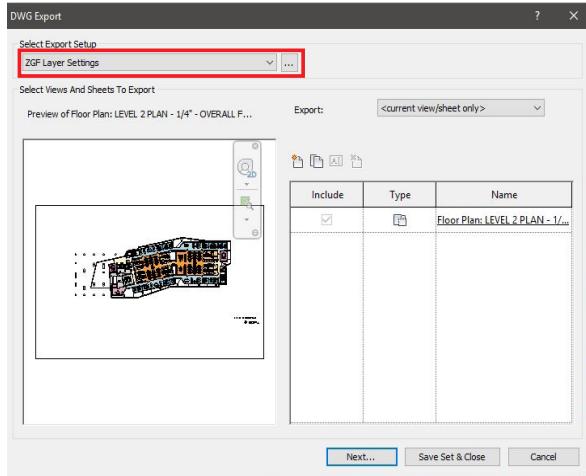
A ready-to-export Revit plan might look like.

Example: ASU 2nd Floor

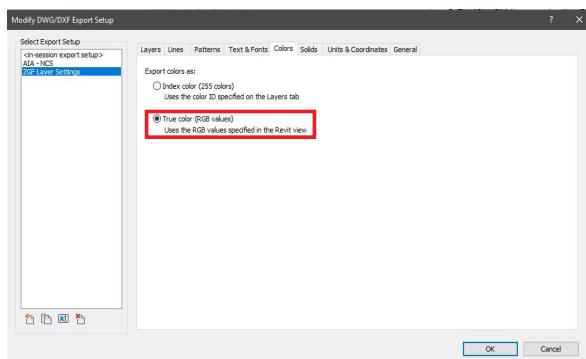


When the drawing is ready to export, click Main Menu -- Export -- CAD Formats -- Dwg.

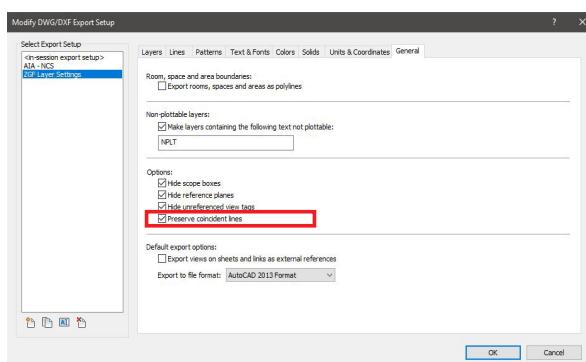
Use ZGF Layer Settings but click on ... to change two settings.



Under Colors tab, use True color (RGB values) instead of Index color (255 colors).



Under General tab, check Preserve coincident lines.

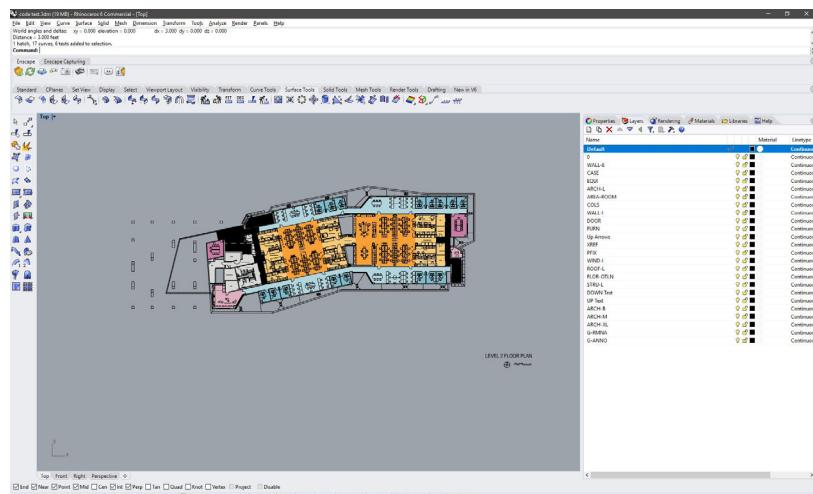


Save the exported dwg file

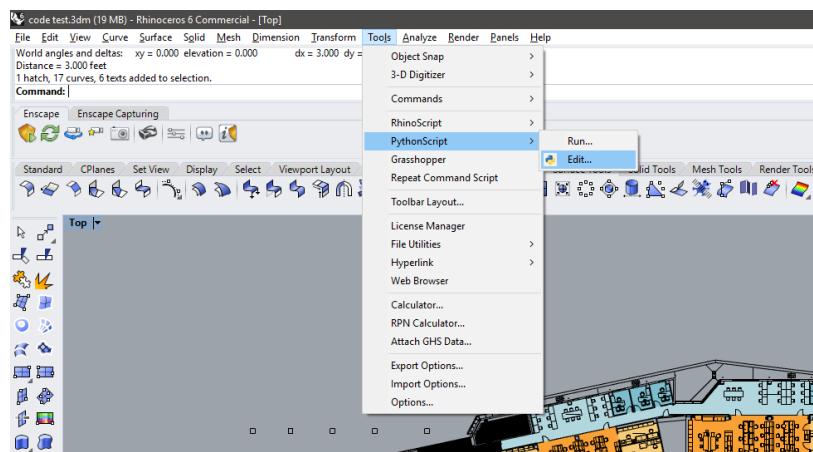
## Editing in Rhino

Open a new Rhino document, set the units at Feet and import the DWG drawing. The drawing should be at the real scale.

While the scripts runs in both Rhino 6 and Rhino 5, Rhino 6 is recommended as it supports editing polylines and hatches better than Rhino 5.

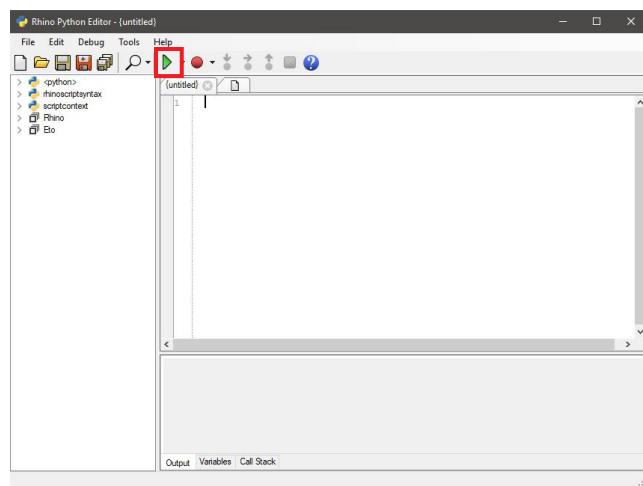


## Click Tools -- PythonScript -- Edit...



Instead of loading the script from a provided location, save a local copy of the script and load the copy. This is because some constants in the script might need to be edited during the drawing production. It also keeps an original script if anything does not behave correctly during the editing.

Click Open to load PresentationHelper.py from where the script is saved.



## Presentation Helper Structure

What the script does is mainly controlled by editing the number of WHAT\_TO\_DO and DRAWING\_NAME indicates the layers where the script's editing will take place. Check the two values and keep them up to date before running the script.

What the script does is mainly controlled by editing the number of WHAT\_TO\_DO and DRAWING\_NAME indicates the layers where the script's editing will take place. Check the two values and keep them up to date before running the script.

Text in green is a brief description of what the script does when values are edited.

These functions will be further explained in this instruction later.

Text not in green are constants that might need to be edited, depending on which project's drawings are produced.

Text in green is a brief description of what the script does when values are edited.

What these constants are and when they should be edited will be further explained in this instruction later.

```
PresentationHelper20190109.py
1  """
2  This module includes tools for marketing presentation drawings.
3  """
4  import rhinoscriptsyntax as rs
5
6  DRAWING_NAME = 'LEVEL 2 PLAN'
7  WHAT_TO_DO = 2
8
9  """
10 WHAT_TO_DO --
11 Edit the number to specify which definition to run
12 [1]Filter Block Instances.
13 ... Select visible and unlocked block instances with block names containing keywords in BLOCKS_KEYWORD
14
15 [2]Replace Same Blocks
16 ... Generate a group of the replacement the instances of similar block definitions with one of them.
17 ... Block instances to be replaced will not be deleted and grouped for review.
18 ... Not used block definitions will be purged before
19 ... Does not support replacement of mirrored blocks at this point. That is, block definitions that are similar will be replace
20 ... with at most two definitions, one is the mirrored version of the other.
21 ...
22 ...
23 [3]Blocks to Groups.
24 ... Convert selected block instances to groups.
25 ... If selection is empty, all visible and unlocked block instances to groups
26
27 [4]Organize Layers.
28 ... Delete unused blocks.
29 ... Organize objects and layers in the document according to DRAWING_NAME, REVIT_LAYERS, HATCH_COLORS and MARKETING_LINE_LAYER
30 ... Delete empty layers after the operations.
31
32 [5]Extend curves to Closest Others.
33 ... Extend every curve in the each group to a closest curve in the selection at each end.
34 ... New curves will stay in the same group.
35 ... Non-curves will not be transformed.
36 ... Curves(polylines, nurb curves) that are not line segments will be transformed to line segments.
37 ... Note: The definition may fail on some occasions to make a collectively enclosed outline for the purpose of CurveBoolean
38 ... Not recommended to be operated on complicated or large numbers of geometries as it might take very long to run
39
40 [6]Delete Empty Layers.
41 ... Unused block definitions will be purged.
42 ... Recommend using occasionally to clean up the layers.
43 ... Empty standard layers will be deleted too but they can be restored by running [7] and [8]
44
45 [7]Add Standard Line Work Layers
```

```
PresentationHelper20190109.py
70 BLOCKS_KEYWORD = set(['Valve', 'Card reader', 'Eyewash', 'drain', 'Pressure Regulator',
71 'Pure Water Fixture', '6_5x2_5_Veritical', '6_Sx2_5_Jamb', 'Faucet', '6065_121_0021', 'Fire Extinguisher',
72 'GRAB-BAR-TOILET', 'Welded_Grating', 'GBF2150', 'B...', 'Faucet', 'Double_Glazed',
73 'White_Base', 'Corner_Guard', 'Mobile_Cabinet', 'Base_Drwr', 'Control_Panel',
74 '3_tier_shelf', 'CDS', 'Cylinder_Restrain', 'Shelf_Stainless', 'Base_Drawer', 'Wall_Shelving', 'EN849SL',
75 'Metal_Panel_Assembly', 'Knee_Space', 'PROF', 'Guard', 'EAF-200', 'SINK-COUNTER-SOLID-SURFACE - SK-01', 'VAULT DOOR COLUMN',
76 'Un-Fixed_Base', 'Benchtop_Support_Bracket', 'Fixed_Cabinet', 'MP0600-0501', 'MP750-0502', 'SC0750-0901F', 'HGC2816', 'Grommet', 'W
77 'CABLE TRAY', 'Exhaust', 'DISPENSER', 'Disposable', 'Grab_Bar', 'CLOTHES_HOOP', 'P-TRIX_POWERS', 'Waste_Receptacle', 'Basic_Wall',
78 'Jamb', 'Top_Rail', 'Tagable_Marker', 'floor_box', 'mirror', 'NOT_PLATE', 'Water_Polisher', 'rying_Rack', '108B_LL005INK',
79 'drying_rack', 'Mobile_Cabinet', 'Emergency_Shower', 'GBF1735DP', 'varsityrack', 'Int_Bi-fold', 'Int_Vertical_Capped-W', 'Overfl
80 'FDC_Connection', 'BOLLARD', 'Knox_box', 'sill', 'Vacuum', 'Pressurization', 'Ice_Maker', 'AED_Cabinet'])
81 """
82 Block name keywords to filter out block instances when find_blocks() runs.
83 Not case sensitive. Modify according to needs.
84
85 Format: BLOCKS_KEYWORD = set(['KEYWORD_1, KEYWORD_2'])
86
87 Sample:
88 BLOCKS_KEYWORD = set(['Valve', 'Card reader', 'Eyewash', 'drain', 'Pressure Regulator', 'Pure Water Fixture', '6_5x2_5_Veritical',
89 'Fire_Extinguisher', 'GRAB-BAR-TOILET', 'Welded_Grating', 'GBF2150', 'B...', 'Faucet', 'Double_Glazed', 'White_Board_with_Tray_148
90 'Corner_Guard', 'Mobile_Cabinet', 'Base_Drwr', 'Control_Panel', '3_tier_shelf', 'CDS', 'Cylinder_Restrain', 'Shelf_Stainless', 'EN849SL',
91 'Metal_Panel_Assembly', 'Knee_Space', 'PROF', 'Corner_Guard'])
92 """
93
94 REVIT_LAYERS = ('Furniture':set(['furn', 'pix', 'case', 'equi', 'SPCC']), '2':set('door', '0', 'giaz', 'stis', 'wind')),
95 '3':set(['wall', 'cols']), '1':set(['wind', 'arch']))
96
97 A dictionary for the script to identify and organize layers in exported dwgs.
98 When organize_layers() runs, everything in the layers containing the keywords in a "set([])" will be moved to the layer whose name precedes ":" of the set.
99
100 Format: REVIT_LAYERS = {'LAYER_NAME_1':set(['DWG_LAYER_KEYWORD_1, DWG_LAYER_KEYWORD_2']), 'LAYER_NAME_2':set(['DWG_LAYER_KEYWORD_1,
101 DWG_LAYER_KEYWORD_2'])}
102 """
103
104 """
```

```
171 LEGEND_MARKETING_LINE_LAYERS = {'NorthArrow_Bold':('display_color':(0,0,0), 'print_color':(0,0,0), 'print_width':0.5),
172 ... 'NorthArrow_Light':('display_color':(190,190,190), 'print_color':(0,0,0), 'print_width':0.085),
173 ... 'ScaleLines':('display_color':(0,0,0), 'print_color':(0,0,0), 'print_width':0.125),
174 ... 'ScaleNumbers':('display_color':(0,0,0), 'print_color':(0,0,0)),
175 ... 'LegendHatches':('display_color':(255,255,255), 'print_color':(0,0,0)),
176 ... 'LegendTexts':('display_color':(0,0,0), 'print_color':(0,0,0)),
177 ... 'LegendFrames':('display_color':(0,0,0), 'print_color':(0,0,0), 'print_width':0.125),
178 ... 'PrintFrame':('display_color':(0,0,0), 'print_color':(0,0,0), 'print_width':0.01)
179 ... }
180 ...
181
182
183
184
185 ######
186 # DO NOT MODIFY ANY CODE BELOW #
187 #####
188
189
190 def find_blocks():
191     """
192     ... Select visible and not locked instances of blocks whose names contain keywords in BLOCKS_KEYWORD
193     ... Does not work for blocks nested in other blocks
194     ...
```

Do not edit anything below this note.

## Presentation Helper's Functions and Work Flow

The work flow suggested behind the scripts' functions are as the following:

Different drawings should be kept in different files, since the script may not edit the drawing that is intended to be edit. They can be copied to one file before step 8.

1. Delete all block instances (former family objects in Revit) not necessary for presentation drawings.

To select (and later delete) instances of block definitions, explode nested blocks (lab equipments are usually nested in one block), set **WHAT\_TO\_DO = 1** and click the green triangle (Start Debugging) in the tool bar. When the script runs, every instances of the block definitions whose names contain a keyword as indicated in **BLOCKS\_KEYWORD** will be selected. Review the selected objects and delete those not needed.

If some block instances are not selected as expected, edit the keywords in **BLOCKS\_KEYWORD** following the format instruction in green. When a block instance is selected, its definition's name can be access by typing "BlockManager" or "What" in the command line.

If some block instances to be kept are selected, select one instance, set **WHAT\_TO\_DO = 1.1** and run the script. The corresponding keyword(s) will be printed in command window. Then use Ctrl+F to find and remove the keyword(s) from **BLOCKS\_KEYWORD**.

This function can always be used before the block instances are turned into groups in step 4.

2. Replace multiple block definitions of the same geometries generated during exporting dwgs with a single one.

When a dwg is exported from a Revit model, instances of the same family but different orientations will become different block definitions. Therefore, this step is to replace the various instances of a single definition so as to allow changes in one instance to happen globally. To do this, set **WHAT\_TO\_DO = 2** and run the script. No editable constant is used in this step. After the script runs, the new replacement will be in a group named "new," while the block instances being replaced will in a group named "old." Review and adjust the objects in each group before delete the old group and ungroup the new instances.

As of version 20190109, the script is unable to replace the same family(block) instances that are mirrored. That is, each family used in the document will be represented by two block definitions at most. Also, in this version, the script takes a while to run (~2min for a document with ~500 block definitions).

3. Simplify block definitions that are too complicated for presentation drawings so that all instances of these blocks reflect the change.

There is no script involved in this step. Block definition can be edited by double-clicking an instance of it, or type "BlockEdit" in the command line with the instance selected.

### Note:

- When all block instances are turned into groups in the next step, no changes in a former block instances will happen globally. Therefore, please take sometime at the end of this step to make sure edits in block definitions are done.
- There is no need to delete dashed lines representing overhead geometries in this step. In step 5, all the dashed lines will be moved to a layer that can be turned off.

4. Explode each block instances in the document to a group so that they are easy to edit and can be moved to other layers.

To turn each block instance in selection to a group, set **WHAT\_TO\_DO = 3** and run the script.

5. Organize the drawing by line weights and colors and delete unnecessary block definitions and layers.

To move lines and hatches to preset standard layers, unlock and unhide objects to be moved, set **WHAT\_TO\_DO = 4** and run the script. Standard layers indicated by **MARKETING\_LINE\_LAYERS** will be added to the document. The objects on layers with the keywords indicated by **REVIT\_LAYERS** will be moved to their corresponding standard layers. Color hatches will be identified by their colors and all the text objects will be moved to the standard layer for labels. Adjust **REVIT\_LAYERS** following the format instruction in green if needed.

As of version 20190109, hatches for "Circulation Indoor" cannot be moved to their corresponding standard layer since the color is interpreted as black in Rhino. These hatches need to be manually moved to the standard layer.

## Presentation Helper's Functions and Work Flow

6. Adjust and clean up lines/hatches on each layer based on one's judgment and understanding of the project.

Move objects to other layers as needed for shown/not shown or different line weights (e.g. lab equipments and exterior walls). Redraw the stairs. Clean up complicated line works. This step is mainly manual but there are some functions to help with the process.

To remove empty layers (with no objects on them), set **WHAT\_TO\_DO = 6** and run the script. After this, the objects not on standard layers can be selected from the remaining layers and moved to standard layers.

Note that all unused block definitions and the empty standard layers will be deleted too. Standard layers can be restored by setting **WHAT\_TO\_DO = 7** (for line works) or **WHAT\_TO\_DO = 8** (for color hatches) and running the script.

7. Clean up extra lines in the walls

To clean up "double walls," use CurveBoolean to keep only the outline of a continuous wall. If the lines of a continuous wall are not enclosed, group them, select the group, set **WHAT\_TO\_DO = 5** and run the script. It will extend the lines to a closest other to form an enclosed shape for CurveBoolean. Make sure room-sized shafts are kept in the process.

Note:

- The script can work for multiple groups selected
- The script only extend lines in selected groups. If there are lines missing, the script will not be able to produce a group of enclosed lines.
- Nurb curves will be transformed to polylines.
- The script will not run for nested groups
- It is possible that the script fails to produce enclosed lines for CurveBoolean. Run it again might help but for some cases manual edition is necessary.

8. Clean up and move lines that are not picked up in the previous steps. Copy all the drawing to one file and align them for the export to Illustrator.

Adjust wall lines on two layers with different line weights(Linework\_4 and Linework\_3). The script can only tell exterior walls from interior ones, based on the information provided by the revit file, so it is necessary to have a look at these lines and move them around. For sections, the cut lines need to be manually added in. CurveBoolean might be helpful in the process.

Note:

- Do not run the script on the file with multiple drawings as it might not behave properly.

9. Add legends and scales so that the drawing is ready to be exported to Illustrator where final touches will be made

Select the objects in the drawing, set **WHAT\_TO\_DO = 10** and run the script. The script will calculate the scale(between 1/4 and 1/200) and add a scaled printed size frame(default landscape letter), north arrow, scales and legends to the lower left corner of the frame. Move them around as needed. If no scale can be found for the drawing, nothing will be added.

Note:

- If the script is run in Rhino 5, all text objects generated will shift along the Y axis. This can, however, be easily fixed manually.
- For plans, run the script on the largest floor plan and copy the frame to other plans to make sure the plans and legends appear in the same place. North arrow, scales and legends can be copied to other plans in Illustrator after all layers are organized.

There is, of course, no need to adhere to this work flow as long as the script is still helpful.

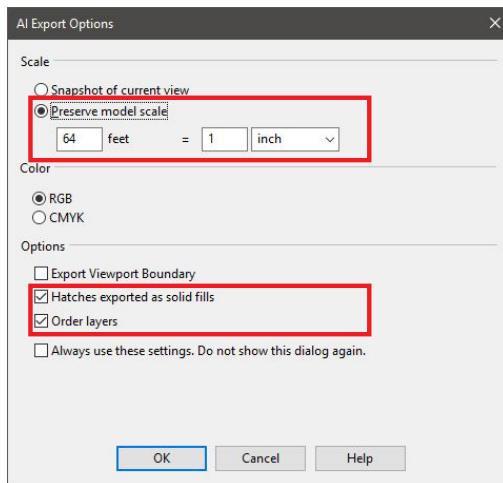
## Exporting to and Editing in Illustrator

Before exporting, make sure all the lines and hatches to be shown are unlocked and visible. Keep the drawing near the origin as it might fall out of the editable area in Illustrator if it is too far.

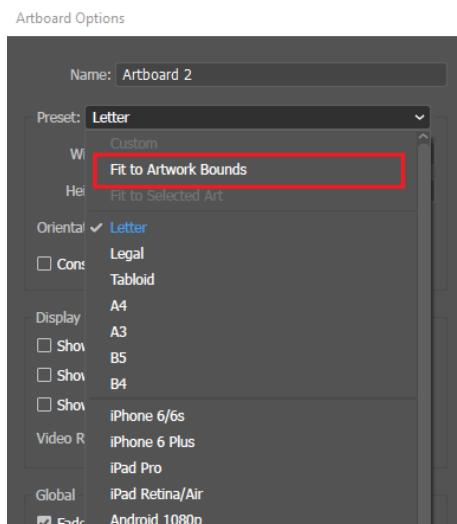
When a drawing is ready to export, select the drawing with its print frame and export it to an Illustrator file.

Select Preserve model scale and input the scale of the drawing.

Check the two boxes for easy editing in Illustrator.

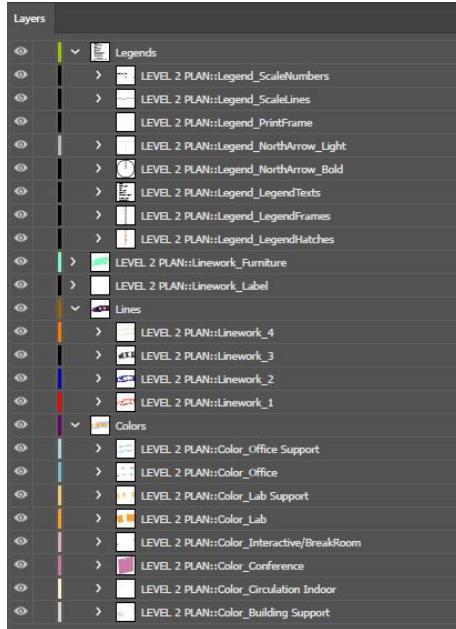


Edit Artboard Options and select Fit to Artwork Bounds. The artboard will snap to the frame set up in Rhino. After setting up the artboard, delete the frame.



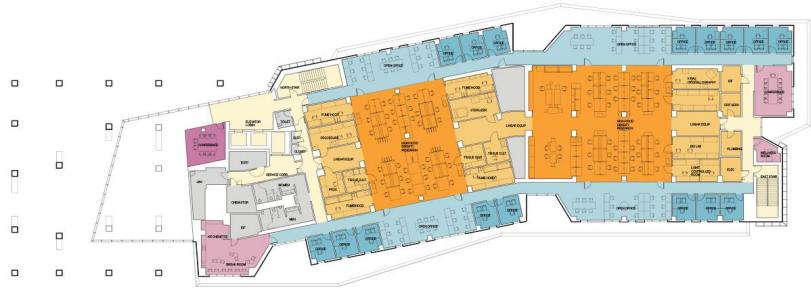
## Exporting to and Editing in Illustrator

Organize layers for future turning on/off in pdf.



Set all lines to black and change line weights according to marketing standards.

As of version 20190109, the script set up standard layers with line weights for plans at 1/64 scale. That is, if the plan exported to Illustrator is in 1/64 scale, there is no need to change the line weights.



Save the final drawing as a pdf.