BLD122445

AutoCAD Versus Revit—Common Annotation Tips and Tricks

David Butts, BIM Specialist
Gannett Fleming, Inc

Learning Objectives

- Learn how to define AutoCAD annotative text, multileaders, and dimensions to match Revit annotation types
- Understand basic similarities between AutoCAD Dynamic Blocks and Revit 2D symbol families
- Review specific Dynamic Block actions and features that emulate Revit behavior
- Examine how Dynamic Block visibility and lookup table features are similar to Revit family types

Description

When you have a lot of old-school and productive AutoCAD users, sometimes it can be tough to get them into the Revit way of thinking. One way to get these users on board is to help them relate AutoCAD features to Revit tools, and learn how these similar tools can increase their productivity. In this lesson, we'll begin by learning how the scale of the drawing controls annotations such as text and dimensions. Next, we'll review the similarity of Dynamic Blocks in the 2D symbol and annotation families of AutoCAD software and Revit software. We'll examine how actions and parameters in AutoCAD help the user match Revit family placement behavior and features. The session will close by showing how to make AutoCAD Dynamic Blocks behave more like Revit family types, using visibility and lookup tools.

Speaker

David Butts is an Autodesk Expert Elite Team member and Building Information Modeling (BIM) specialist for Gannett Fleming with over 30 years of experience in the architecture, engineering, and construction field. He is responsible for implementation, training, BIM project support, and management for engineering design applications, including Revit, AutoCAD P&ID, AutoCAD MEP, Navisworks, and more. He was an Autodesk Authorized Training Center (ATC) training manager and application engineer for an Autodesk Reseller for 13 years, providing implementation and training services across the United States, and serving as a subject matter expert for Autodesk, Inc.'s, Building Design Solutions. He has design experience for a variety of project types, and is an Autodesk University top-rated speaker for labs and lectures. He authors training videos for 4D Technologies, and he presents BIM topics for other industry associations annually.

Introduction

Communicating design intent has taken on many forms over the years.

From chalk on a rock…

To ink and linen, and mylar…

To pencil and vellum…

When we started down the AutoCAD path, the beauty of the program is that it gave us many options. Whether you were drawing actual size, or scaling a detail, you could always edit the style of text to be a specific size. In my earlier days at Piedmont Olsen, the CAD management team developed a scale factor lisp routine, which would take the printed height of text and multiply it times the scale factor of a drawing, which a user selected from a menu list.

In AutoCAD 11 (sorry not 2011…*11!*), Autodesk introduced **paper space**, which allowed the sheet of paper to be in a separate environment from model space. This allowed you to continue drawing linework at "actual size" – meaning that the line representing a wall was drawn the actual length, and another line offset to represent the other side of the wall or wall component.

The sheet of paper would remain actual size as well, and a "viewport" added that could be set to the scale factor needed for the drawing to fit on the sheet.

AutoCAD 2007 introduced **annotation scaling** to a DWG file, which incorporated the scale factor tools we had been using for years. This way, any annotative object – text, dimensions, 2D symbol blocks, etc. – could be drawing at symbol size for the sheet of paper, and then scaled as needed for the AutoCAD paper space viewport. Later enhancements allowed multiple instances of the annotative object to be placed at various locations, which allowed for more clarity when the scale of the drawing was changed.

Revit has always incorporated the paper space/annotative scaling behavior, and utilized the "model space – view – sheet" environment. Instead of **styles**, the program called the objects **types**. But these items still represented the same type of object.

As an industry, we've been as divided as the current political climate, with passionate feelings about whether AutoCAD or Revit is a better design solution. With the industry moving inexorably towards the 3D design environment, there is still a high demand for 2D documentation, so the question becomes, where do you draw the line (pun intended!)?

At our firm, we've been guilty of one of the biggest industry sins – ignoring CAD training at the expense of modeling training. It's been easy to justify the expense of taking design team members to the BIM environment, but covering basic AutoCAD skills has always been assumed. We discovered that we had many users – from 30-year old-timers, to fresh out-of-school millennials that shutter at the thought of "drafting" instead of modeling. But the biggest loser has always been the annotation tools, which has led to the wide variety of methods employed today.

In this class, we want to take you back to the construction documentation part of the project, and help you gain consistency between how you produce the designs in the AutoCAD environment, while leveraging the same workflows and tools that are built into Revit. Some steps will be faster in AutoCAD, while others are quicker in Revit. To have a clear understanding of these tools, **both methods are covered in this document**.

You may have already adopted some of these methods, but we might be suggesting some changes you probably didn't think about, which also impacts external documentation from Word, Excel and more. Hopefully, these old dawgs will help you learn some new tricks, to help you improve the quality of your project deliverables – and move faster through the design process with a more consistent quality.

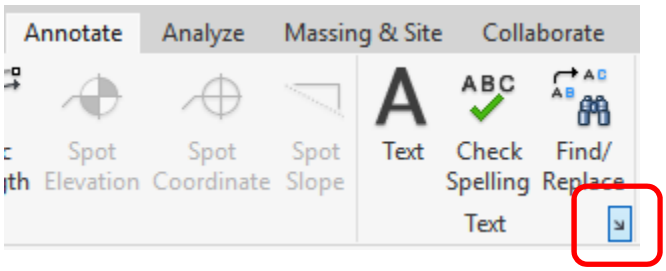
Matching AutoCAD Annotation Styles to Revit Annotation Types

When setting up your templates, start with text for your annotation standards.

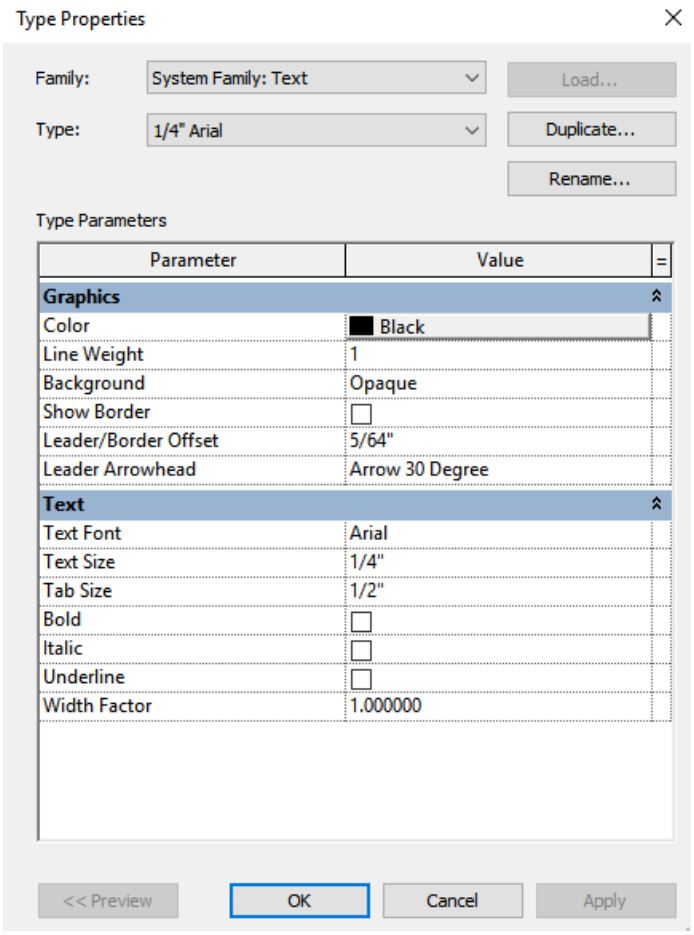
Text Styles/Types

The first step is to review the **Revit Text types** that are loaded into a project.

1. From the **annotation** tab, select the **text** tab, and use the **arrow** in the lower right corner to expand it and expose text types:



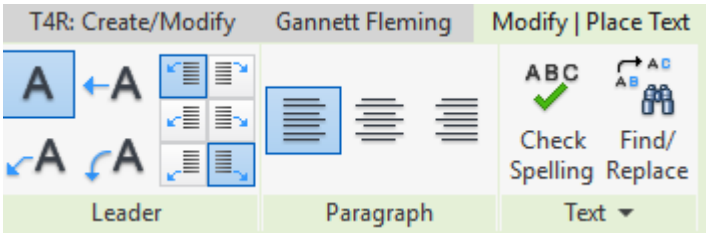
2. The **Text Type Properties** dialog appears:



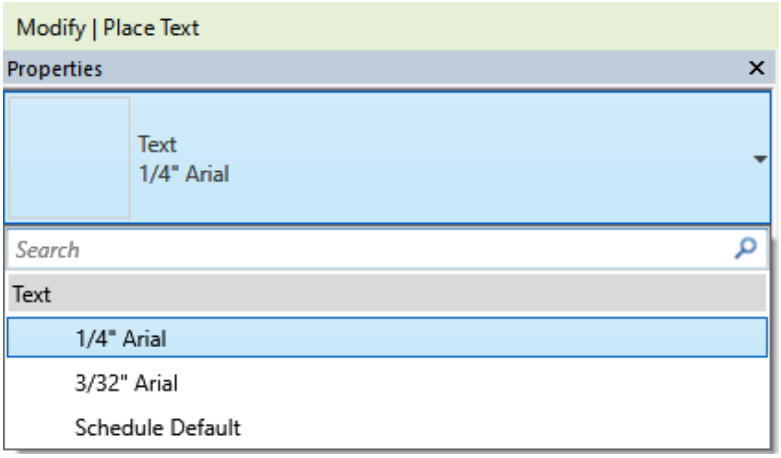
3. Review the key settings, including:

a) **Color** – sets the color of the text;
b) **Lineweight** – sets the lineweight of the text object;
c) **Background** – tells the text whether to mask objects behind the text (opaque) or keep the text transparent;
d) **Show Border** – draws a box around the text – as the text string changes size, the box will increase to match the width and height of the text;
e) **Leader/Border offset** – sets the distance from the edge of the text to the edge of the border when it's used, and to the start point of a leader when the leader is added;
f) **Leader Arrowhead** – sets the default leader that will be placed, when a leader is added to a text element based on this type;
g) **Text Font** – sets the font used;
h) **Text Size -** sets the font printed height on a sheet of paper (with the scaled size in a view based on this value times the scale factor)
i) **Tab Size** – sets the spacing for text in a note, as well as the indent for text lists;
j) **Bold/Italic/Underline** – set the additional behavior for the type
k) **Width factor** – compresses or expands the default width of the text element.

Revit text is placed like AutoCAD text. When you select the command from the Annotate tab, the modify/place text tab appears on the right side of the ribbon:



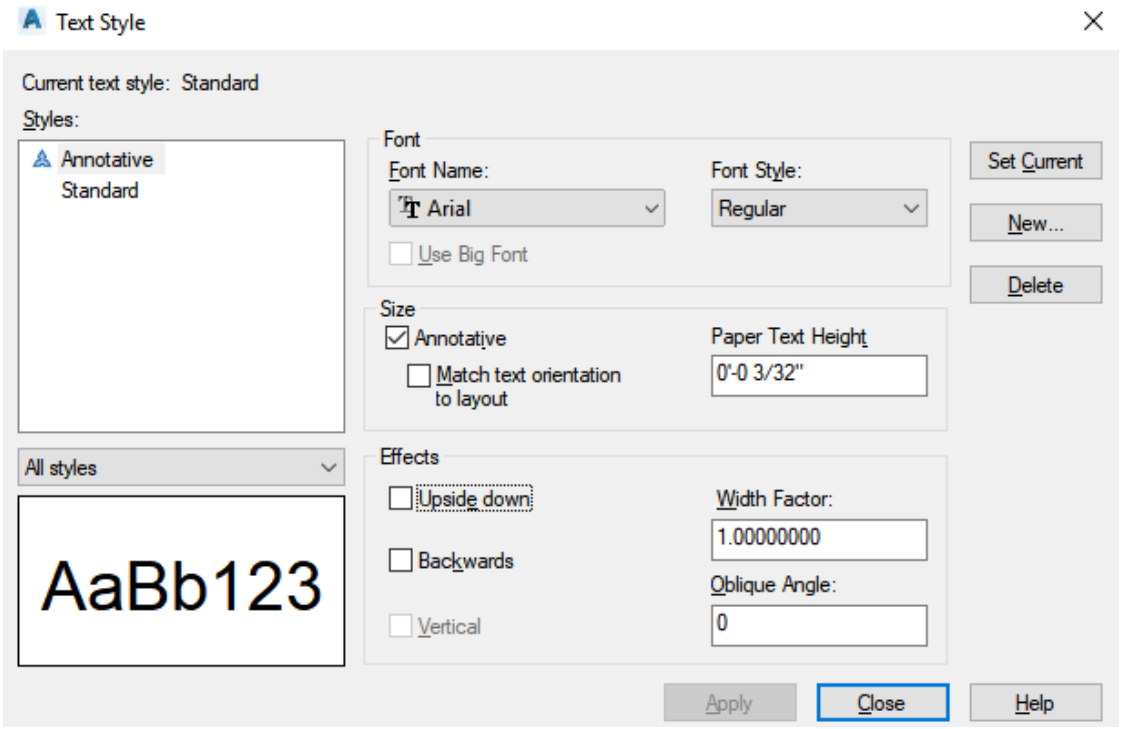
From here, you can select a **leader**, set **justification**, check **spelling** and **find/replace** text as needed. The **Properties** palette lists the types, and you select which type to use in the view:



4. When creating names for Revit types or AutoCAD styles, make sure you follow the same naming behavior from AutoCAD – avoid **special characters** such as "\" that can't be

used in an AutoCAD text style name. Stick with **True Type** fonts

In **AutoCAD**, **text styles** are defined using the **STYLE** command:



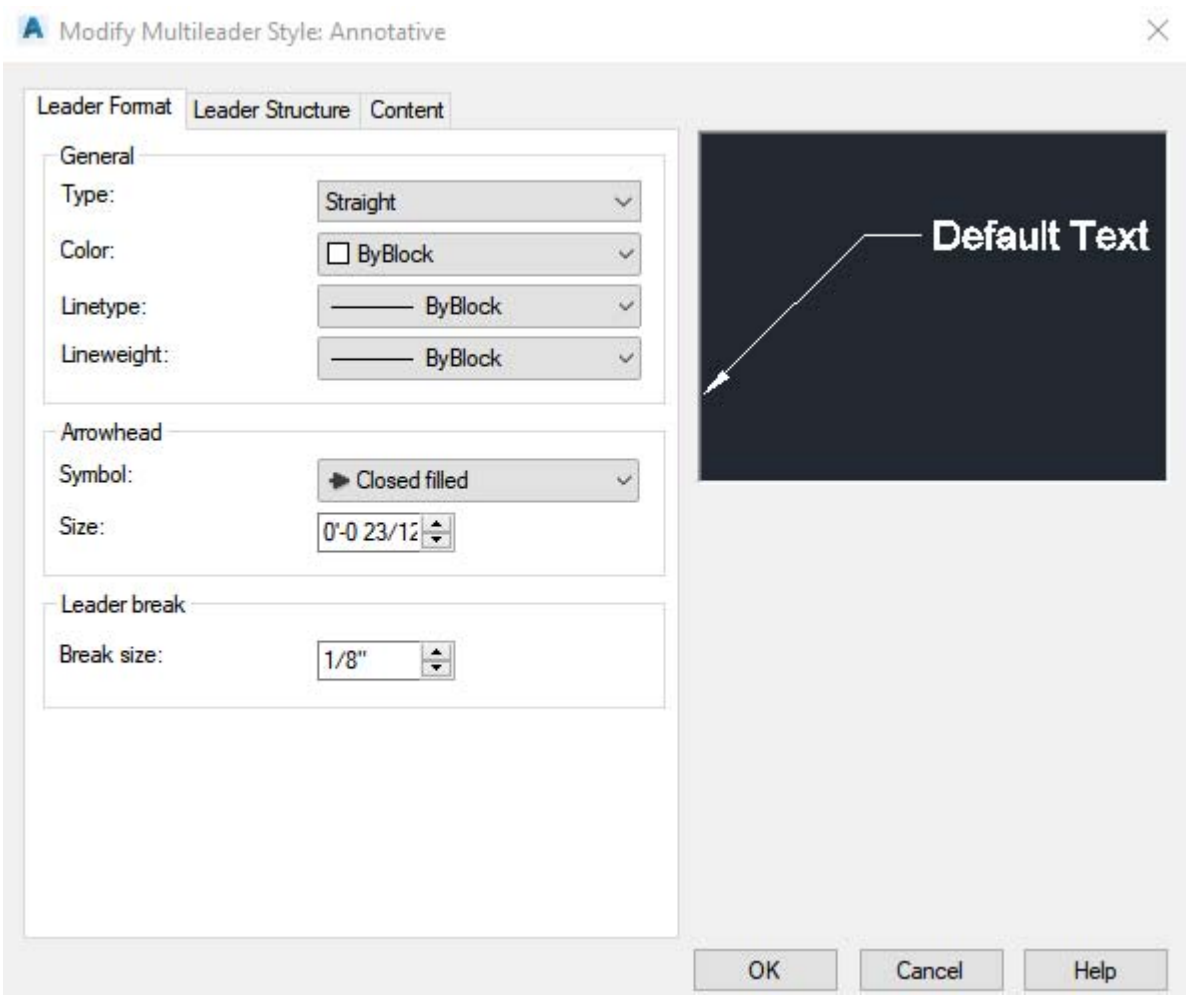
1. From this dialog, you set the current style, as well as:

    a) **Font** – sets the font used
    b) **Font Style** – adds options for regular, bold and italic
    c) **Annotative** – the key feature, this tells the text to respond to the scale assigned to model space and the paper space viewport;
    d) **Paper Text Height** – this also is critical – while assigning this value is not required, you'll want to go ahead and set it to get the same behavior as the Revit text type.
    e) **Effects** – include upside down and backwards, are not use in Revit, and are rarely used in AutoCAD.
    f) **Width factor -** compresses or expands the default width of the text element.
    g) **Oblique Angle –** an older command that mainly applies to traditional SHX files, and adds italicized behavior, this tool is less commonly used.

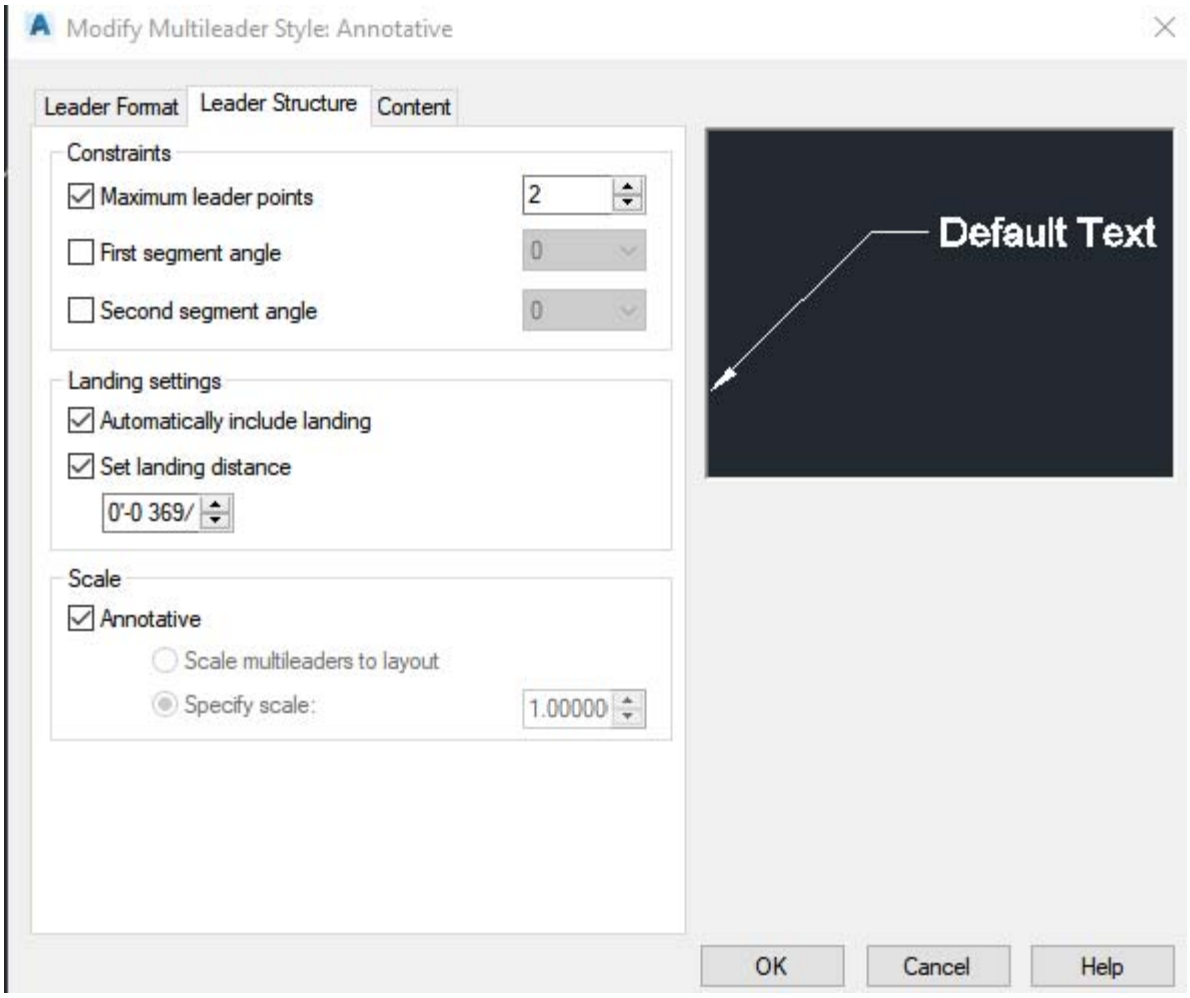
AutoCAD MLEADER versus MTEXT

Stick with MLEADERS in AutoCAD, since this tool also includes the leader as part of the text. This matches the Revit text behavior and gets users acclimated when the method is similar.

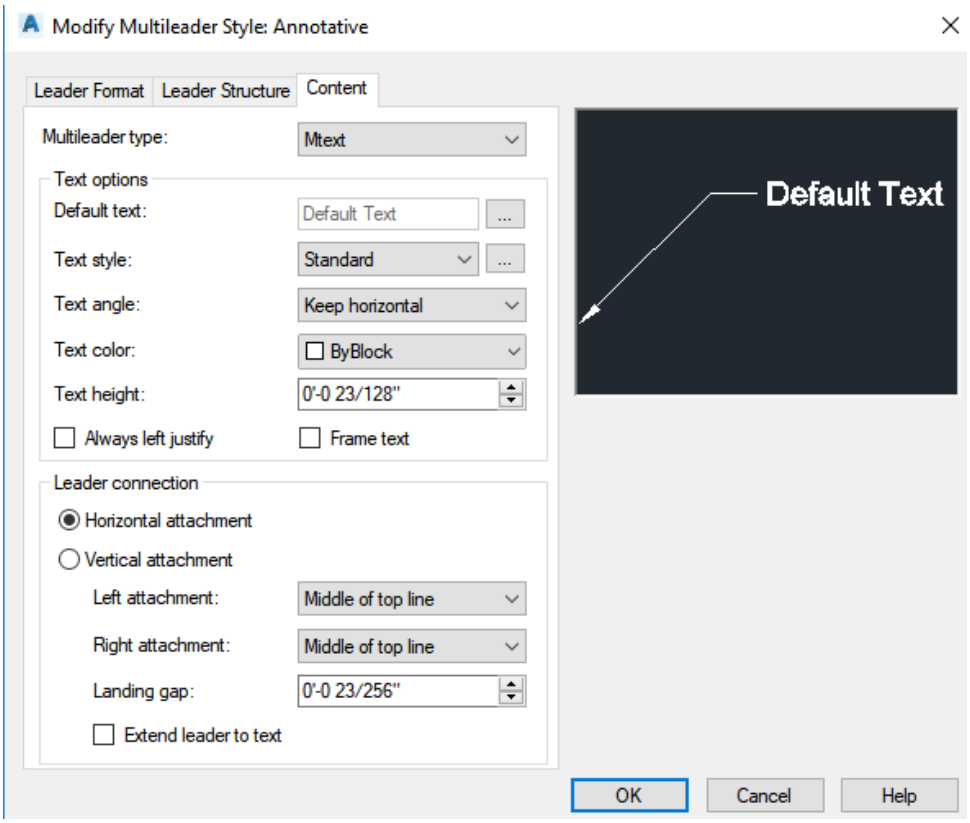
1. The MLEADERSTYLE command lets you set similar behavior and includes the following settings:



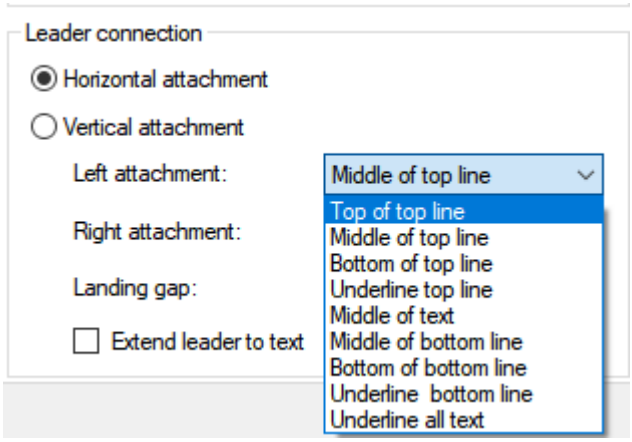
2. **Leader Format** allows you to choose the leader **type**, including **straight**, **spline** or **none**. It also lets you define the **color**, **linetype** and **lineweight** for the leader lines, as well as set the **arrowhead** symbol and style. The **Leader Break** setting is the same as the **Leader\Border offset** for Revit text.



3. **Leader Structure** provides a little more detail than the Revit text feature, as it defines constraints used regarding the number of leader line segment **points** and segment **angles**. The **Landing Settings** indicate whether the landing is included, and the **distance** of the landing line segment. **Scale** includes the key feature of telling the style to use **Annotative**, and respond to the scale of the drawing. Setting this matches the default Revit text type behavior.



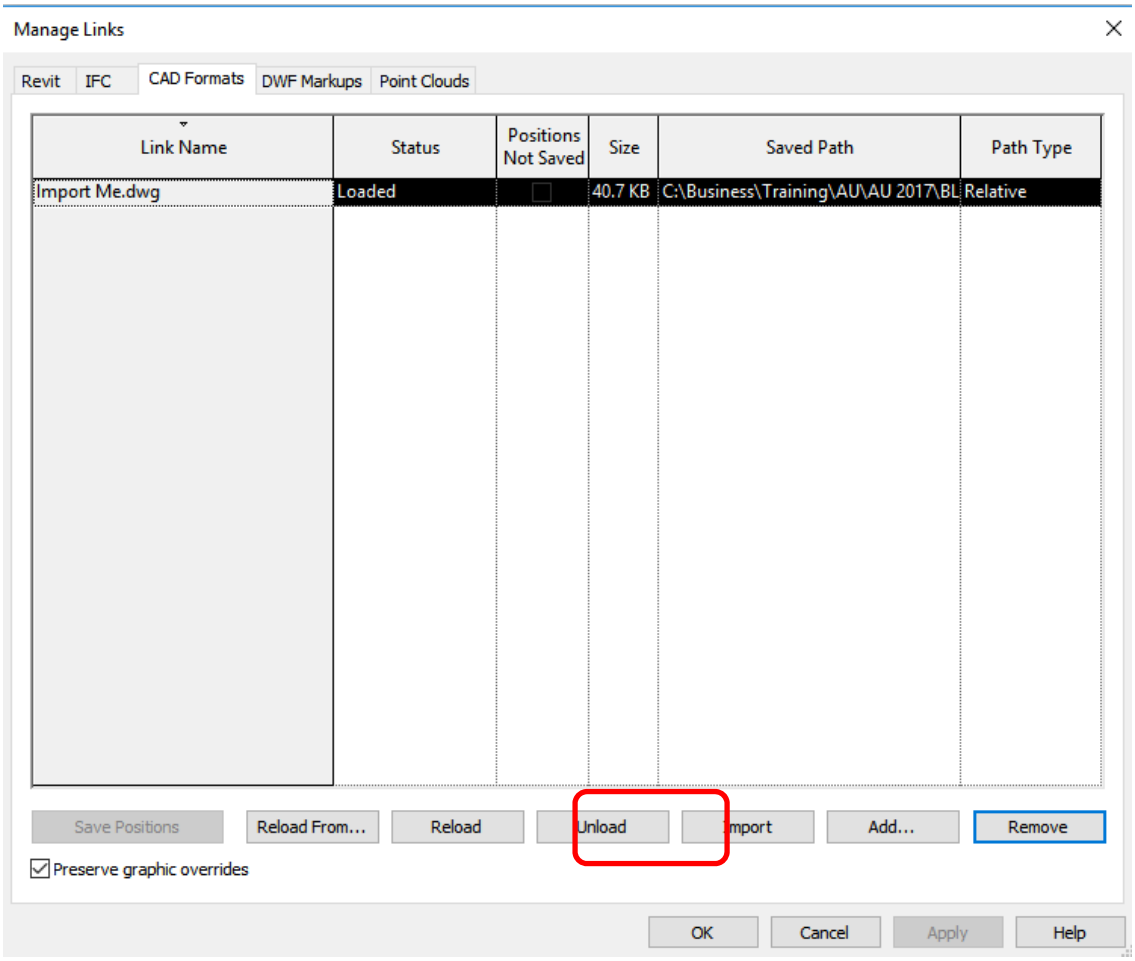
4. **Content** also provides a little more detail than the Revit text type, with the ability to set the multileader type. The options include Mtext, Block (for symbols) and none. Text style, angle and color are defined here, like the Revit text type. When the style has a defined height, the text height option will be non-editable. Frame Text provides the same type of box outline as the Revit text type, but Always left justify is an option specific to AutoCAD.



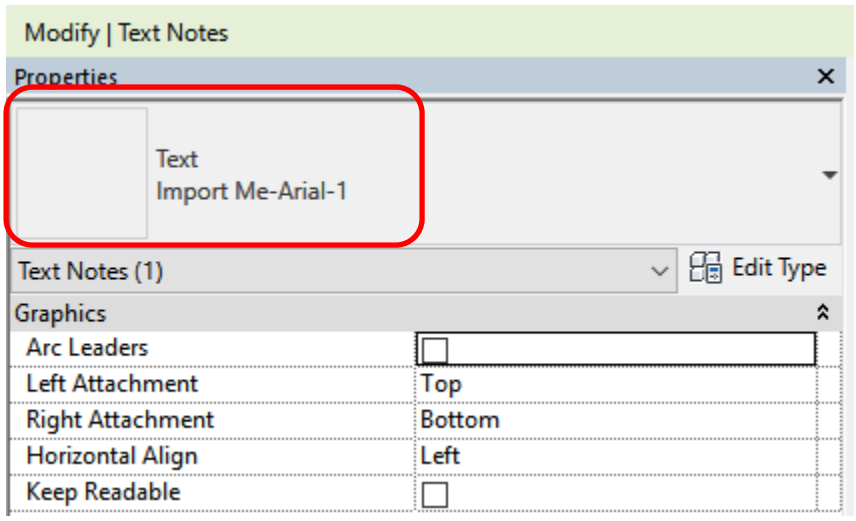
5. **Leader Connections** are like the options in the actual Revit text command, with additional options for the left and right attachments,

Additional Text Notes:

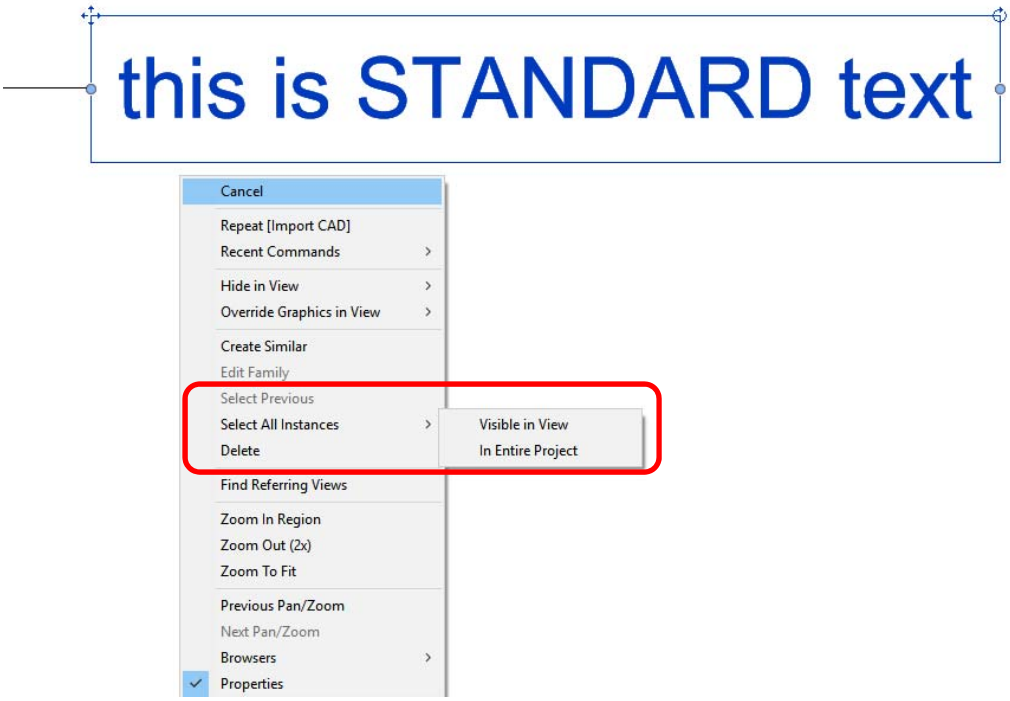
- When a drawing file is **linked** into Revit, the style is not imported. The style only becomes imported when the drawing is **inserted** as a non-linked file, or a linked DWG file is **imported** using the **Manage Links** tab:



Once the DWG is imported, it can be exploded – that's when the style is added to the drawing based on the **original drawing name** and **text font** used – not the Style name.



- To change the text style to match a Revit text type, the fastest way is to use the right click menu, and the **Select All Instances** tool. If a drawing is only present in the current view, you can choose in the current view, but you can also use the Entire project tool, if a drawing is visible in all views – and exploded (don't do this!)



- Where possible, use the same **style name** as the Revit text type, keeping in mind the limitations for the AutoCAD style name. For example – try using Word document text types, such as **Normal, Title, Sub-Header, Emphasis**. Get the focus away from the size to how the text is supposed to be used.
- Be aware – when text is **imported** from **Word** or **Excel** into AutoCAD or Revit, the text always used the current **style** or **text type** when pasted into a drawing or view. The

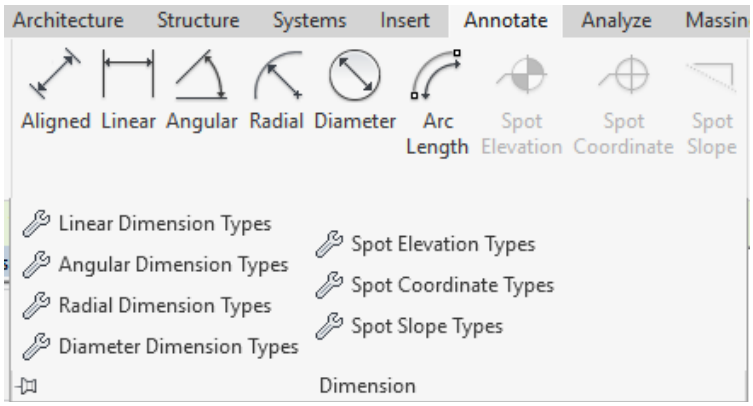
settings applied to the style will override any settings from the Word document, but items such as bullets and numbering will be maintained.

Dimension Styles/Types

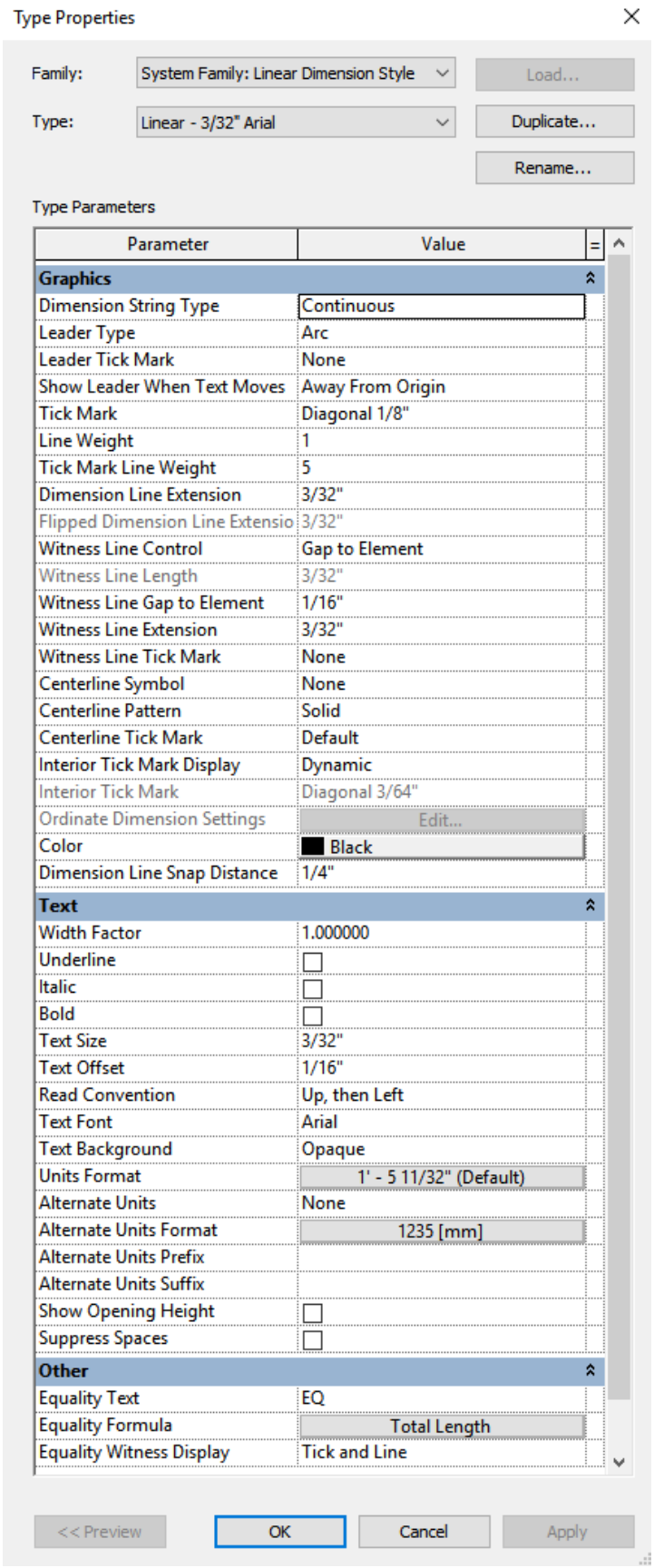
For dimensions, use the same rules as text – for example, use the same text type as defined for your default text.

Start by reviewing the **Revit Dimension Types**.

1. From the **annotation** tab, select the **dimensions** tab, and expand it to expose **dimension types**:



2. In Revit, dimension types are **separate** entities, based on how the dimension is used – **linear, angular, radial, diameter** and more are individual types within a project. Note the key settings (*next page*):

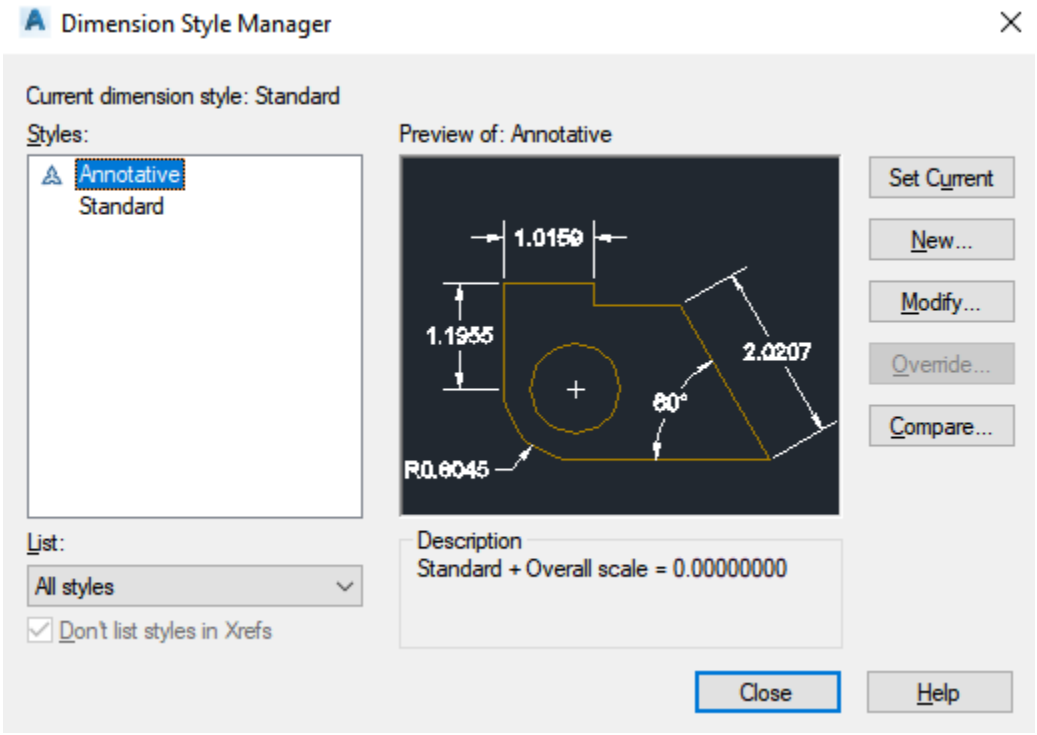

Type Properties
Family:
System Family: Linear Dimension Style
Load...
Type:
Linear - 3/32" Arial
Duplicate...
Rename...
Type Parameters
Parameter
Value
Graphics
Dimension String Type
Continuous
Leader Type
Arc
Leader Tick Mark
None
Show Leader When Text Moves
Away From Origin
Tick Mark
Diagonal 1/8"
Line Weight
1
Tick Mark Line Weight
5
Dimension Line Extension
3/32"
Flipped Dimension Line Extensio
3/32"
Witness Line Control
Gap to Element
Witness Line Length
3/32"
Witness Line Gap to Element
1/16"
Witness Line Extension
3/32"
Witness Line Tick Mark
None
Centerline Symbol
None
Centerline Pattern
Solid
Centerline Tick Mark
Default
Interior Tick Mark Display
Dynamic
Interior Tick Mark
Diagonal 3/64"
Ordinate Dimension Settings
Edit...
Color
Black
Dimension Line Snap Distance
1/4"
Text
Width Factor
1.000000
Underline
Italic
Bold
Text Size
3/32"
Text Offset
1/16"
Read Convention
Up, then Left
Text Font
Arial
Text Background
Opaque
Units Format
1' - 5 11/32" (Default)
Alternate Units
None
Alternate Units Format
1235 [mm]
Alternate Units Prefix
Alternate Units Suffix
Show Opening Height
Suppress Spaces
Other
Equality Text
EQ
Equality Formula
Total Length
Equality Witness Display
Tick and Line
<< Preview
OK
Cancel
Apply

3. Instead of multiple tabs in **DIMSTYLE** AutoCAD command, most of the settings are on one palette.

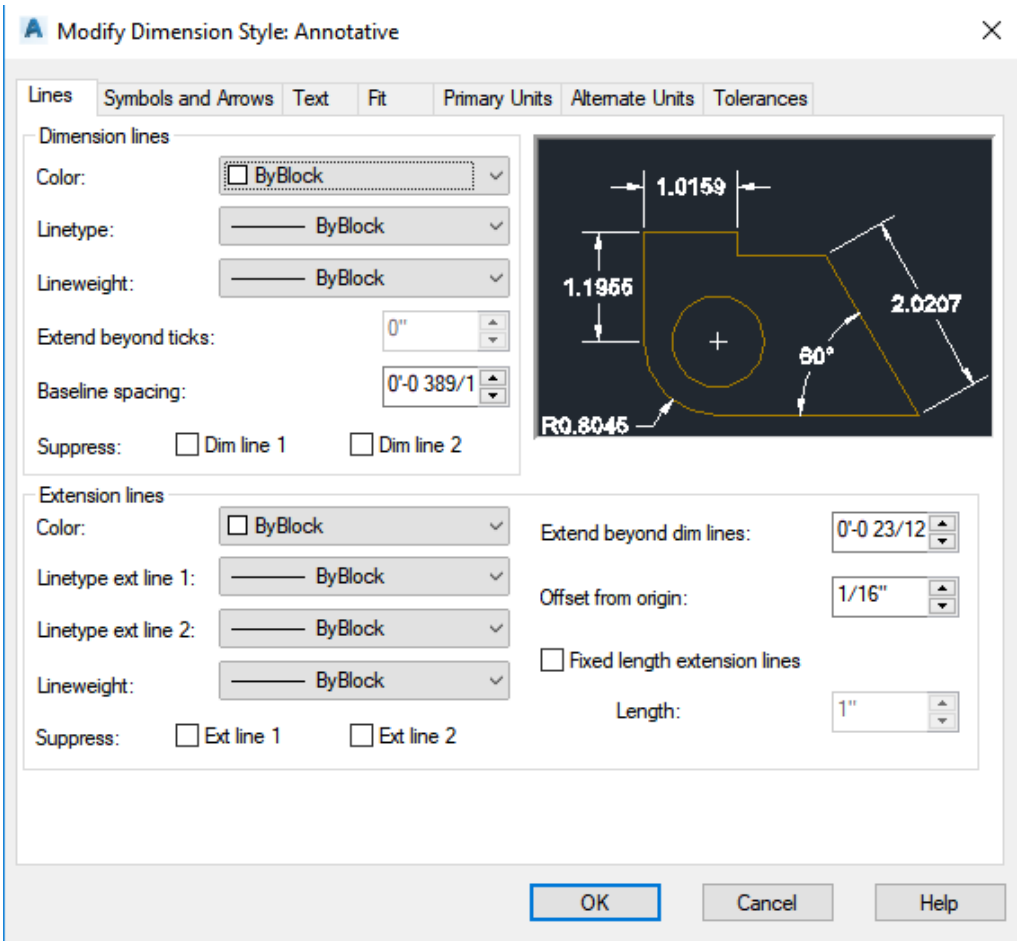
    a) **Graphics** control how parts of the dimension linework is defined, including the string type, leader settings, tick marks, witness lines and centerlines, color and gap settings.
    b) **Text** controls the text information – but in this case, does not require a predefined text type, like AutoCAD. The text settings are the same, with added features for **alternate units**, **read convention** related to the direction of the text, and **units formatting** specific to this dimension type.
    c) Other includes the equidistant or **equality** feature, which is unique to Revit. This feature allows you to use the dimension to space items evenly across a distance, or center reference planes along a single plane.

4. As with text objects, stick with **True Type** fonts in dimension **text**.

In AutoCAD, the **dimension** features are controlled by **style**, which can be accessed using the **DIMSTYLE** command:

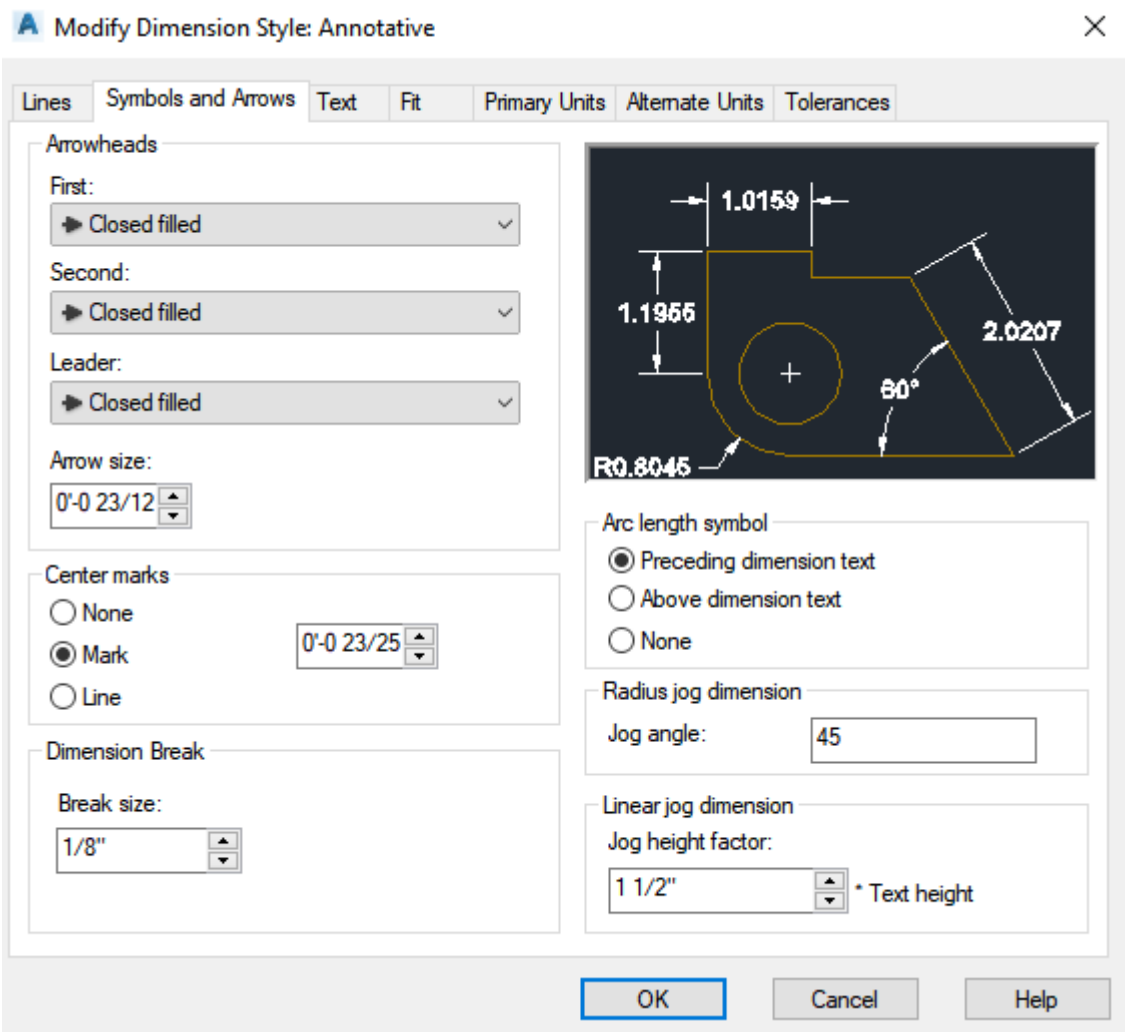


1. The dialog allows you to edit all types, but unlike Revit, **linear, angular and radial** dimension settings are defined by one **style**.



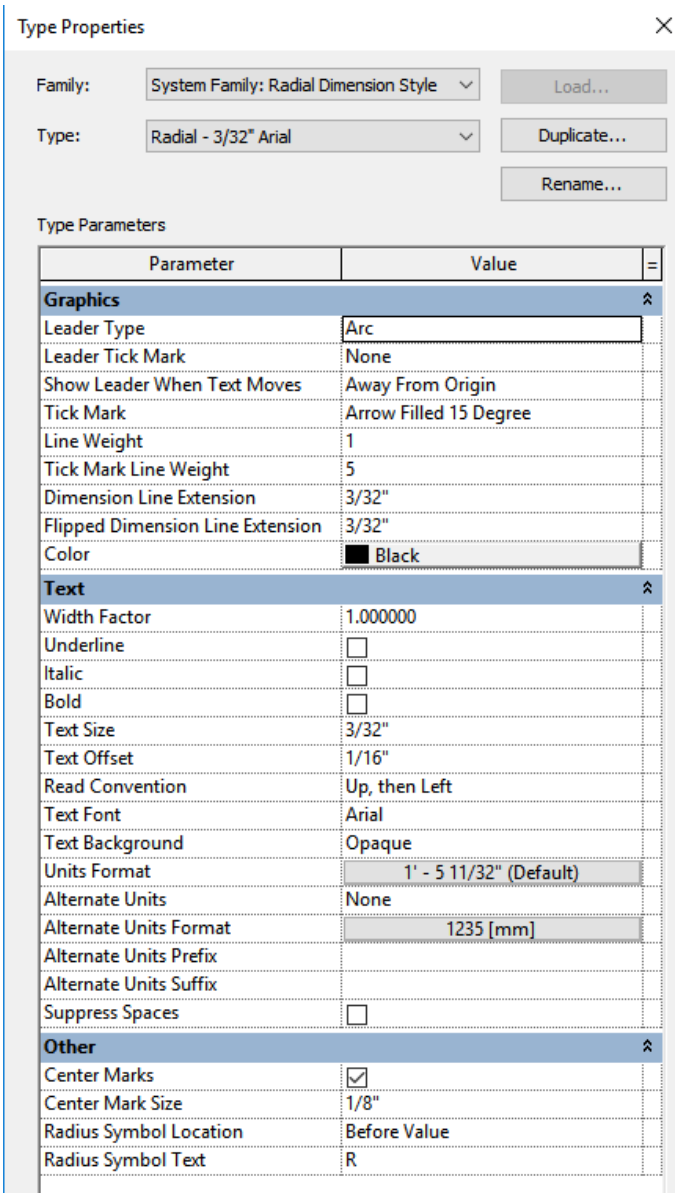
2. **Lines** controls the linework for the dimension, regardless of the type. Make sure you check settings such as **extend beyond dim lines** – this matches the Revit **Dimension Line Extension** setting in the type. Use BYLAYER or BYBLOCK for the color, linetype and lineweight settings, so the dimension line is controlled by the layer. This will also match the Annotation category settings for dimensions in Revit. AutoCAD dimensions will adapt to these settings when imported (and exploded) in a Revit view.

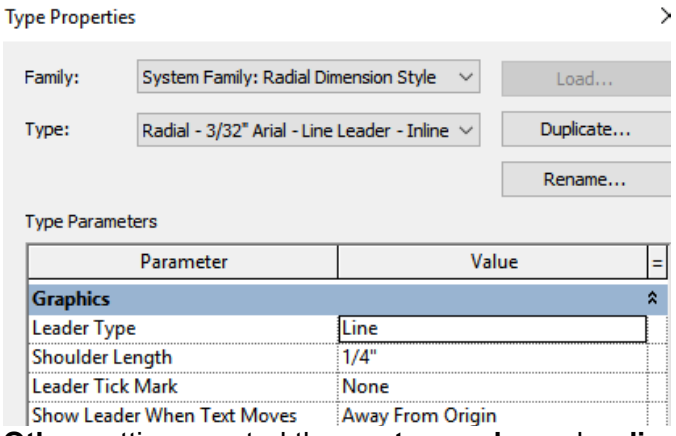
   TIP – Revit text and dimensions, when exploded, do not break down into individual components as AutoCAD dimensions will do. Avoid exploding dimensions in an AutoCAD drawing (or using non-associative dimensions) as these take on characteristics of detail lines and individual text, and lose their key functionality.



3. **Symbols and Arrows** set the **arrowheads**, **center marks** and **dimension breaks**, but also control **arc length**, **radius jog** and **linear jog** settings – which are not available in a Revit linear dimension style.

   Side step here to Revit – **Radial** and **angular** dimensions are separate Revit dimension types, and have their own unique settings:

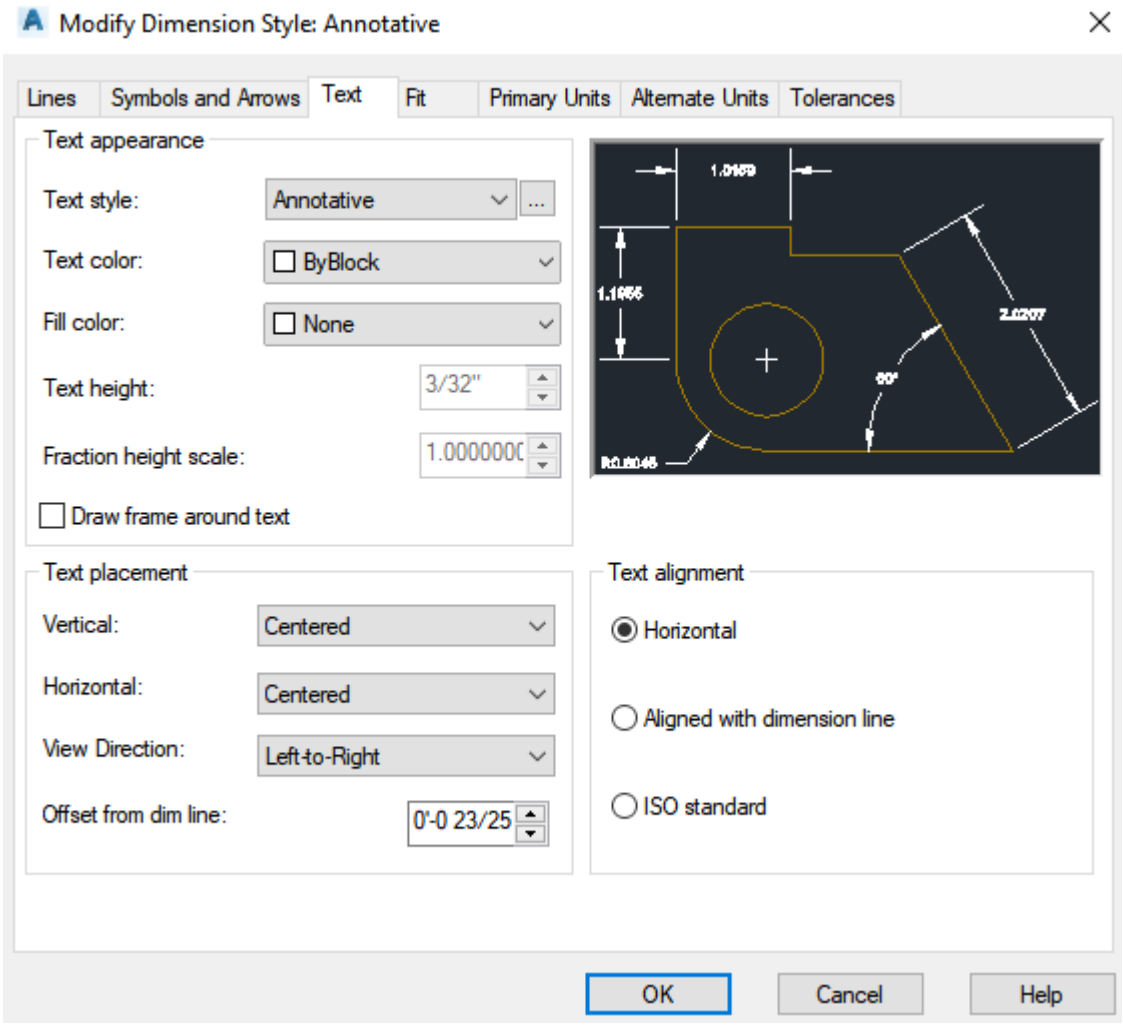




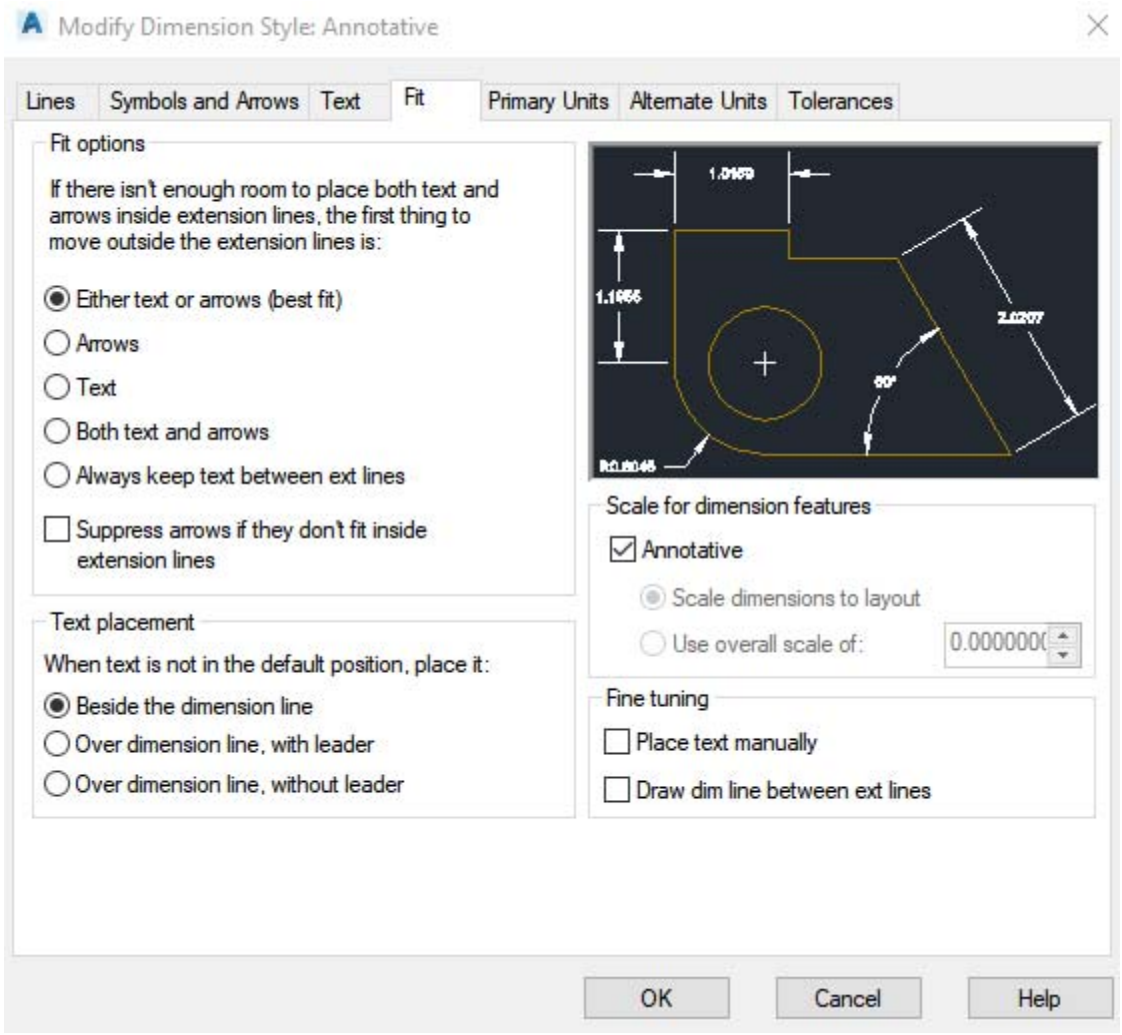
4. **Other** settings control the **center marks** and **radius symbol** location settings, while additional dimension types add **shoulder length** and **leader** settings when **inline** text is

used on radial and angular dimension types.

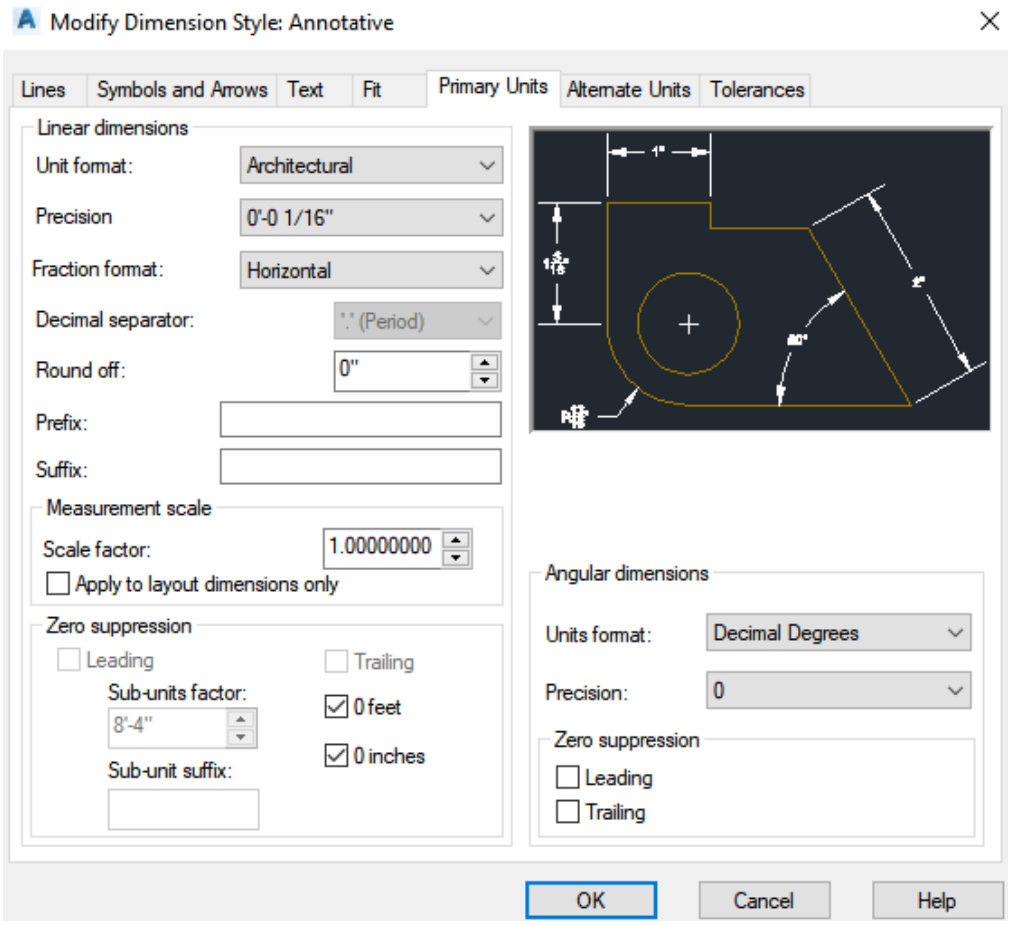
5. Back to AutoCAD, review the **Text** tab:



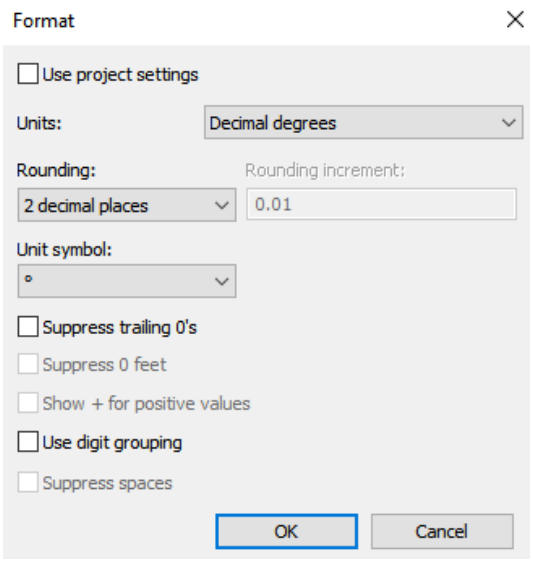
6. Selecting an **annotative** text style here automatically follows the text style, while the Revit dimension type still requires you to set the font and size, as well as the **draw frame around text** option. Text placement relative to the AutoCAD dimension is set here.



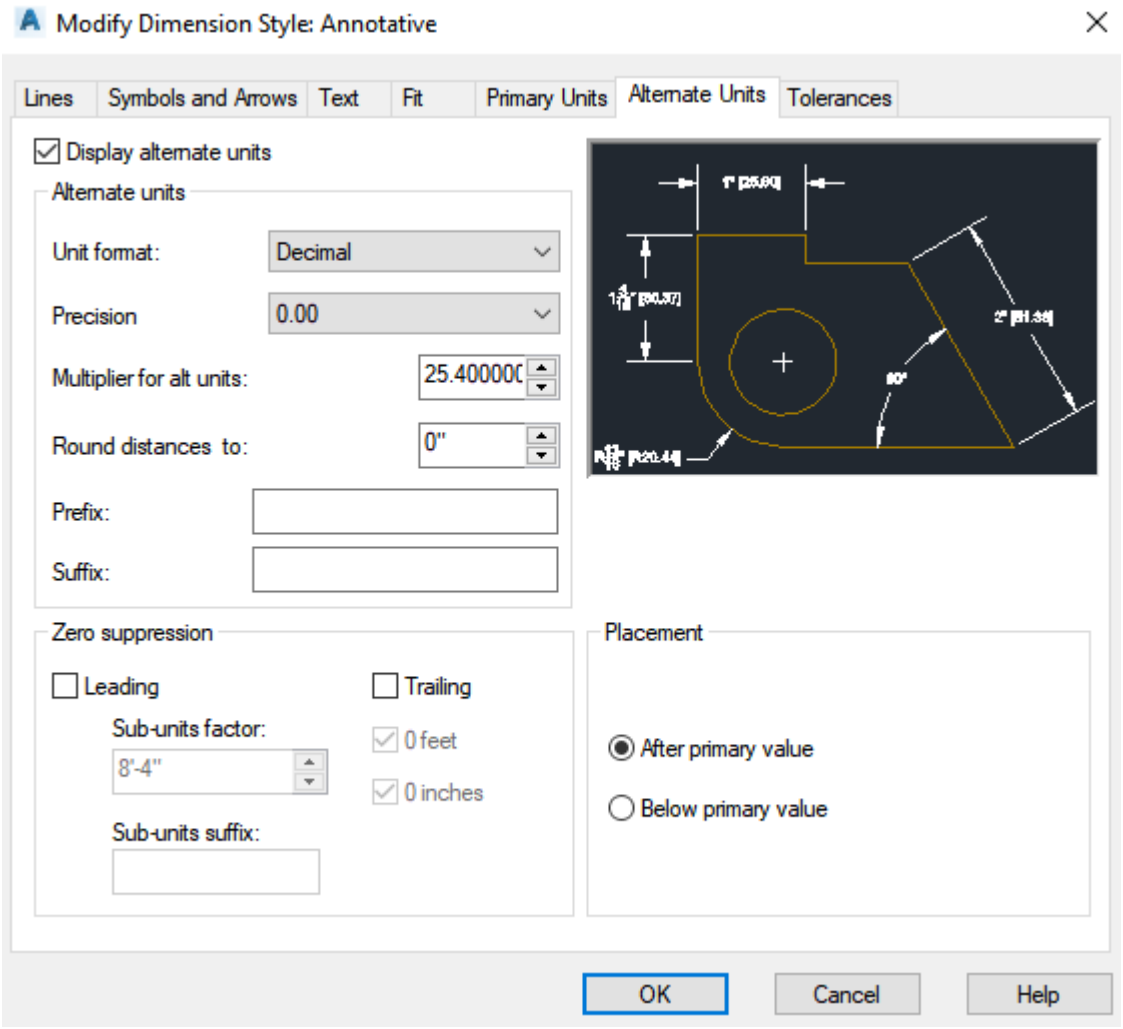
7. **Fit** includes the key setting for making the entire dimension style, and all its **size/offset** and **length** values, match the scale factor behavior as text. While the placement options are like the Revit **read orientation**, AutoCAD includes additional options for fit that are not included in Revit. An option to **place the text manually** during placement, as well as a **manual draw dimension** command – but these are also rarely used.



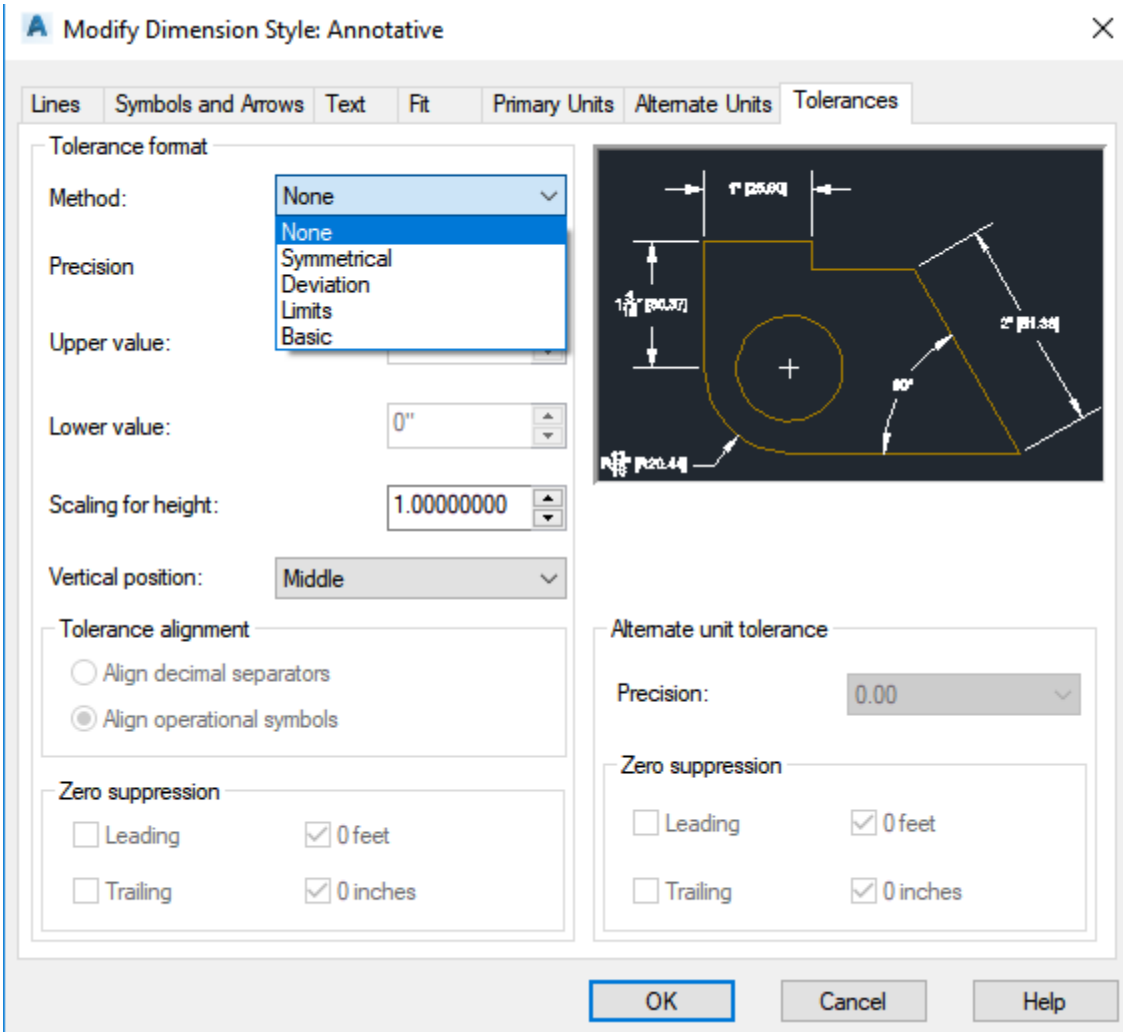
8. **Primary** and **alternate** units, which set **format**, **precision** and more, are controlled on a single tab in AutoCAD. While the **unit** settings are specific to a project in Revit, you can override specific items such as **zero suppression** here for AutoCAD. In Revit, the same feature exists under the **Units Format** option under the **Text** section of a dimension type:



9. Making changes in this dialog allow you to override the default **units** setting in a project, and the AutoCAD **primary units** tab provides the same functionality over the Revit Units command.



10. **Alternate units** are on the same palette as the Revit dimension type settings, but have their own palette in AutoCAD. These settings are used when multiple units (such as imperial and metric) should both be displayed in the dimension text.



11. **Tolerances** provides additional settings for the format method that are not exposed in Revit, but are typically used in manufacturing drawings more than in building design projects. Selecting one of the **method types** enables the distinctive features specific to the tolerance method.

Additional Text/Dimension Notes

- As with text, make sure the dimension settings, such as dimension line extension, tick marks, offset and more all match between the Revit type and AutoCAD style.

- What make these features work is using the same method as Revit – for example, always using associative dimensions in AutoCAD helps assure the behavior stays the same when imported into a Revit project.

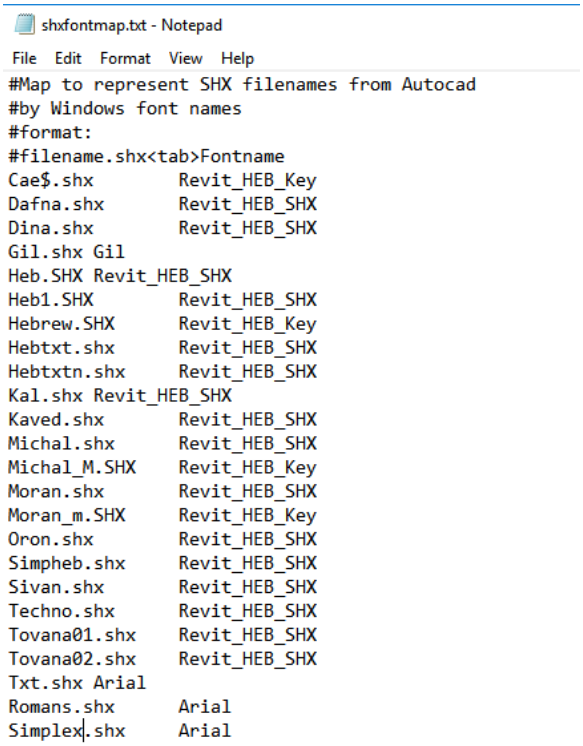
- Make sure that you always use **True Type** fonts. If a drawing is inserted into a Revit drafting view or other view with an SHX based font, it will still be recognized as text, but when the same view includes an exploded drawing, and is exported back to AutoCAD, the text **becomes line segments**.

If your drawing contains AutoCAD SHX files that you want to go ahead and convert to a TrueType font in Revit, locate the **shxfontmap.txt** file. In Revit 2018, the file is located here:

C:\ProgramData\Autodesk\RVT 2018\UserDataCache

The location in older releases will be similar – search the Program Data, Autodesk folder location, making sure you allow File Explorer to include hidden files:

You can edit the file with Notepad:



Simply copy and paste one of the default lines, such as **Txt.shx Arial**. Note that a **TAB** space exists between the shape file name and the Revit font. Change the SHX name to your **font** name and then edit the **type** after the tab key is used to create the space.

AutoCAD Dynamic Blocks and Revit 2D Symbol Families

Concepts – Dynamic Blocks versus Families

Annotations in a project are much more than just text, leaders and dimensions. A wide variety of 2D symbology can be used to represent physical items that are too small to appear in a large-scale print. Tags are another form of symbology that is used to label rooms, equipment and more.

In Revit, annotation families are used to represent these items, so they can respond to the scale of the view. Dynamic Blocks in AutoCAD can be used to create groupings of symbols, add parameters that allow quick plan view modifications, custom linework that can resize itself using grips, and much more.

The key is to understand when you need several examples of blocks, as opposed to actions that make the use of the block more efficient. Some of these actions include:

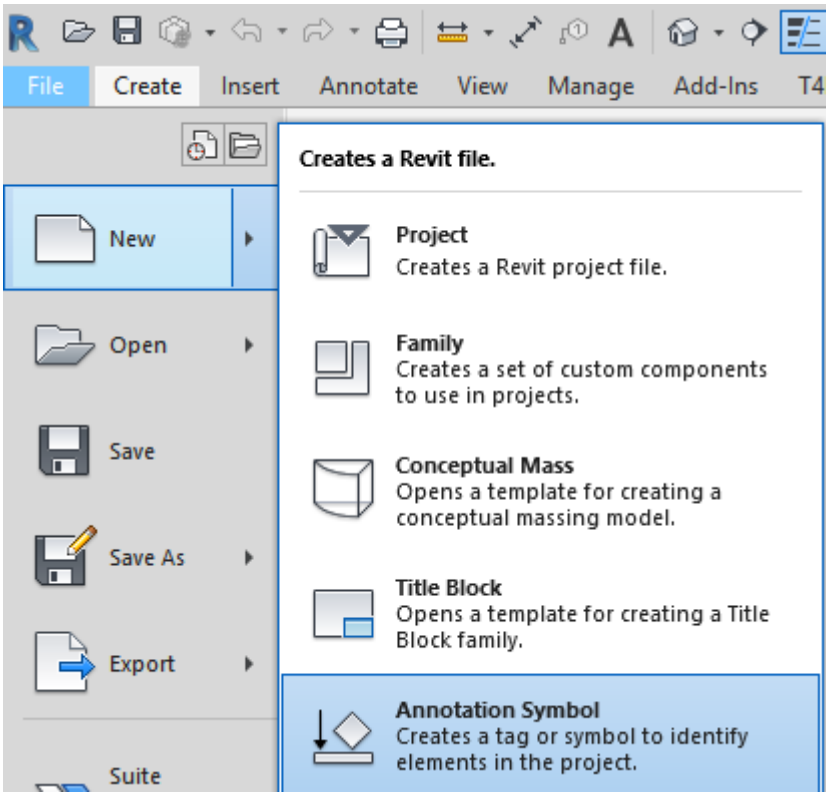
- Grips that allow you to quickly flip or align a symbol block, faster than selecting the object and the command;
- Editing the shape of a block per each instance to create assorted sizes;
- Leverage a series of nested blocks to show different symbols, using one dynamic block made up of several predefined blocks that can be toggled on or off.

In this series of exercises, we will cover the most commonly used types of actions, parameters, and parameter sets, and how they behave like Revit features.

Revit – Creating the Annotative Symbol Family

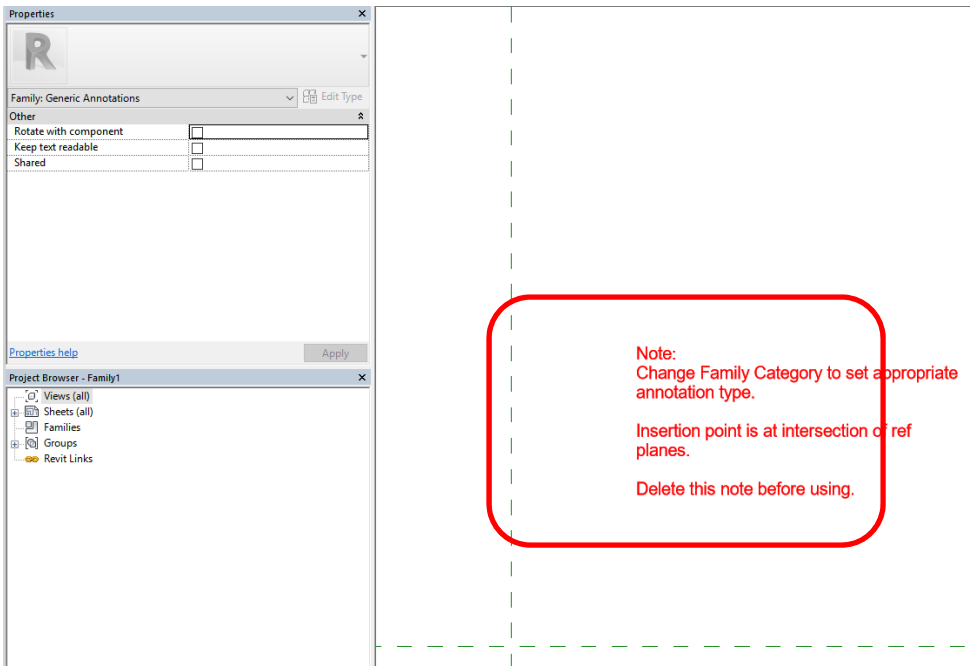
For Revit users, annotative symbol families can be nested into a model representing a small physical object, such as receptacle. The annotation symbol family will only appear in a plan view, so they are placed and defined in a ref level view in the family editor. To be annotative, you must start with an annotation family.

1. From the **Electrical Fixture – Wall Mounted – Non-hosted.rfa** file, review the family. Note that includes a parametric box that represents the box and face plate. At larger scales such as 1/8” = 1’0”, a 2D annotative symbol Is used as the industry standard to represent the receptacle. If the symbol is already located in the drawing, you can remove it to create and include your new symbol.
2. From the ribbon, **File** tab, **New** tool, choose the **Annotation Symbol** option:

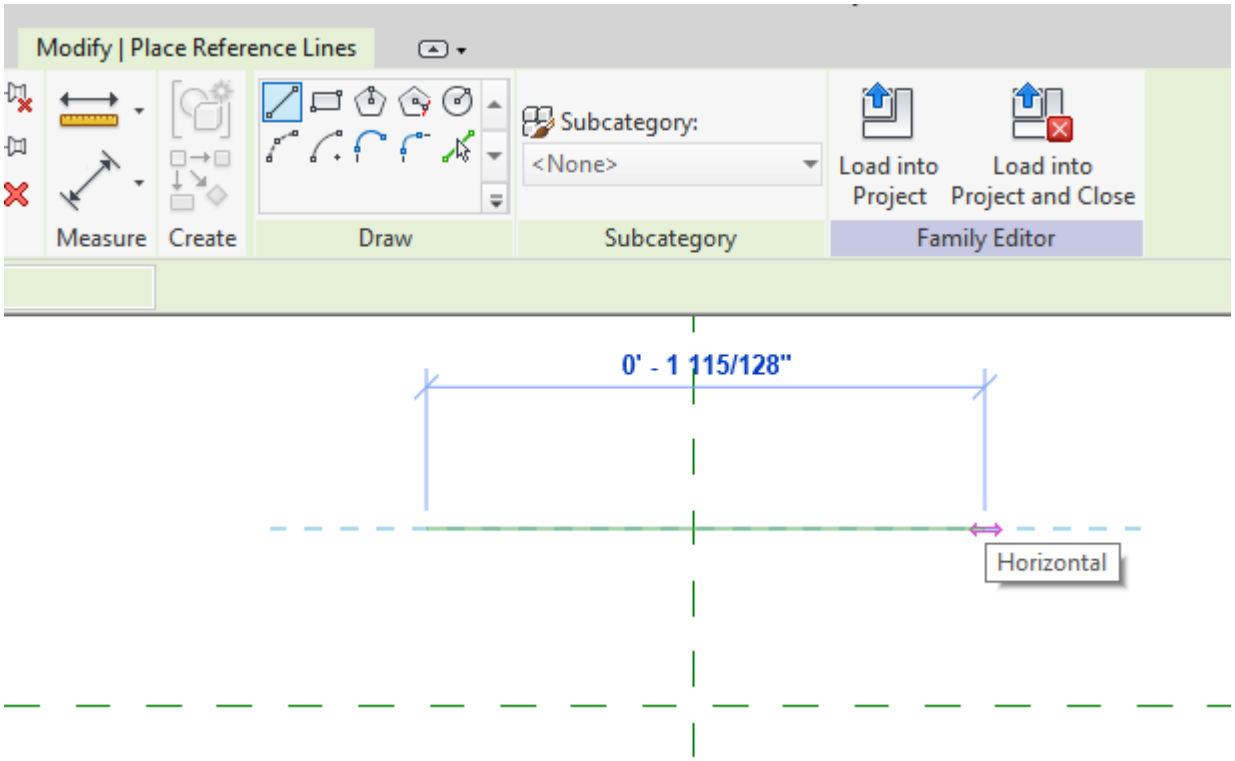


3. When prompted, select the **Generic Annotation.rft** template. Use this to create any symbol that needs to be annotative in a view. Click **Open** to create the file.

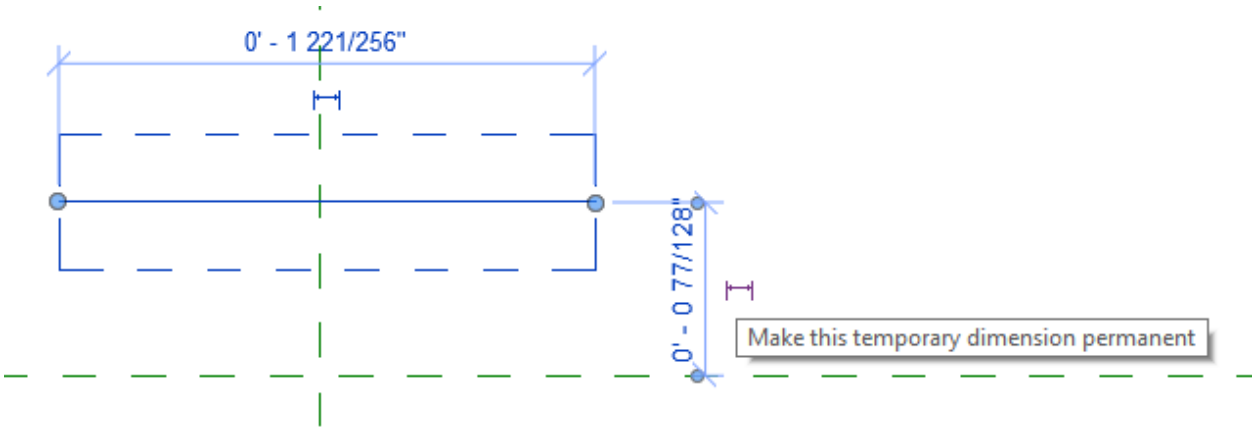
4. After the file is created, note that there are no other views in the file. Since annotation families are specific to the view they are placed in, only a plan view is needed. From the view, select the **text notes** and delete them:



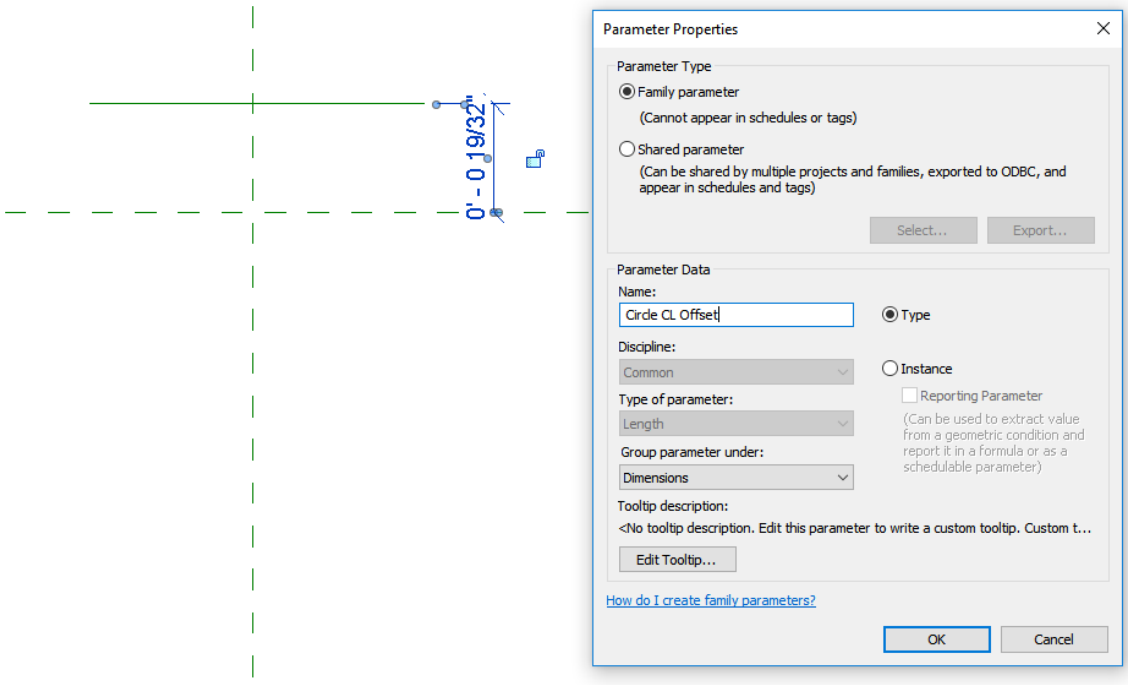
5. In some cases, it may be helpful to add **reference lines** to an annotation symbol. These can be used to help control the placement of detail lines related to the geometry in the symbol. For example, add a reference line just above the intersection of the reference planes as shown:



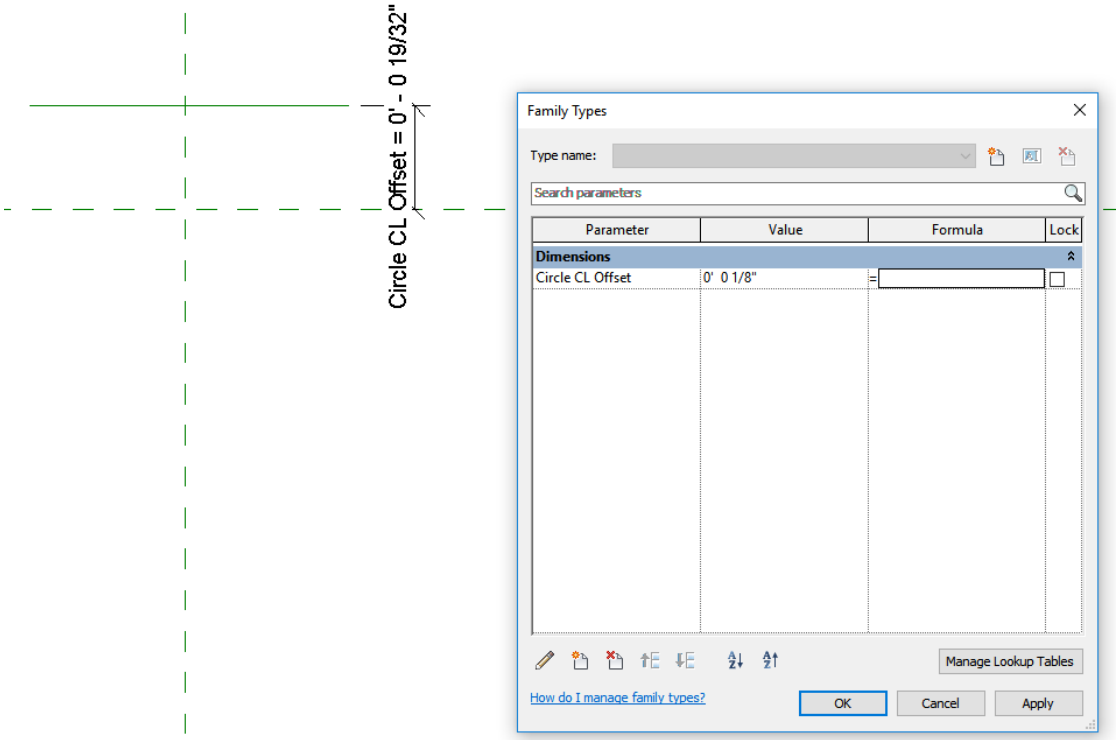
6. Once the line is placed, you can use the **intersection** of the reference line and reference plane to set the **center of the circle annotation** you're going to add, as well as change a temporary dimension to a permanent dimension with a label. This will help you control the offset distance from the insertion point to the center of the circle:



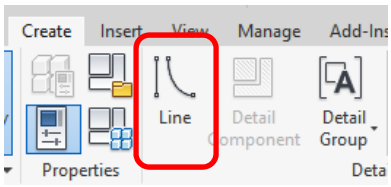
7. Select the **grip** to make the dimension **permanent**, and then use the **label** tool to add a family parameter. This value will control the offset value:



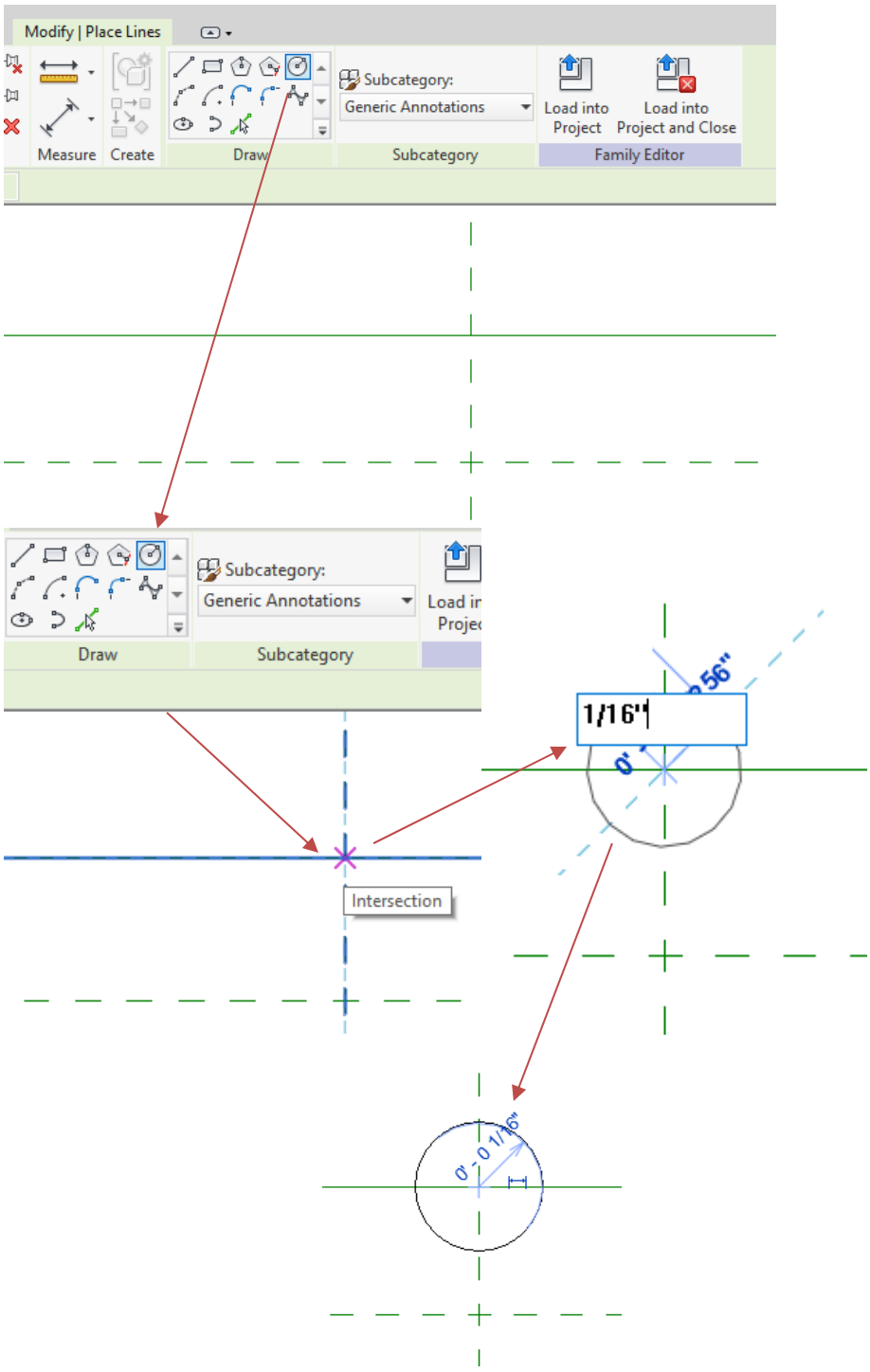
8. From the **Family Types** properties tool, set the default value to **1/8**":



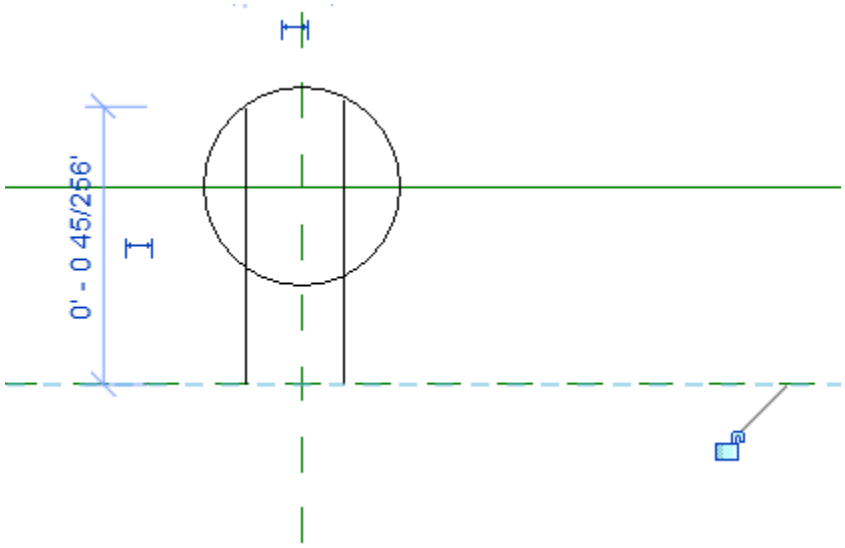
9. Click **OK** to complete the change and exit the Family types dialog. Next, add the circle using the **Line** tool on the **Create** tab:



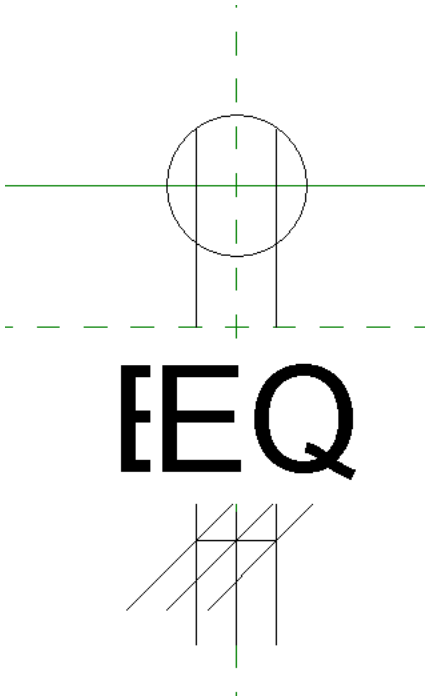
10. Select the **circle** option, and place a circle that has a **1/16”** radius:



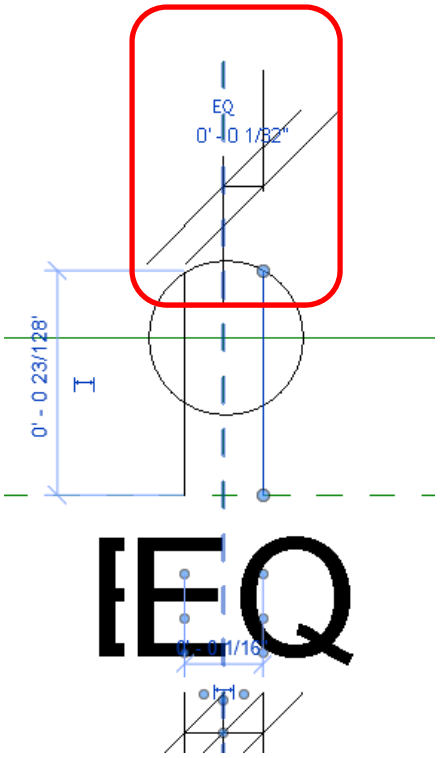
11. Once the circle is created, you can add a **label** to control the radius of the circle as needed. Leave it assigned to the default value, and return to the **lines** tool. Use the individual line tool to add **two lines** that represent the legs of the receptacle. You may notice a **lock** symbol appear after placing the line – this indicates that the end of the line is constrained to the reference plane:



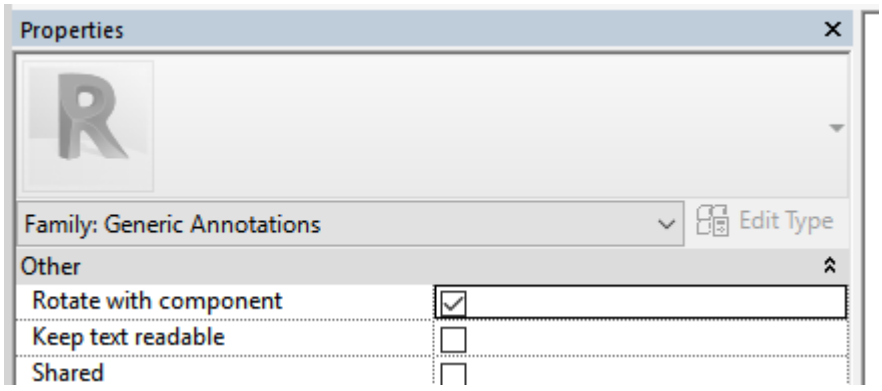
12. At this point, you can use **dimensions** to set the offset of the lines from the center left/right reference plane, but you can also use the **EQ** dimension feature to set the spacing to be **equidistant**:



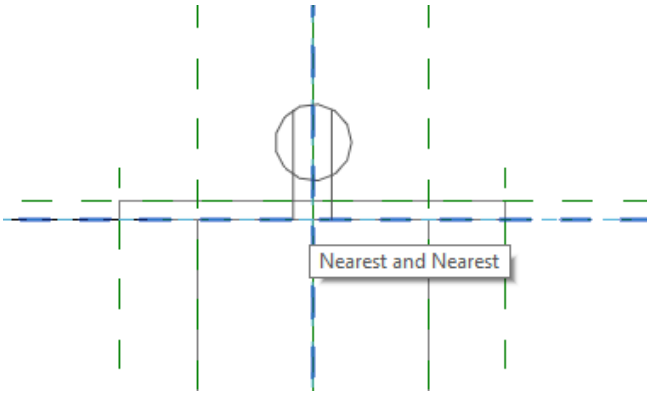
13. You can also use a dimension to set the width of the lines:



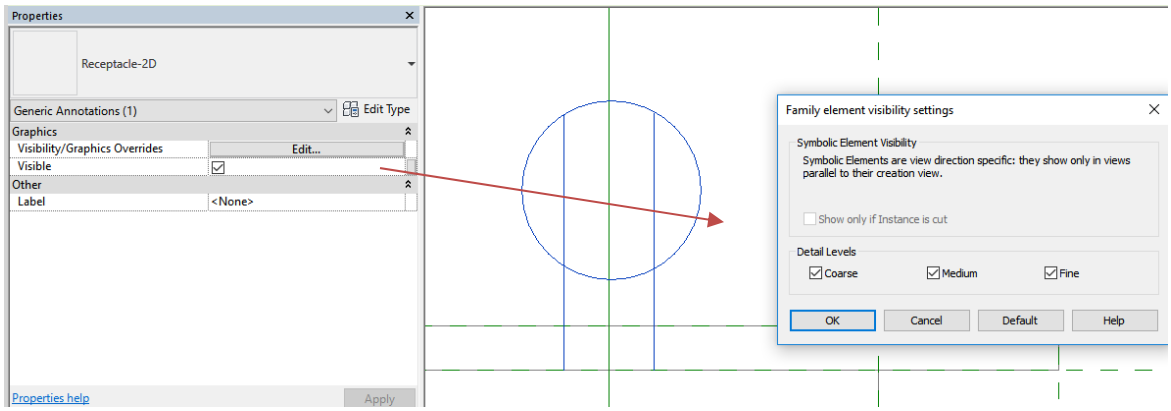
14. To make sure the symbol rotates with the family, review the properties palette when nothing is selected. If you set this to rotate with component, the symbol will rotate when the family is rotated. If your symbol includes text, then setting the option to keep text readable forces text to be horizontal and vertical at only 0 and 90 degrees rotation. Leave this option deselected if you want the symbol to rotate a full 360 degrees with the family:



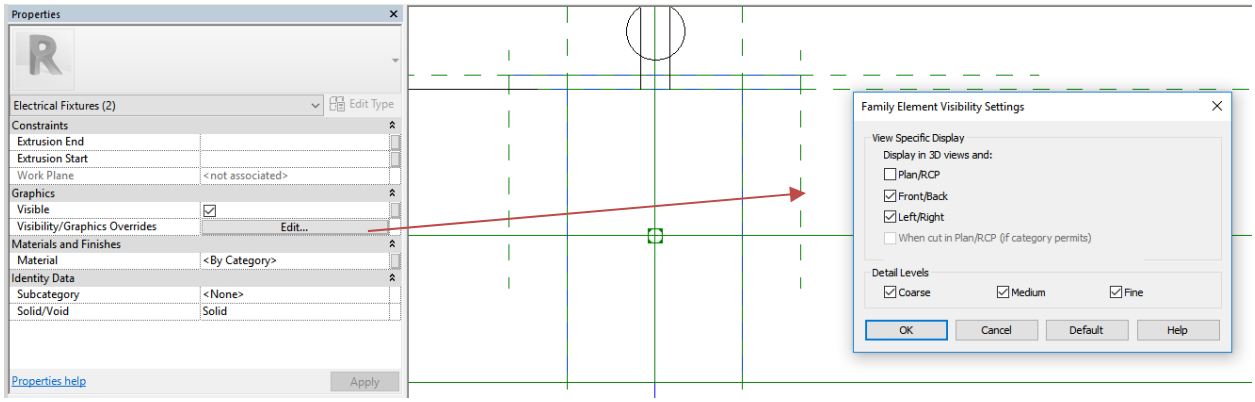
15. The point is to make the symbol as accurate as possible. **Save** the symbol, using a name that indicates how it is intended to be used, such as **Receptacle-2D**.rfa.
16. Once the symbol is defined, **load** it into the model family – if you are in a plan view, you can place the symbol immediately:



17. If you place the symbol at the **insertion point** that is defined by the **intersection** of the two reference planes, then the symbol is located and **anchored** to this point. The **scale factor** is applied **from** the **insertion point** of the family.
18. To make sure the symbol is all that appears in the view, check the **visibility** settings of the symbol first, and make sure it is visible at all detail levels:



19. Next, turn off the model representation in **plan** view by selecting the solids in the project, and changing their **visibility/graphic overrides**:



20. For the receptacle, the only views to show the **symbol** are **plan based**, and the **3D** model components will appear in the remaining **view directions**, at any detail level.

21. Once the symbol is placed into a project, insert an example in a plan view, and change

the **scale**:

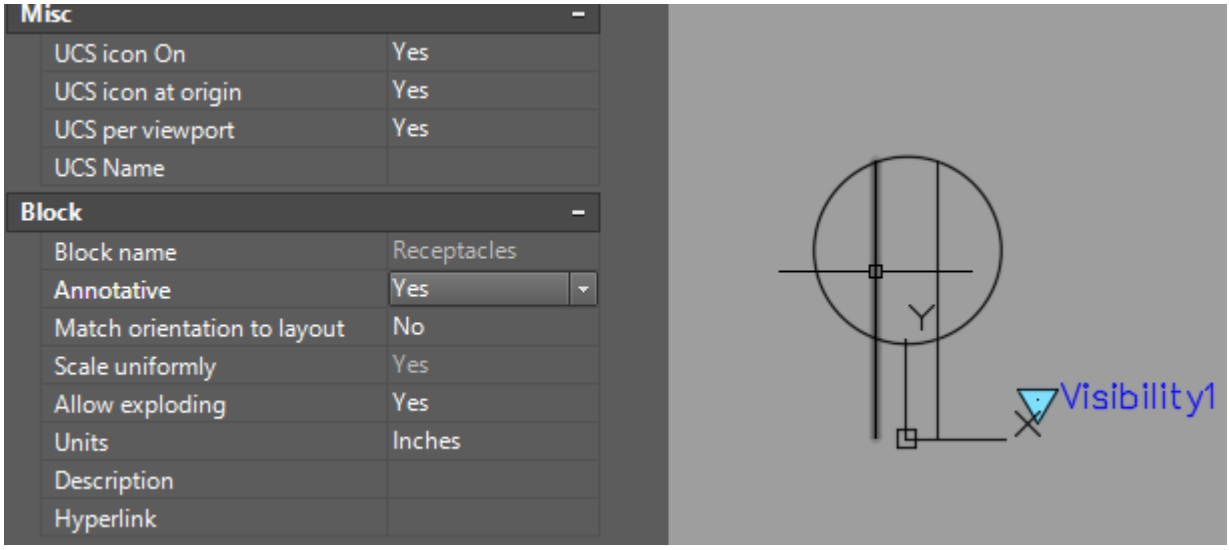
22. The symbol will resize itself, and regardless of the view scale, and be the same size in the crop region on the sheet (regardless of the scale).

AutoCAD – Creating an Annotative Block

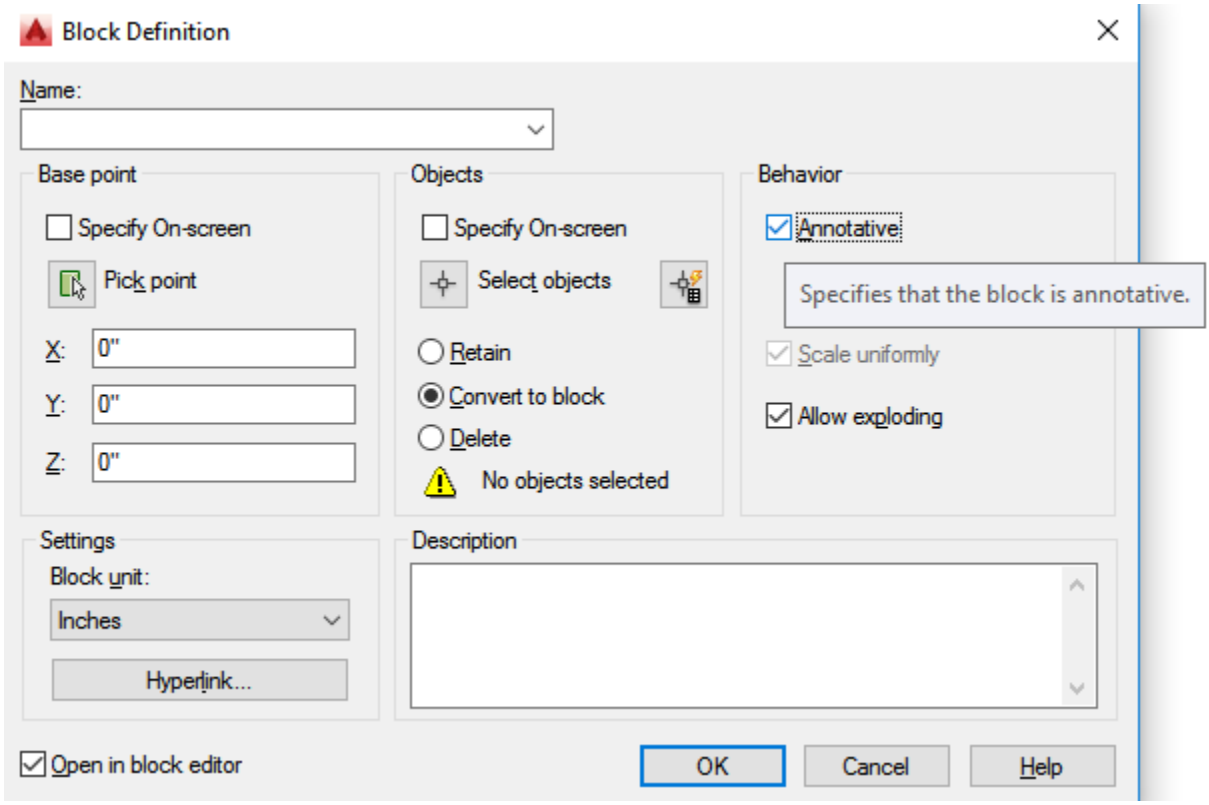
In AutoCAD, the types of annotative objects, such as a light switch, receptacle, or other part is too small to be clearly represented on sheet that is printed to a large scale (such as 1/8" = 1'-0"), is represented as an **annotative** block. Both annotative blocks and families will react to the scale assigned to a view.

In this example, we will skip the steps of making the block, but need to point out that the symbol should also have 0,0 defined as its insertion point. This way, the symbol correctly resizes itself relative to where it's placed in the drawing.

To make the original linework to be annotative, simply open the **Receptacles** block using the **Block Editor**. With nothing selected, check the **Properties** palette, **Design** tab. Set **Annotative** to **Yes**:



Changing the scale will adjust the size of the block, so make sure you draw the linework representing the block to match the desired size in a scaled viewport in paper space. You can use this feature to convert any AutoCAD block to annotative, as well as select that option when defining the block:



Important Note – do NOT apply annotative behavior to a block that is showing objects at ACTUAL Size!

Revit Family Enhancements versus Dynamic Block Actions and Parameters

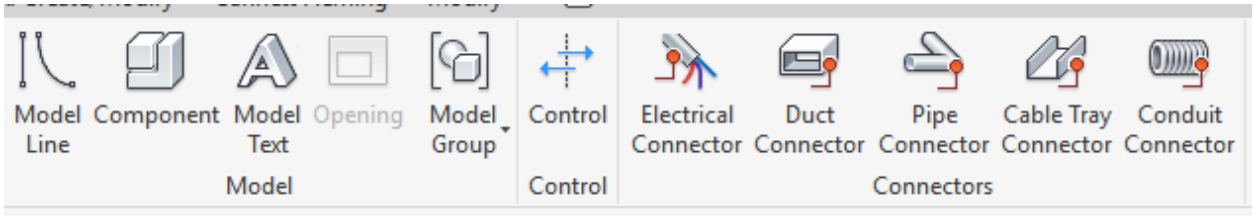
Concepts – Parametric and Action Behavior

In both AutoCAD and Revit, there is going to be a need to make minor modifications to annotations and symbology once they are placed. Any time you can reduce the number of mouse clicks related to a task, the numbers will add up to some significant savings. In this case, you need to understand how some basic controls and actions can make editing annotation content easier.

Revit – Adding a Control for Flip Action

In Revit, any family can include additional control items to help with the placement of objects. This includes both system families, such as walls, ducts and pipes, but also include loadable families. All Revit families that cover physical representations can include one of these grips tools, but any nested annotation included in the family will also respond to the action.

1. In the receptacle family, on the **Create** tab, select the **control** tool:



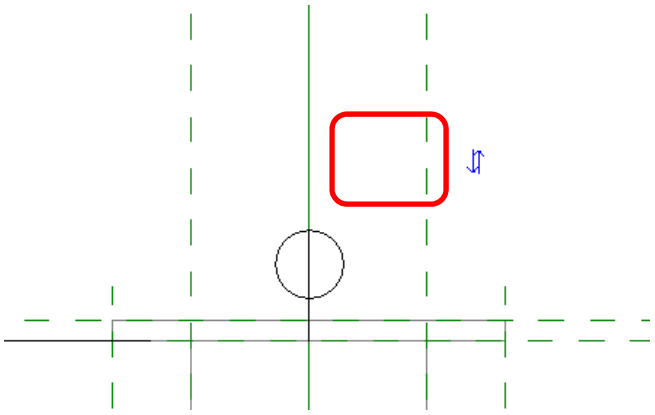
2. There are **four types** of control items:

a) Single Vertical
b) Double Vertical
c) Single Horizontal
d) Double Horizontal

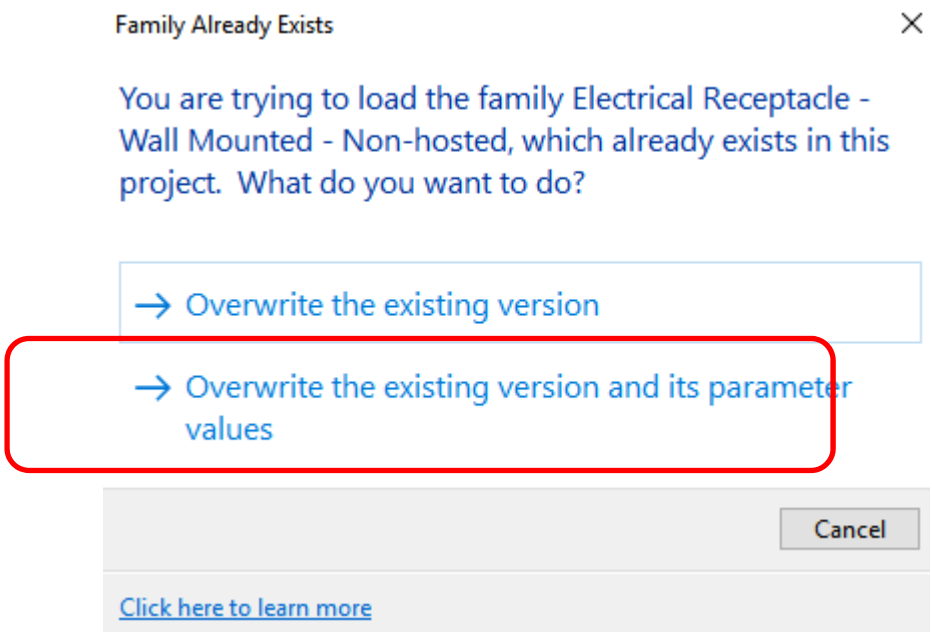
Each of these perform the same task, with single and double referring to the number of arrows you see when the family is selected. The vertical option will flip the family along its center – front/back axis, while the horizontal will flip the family along the center – left/right axis.

To see how a control behaves, begin by adding a double vertical to the current family:

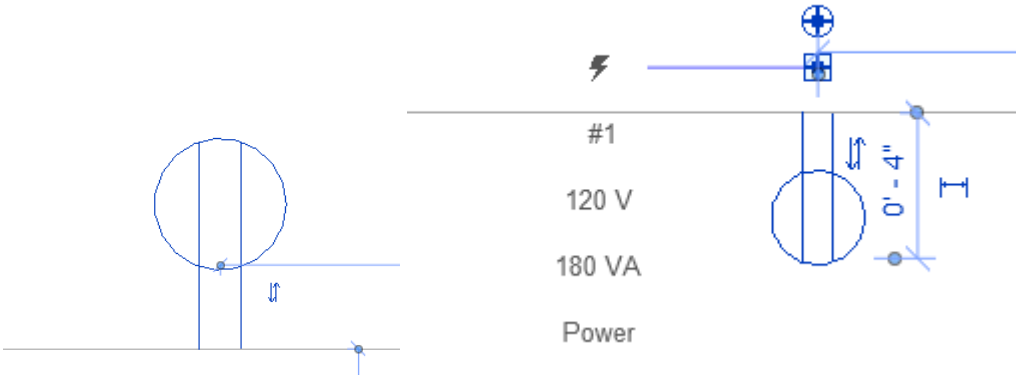
3. Select a **point** close to the receptacle symbol. The control will appear as **two arrows**:



4. When finished, save the changes, and load the family into a project. If you already have the family in a project, make sure you select the option to overload with parameters, since this is adding a new parametric action:



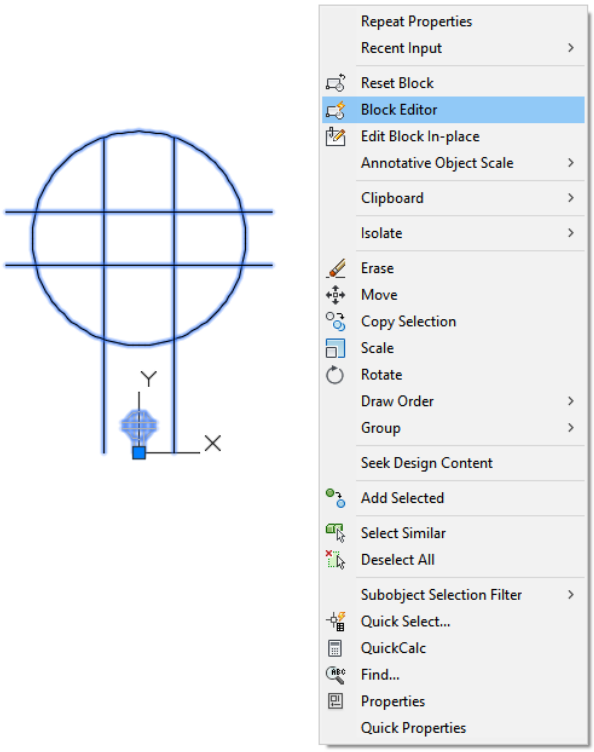
5. When the family is placed, select it and locate the flip grip. Click it and observer the behavior:



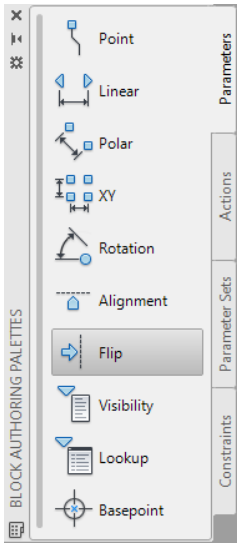
6. The entire family – including the receptacle box – is flipped in the opposite direction. At any time, you can select the grip again, and the device will flip back into its original position.

AutoCAD – Adding a Flip Parameter and Action

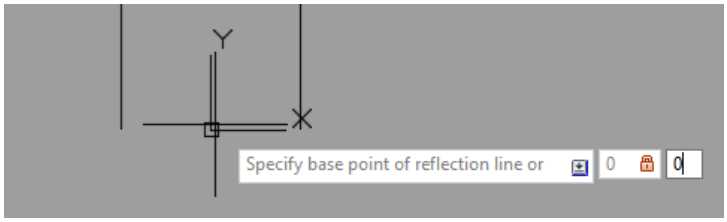
In AutoCAD, the same tool is applied as a dynamic block action to a parameter. In the **Receptacle – Wall Mounted.dwg**, select the **Quadraplex Receptacle** block. Note that the block is already set to be annotative – right click and click **Block Editor**:



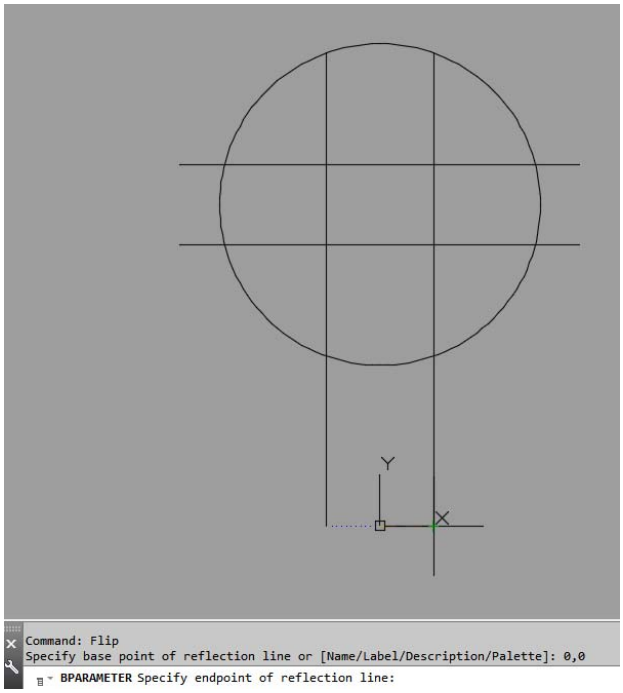
1. When the block opens, select the **Flip** parameter from the **Block Authoring** palette, **Parameters** tab:



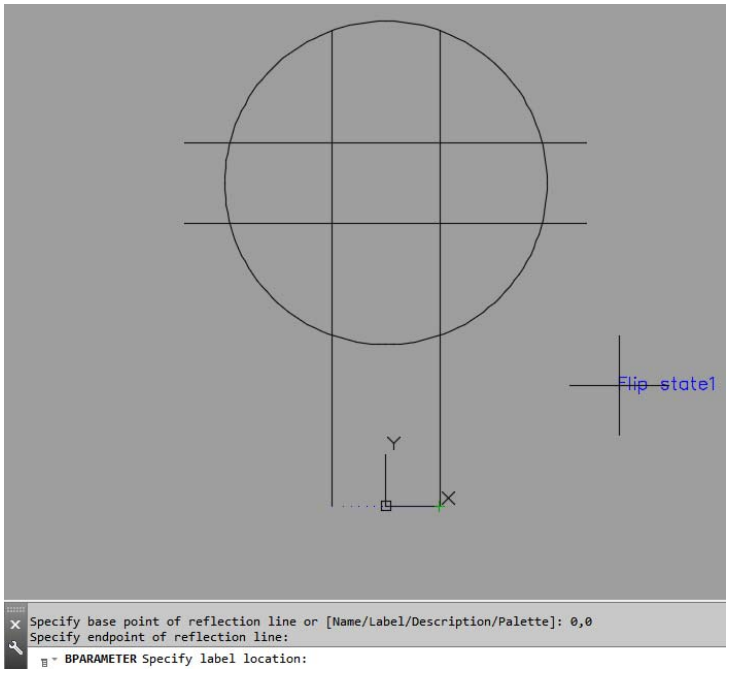
2. For the **base point**, select the **insertion point** of the receptacle, or **0,0**:



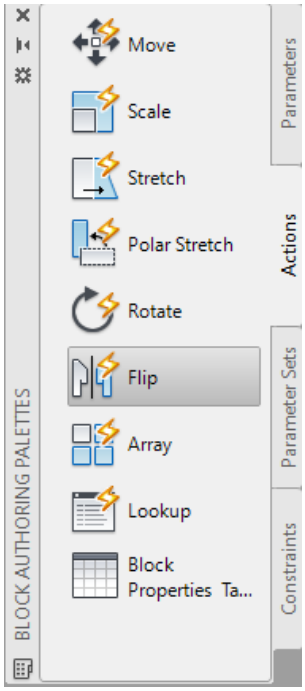
3. Next, you will be prompted to specify the **end point of the reflection line** of the parameter – select a **point** at the end of the fixture:



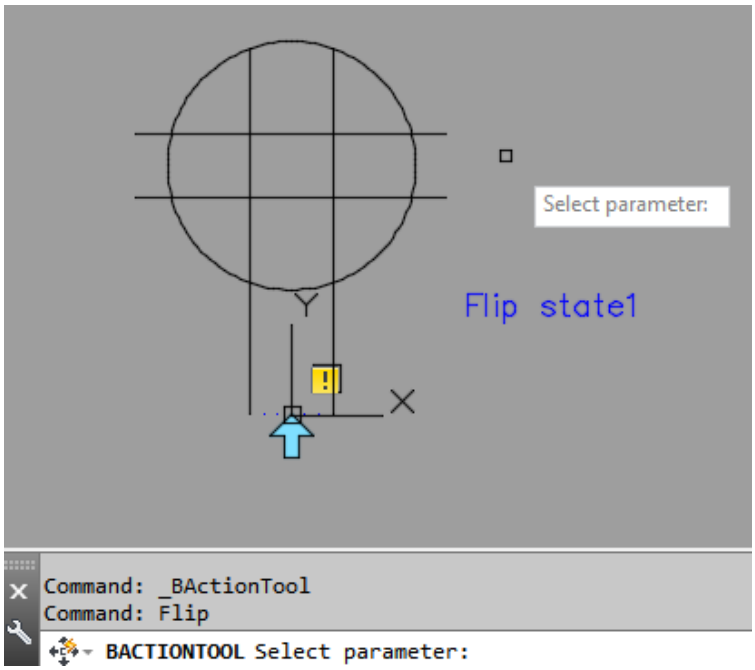
4. When prompted for the default **label location**, select a point to the right of the symbol, in a similar location to the Revit flip control:



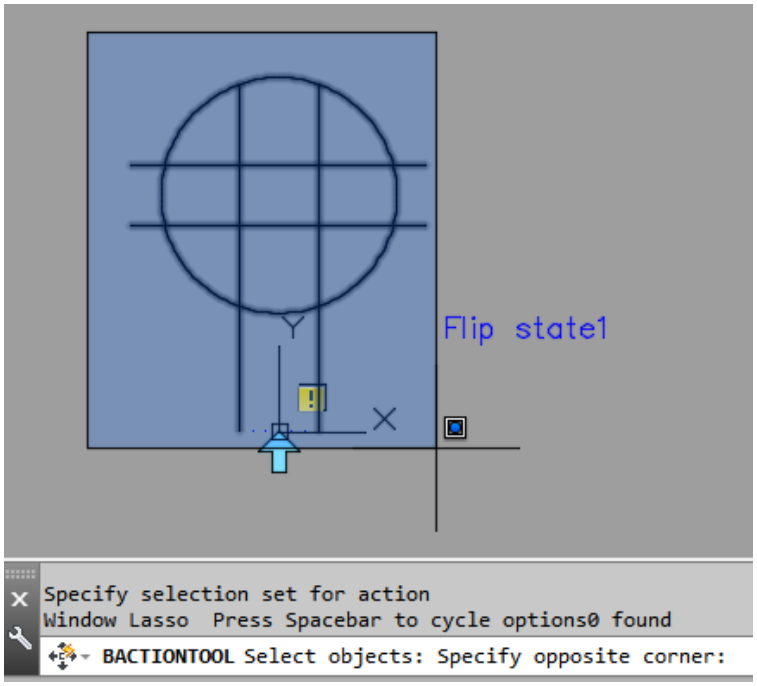
5. After the parameter has been added, switch to the **actions** tab, and select the **Flip** action.



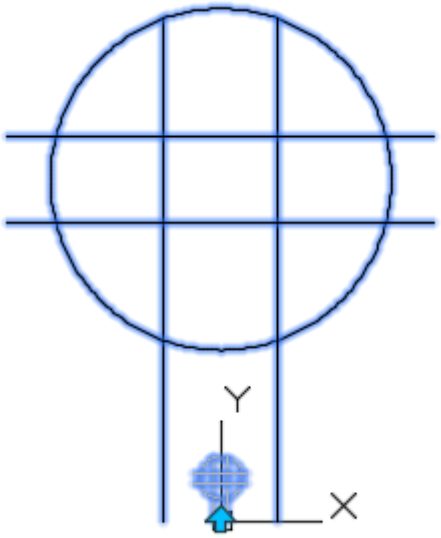
6. Next, you will be prompted to select the **parameter**. Choose the **arrow** that appears at **0,0** (note: the **yellow warning** indicates that **no action** is associated with the grip, so you're adding it now)



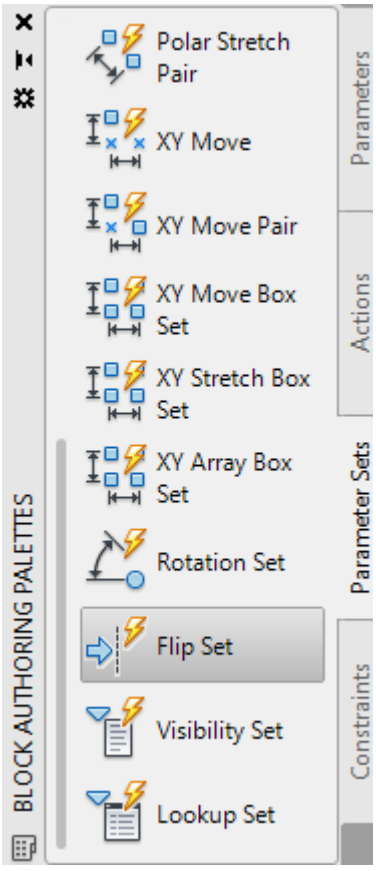
7. Next, you are prompted to **select** the objects that are to be flipped:



8. When the objects are selected, press **ENTER** to complete the command. **Close** the Block Editor, **saving** the changes when prompted. The grip will appear at the insertion point of the symbol:



9. Flip the block to see the results of the tools. The next time you try this, you can save a couple of steps by using the **flip set parameter set**, which is found on the **Parameter Sets** tab:



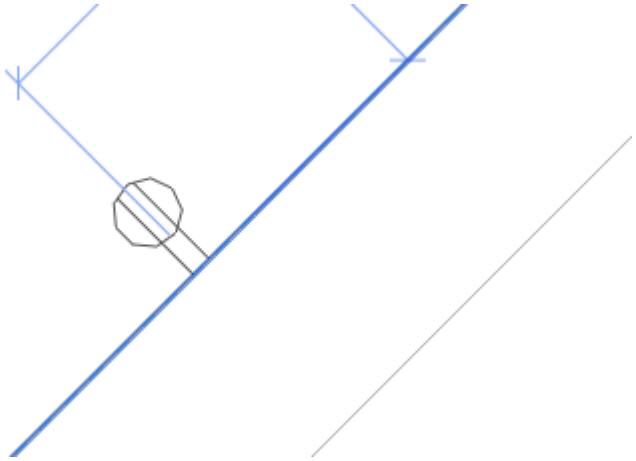
This combines all the tools into one set, making it easier to complete the task.

Concept – Editing AutoCAD Blocks to use Revit Workplane-Based Family Behavior

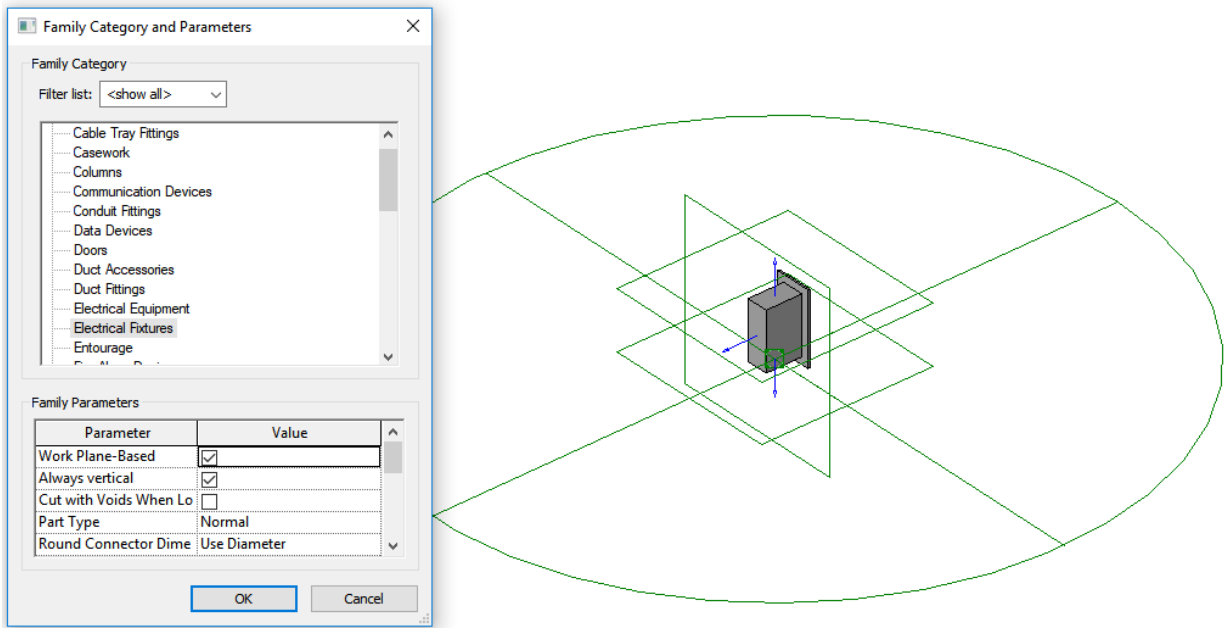
Most drawings do not contain straight lines, and in some cases, you need to align a block with another object in the drawing. Dynamic blocks include an alignment feature that allows you to follow linework in a drawing where you are placing the block. Revit includes the same alignment feature with non-hosted and hosted blocks, so using an alignment will help you place symbols accurately.

Revit – Making a Family Workplane Based

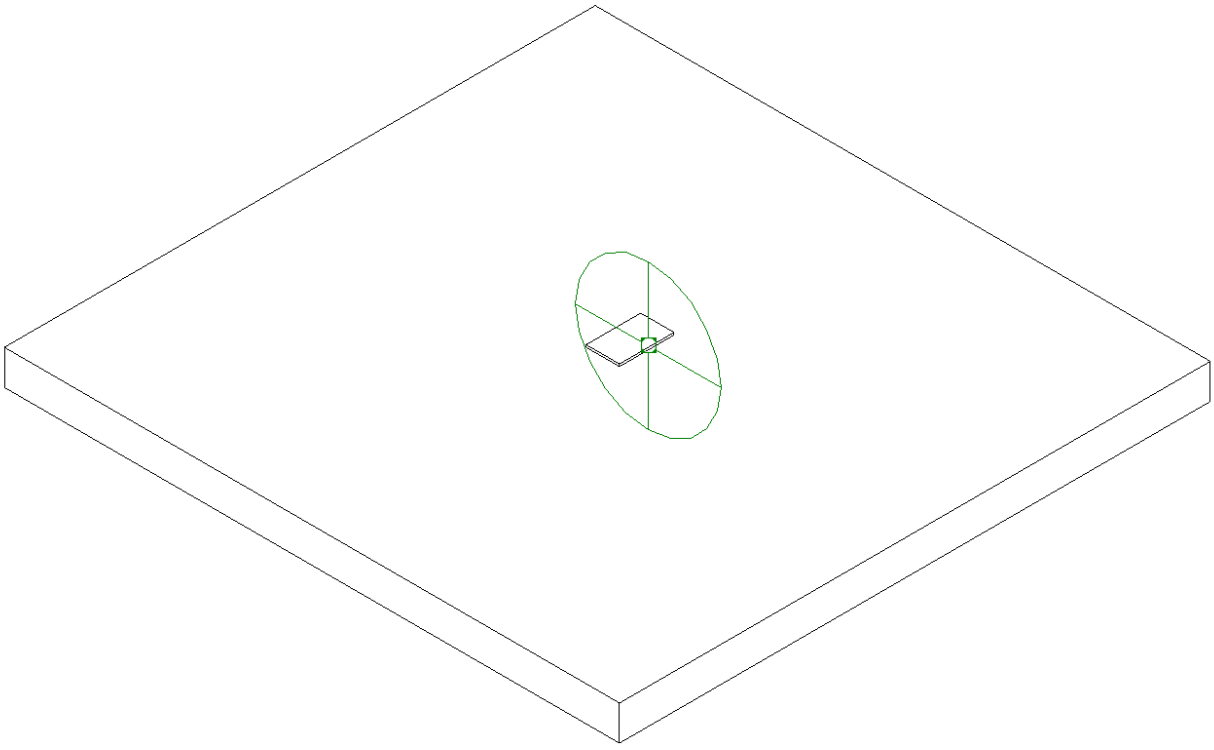
There are two types of loadable component families in Revit – **hosted** and **non-hosted**. A non-hosted component family, like our receptacle, can be placed anywhere in a project. But it doesn't need an extra tool to make it align with a wall, it just needs an extra step. When you drag your symbol along another object such as a wall, Revit automatically looks at the **surface** being used as a placement tool. If you tap the **TAB** key on the keyboard while the alignment is highlighted, Revit will **rotate** the fixture to match the **angle** of the surface:



You can edit the family and set it to search for a **workplane** when placed, but the item still should be rotated, and is not constrained to the wall. Workplane settings are better for other faces that may be sloped, or defined by a reference plane in a model:

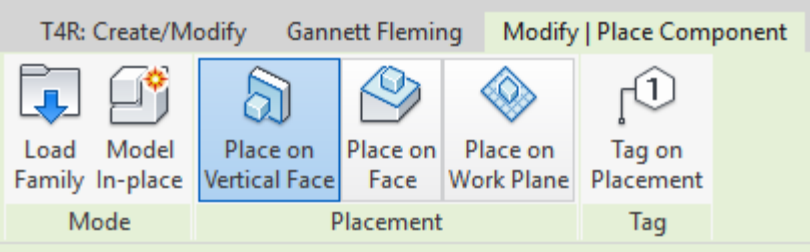


An even better solution is to use a **hosted** version of the family. The **Duplex Receptacle** family is an example of the hosted family. If you edit the family, you will see the face that is used to "host" the elements:

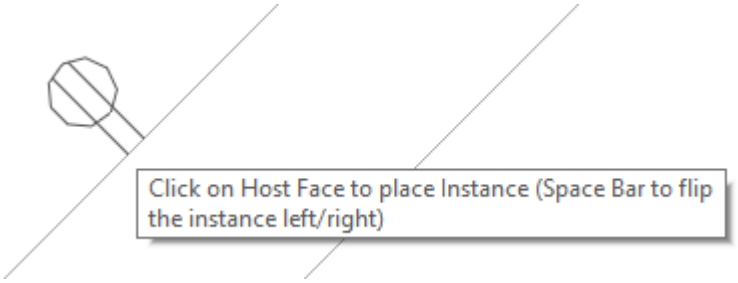


It's critical to understand the host in this case. **Stick with Face based hosts for MEP fixtures**, such as **receptacles**, **light switches** and **plumbing fixtures**. Other types of hosts, such as **wall**, **ceiling** or **floor**, cannot recognize these objects when they are in a linked model – the actual host must reside **in the same file** for these types.

When placing the hosted element, you can select **vertical face** to represent the wall surface from the **Modify | Place Component** tab:



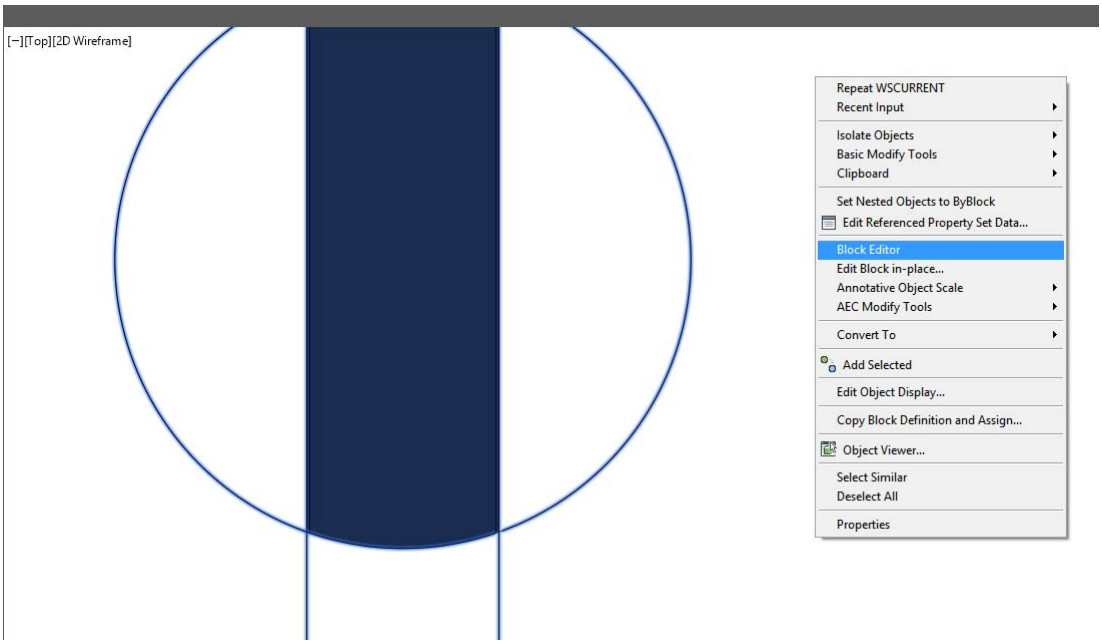
This forces the symbol to **align** to the wall surface without requiring rotation or TAB selections:



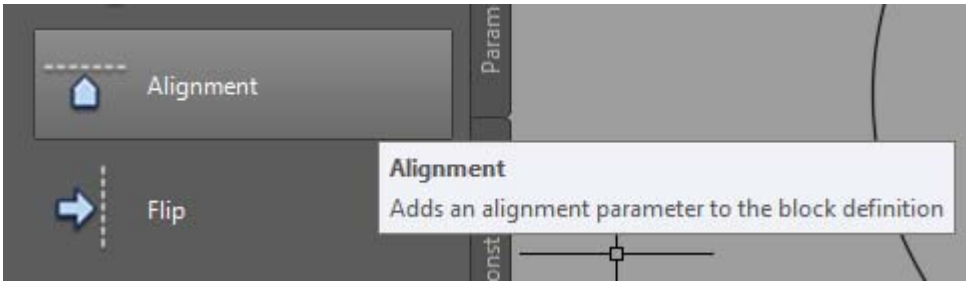
The item is now placed – if the wall moves, the symbol will move.

AutoCAD – Adding an Alignment to a Block

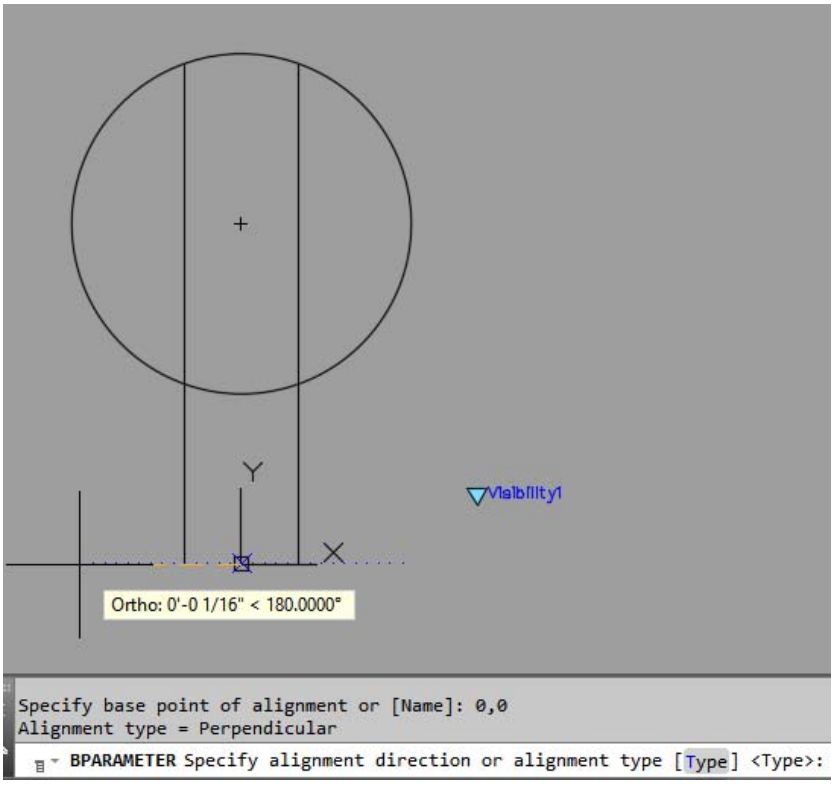
1. From the drawing, **Receptacle - Wall Mounted-Constrained.dwg** – select the receptacles block, and then right click. Select **Block Editor** from the list to open the block.



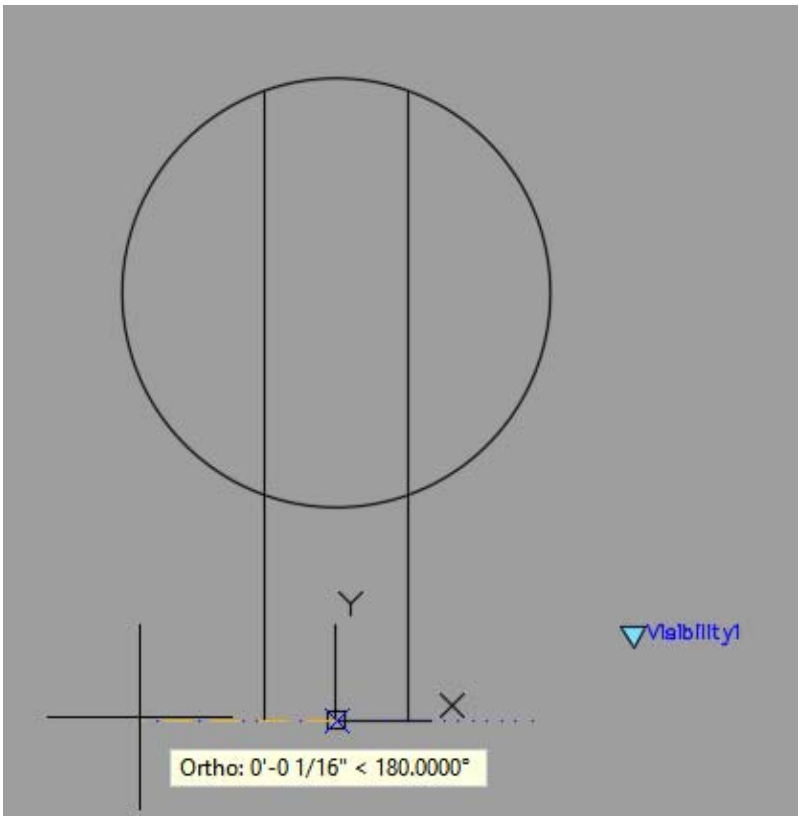
2. From the block editor, select the **Alignment** tool from the **Block Authoring** palette, **Parameters** tab:



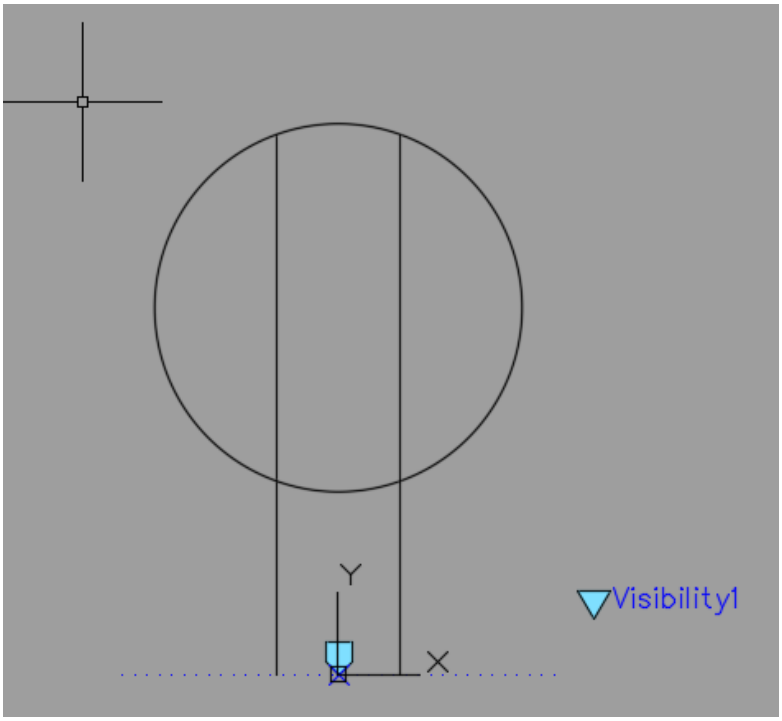
3. For the **base point**, type **0,0** on the command line and press **ENTER**. When the next prompt appears, type **T** for **Type** and press **ENTER** to set the alignment type:



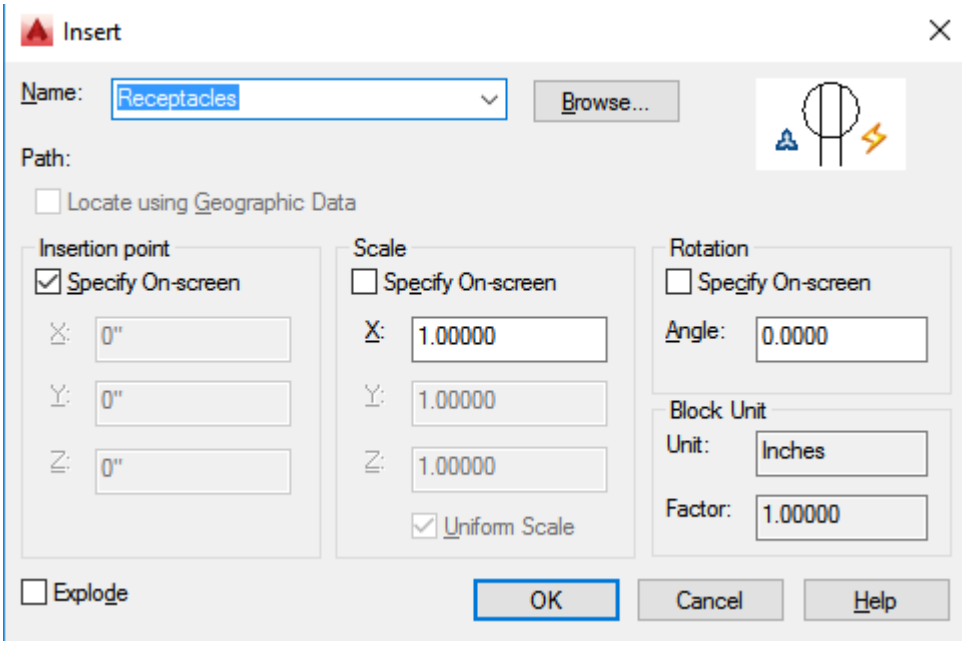
4. There are two options – **Perpendicular** is applied to any linework that allow an object to user a perpendicular snap, while **tangent** is applied to arc or curve type segments. Select **Perpendicular** from the options and press **ENTER**.
5. Next, make sure you have the **ORTHO** option enabled, and then drag your mouse to the left to place the **grip** for the alignment:



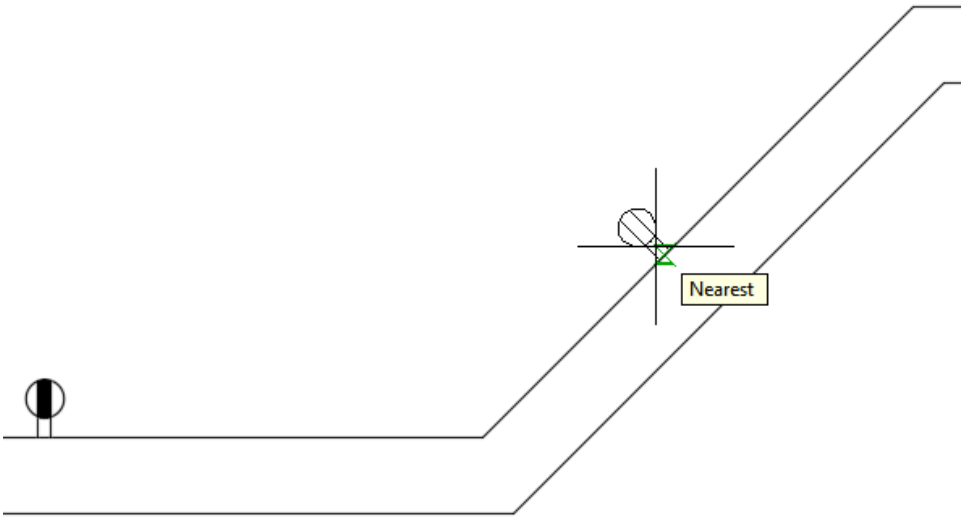
6. Once the alignment is placed, you can see the **parameter** assigned to the block. **Close** the block editor, saving the changes.



7. Next, open the drawing **Walls.dwg**. From the command line, select **Insert**, and select the block **Receptacles** (this drawing includes the constrained block you edited in the previous steps). Make sure the option to **Specify insertion point onscreen** is selected, and click **OK** to continue:



8. Move your mouse along the lines representing the walls:



You can use the nearest snap to help with placement, but you will not need to rotate the symbol – it will align with the geometry as it is placed.

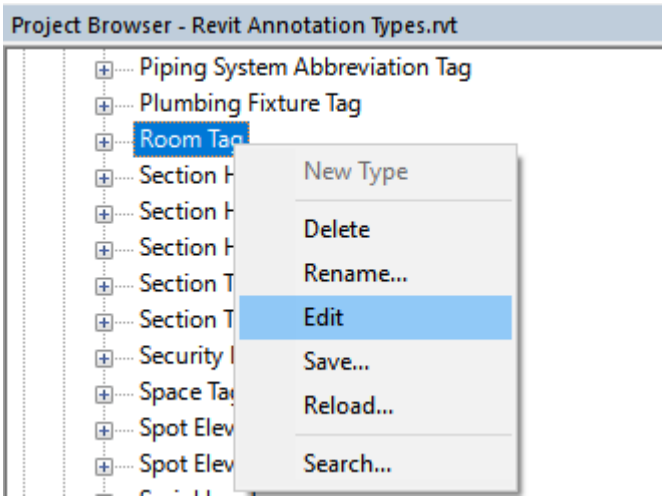
Concepts – Editing Symbol Linework

Some firms use blocks as tags to label items in a drawing, such as room names, equipment and more. A customary practice is to add a border to specific tags to emphasis the data. Both Revit and AutoCAD have tools that allow you to change the shape of this symbol.

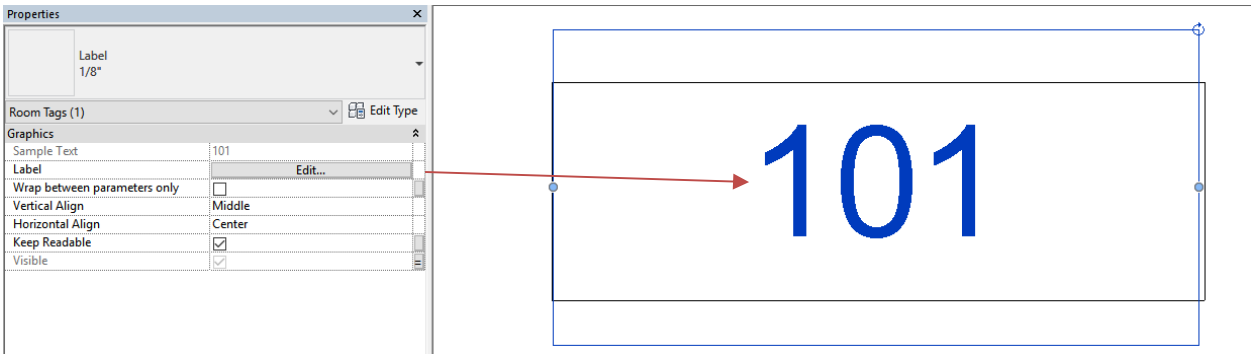
Revit – Adding Text Boundaries

To begin, let's examine how a room tag is defined, and then learn how to add grips to the box that allow it to be resize after it's placed.

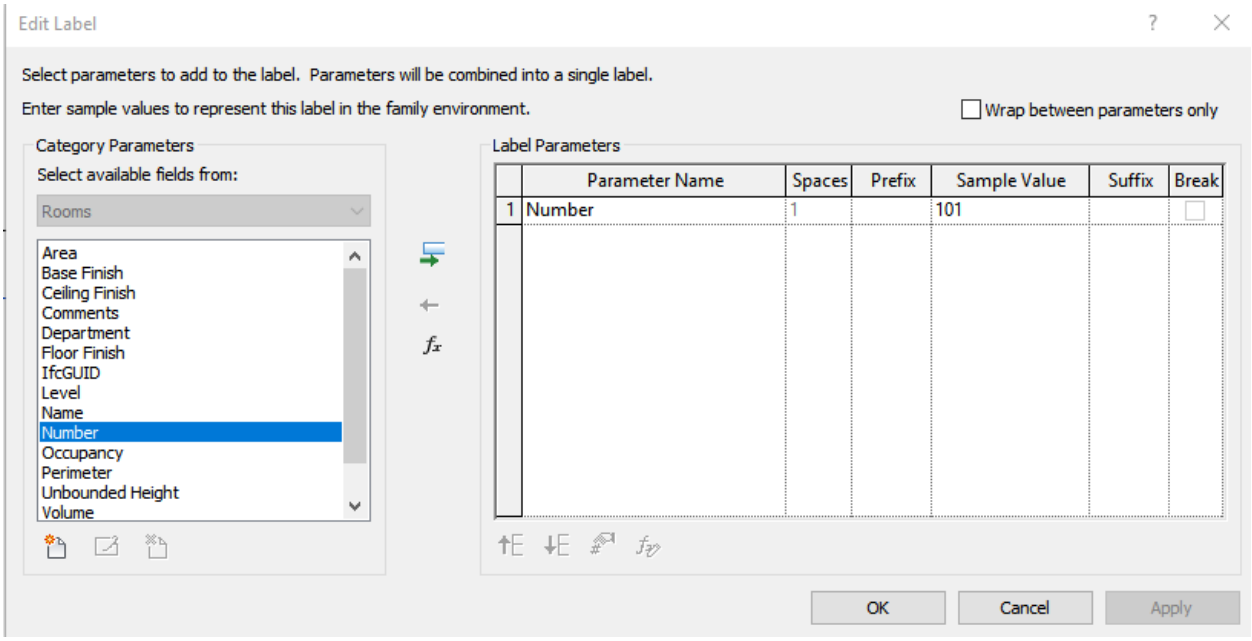
1. In the Revit Annotation Types project, browse under Annotation Families to locate the Room tag. Right click and choose Edit to open the family:



2. First, select the **number** inside of the box. This piece of text is a **label**, and can derive its information from the properties of a Revit room element. From the Properties palette, select **Edit** tool for the Label:



3. When the dialog appears, you can choose from the available category parameters that are built in to Revit, or you can use the **New** tool to create a new parameter, based on a shared parameter. Under **Label Parameters**, note the **name**, **spaces**, **prefix**, **sample value** and **suffix**. Since the value is a single item, the test string can be as long as needed.



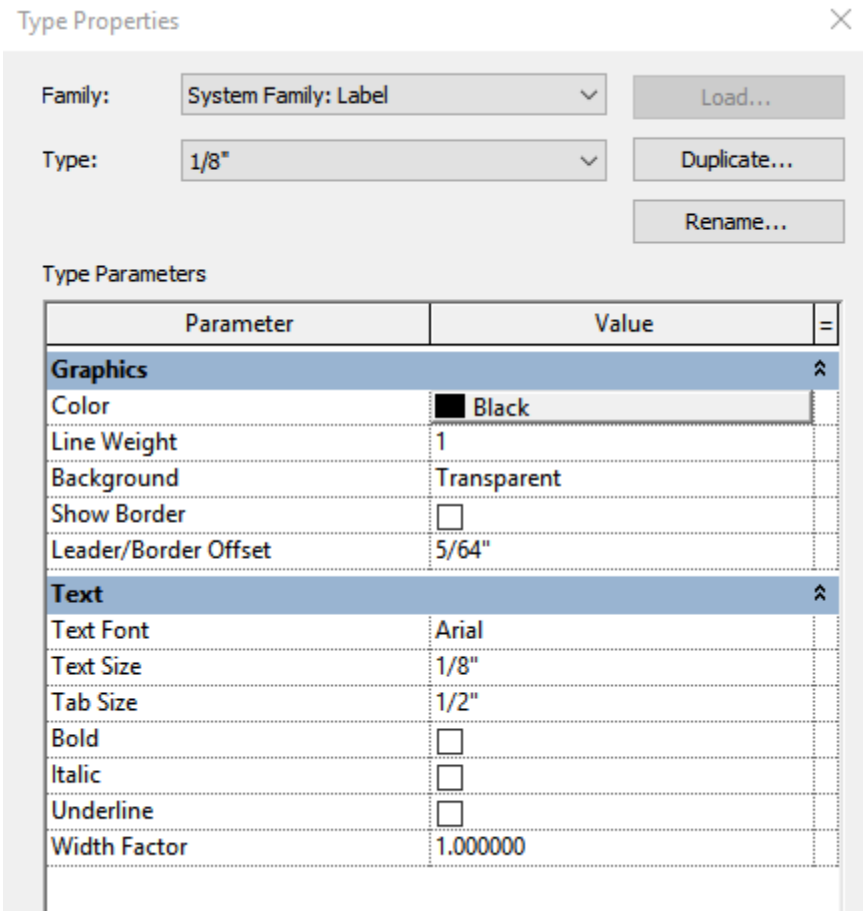
4. Click **OK** to exit from this dialog. While the text is still selected, notice the **boundary** around the text:

5. This can set the **width** for the **text object** – if the value exceeds this space, including spaces and wild card characters, the text could be forced to wrap to a second line. To expand the property of the label to allow longer field names, select a **grip** at the side of the box:

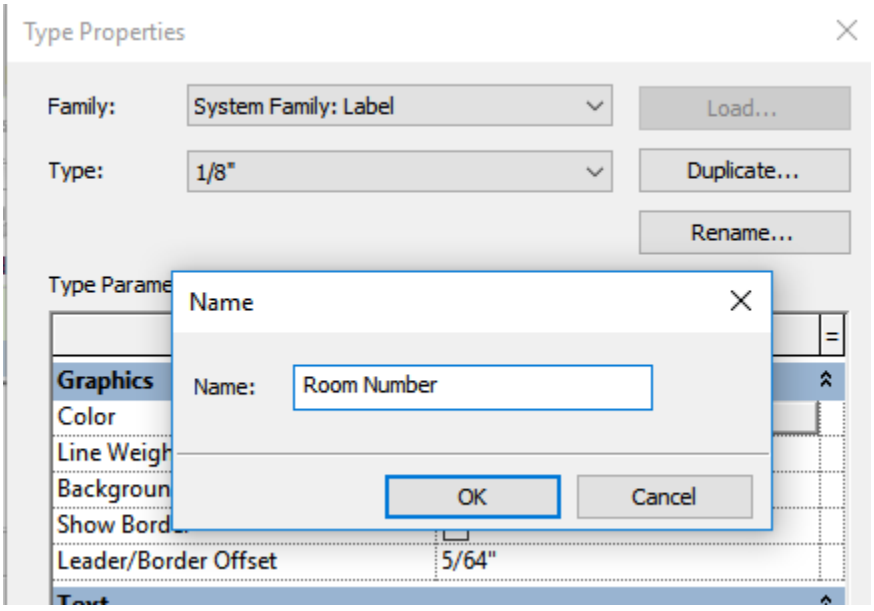
6. The circle grip allows you to **stretch** the overall size of the text. Since the justification is set to **center** and **middle**, the size grows outward on each end. Stretch it out larger than the linework used to define the box.

   To make the box stretch an equal amount on both sides of the text, you have two options. You can use **dimensions** and associate them with the vertical lines in the box using a **type parameter**, or you can edit the **text type** used for the label to include the **border**. With the last option, as the text string is expanded, the box will grow automatically with the text.

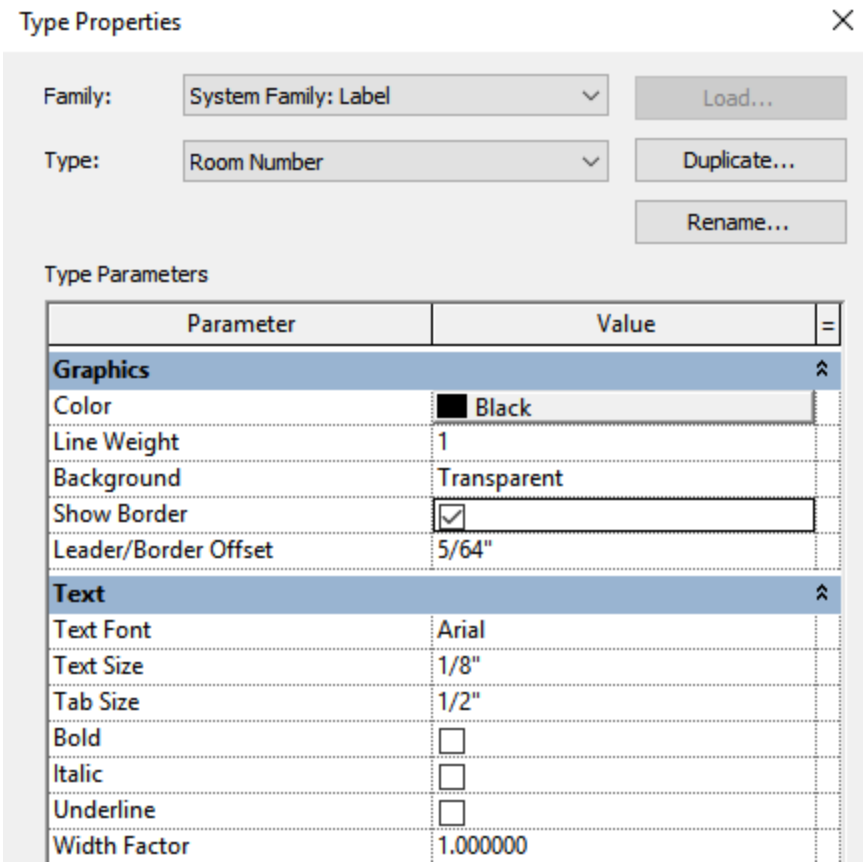
7. Select the **label**, and then select **Edit Type**:



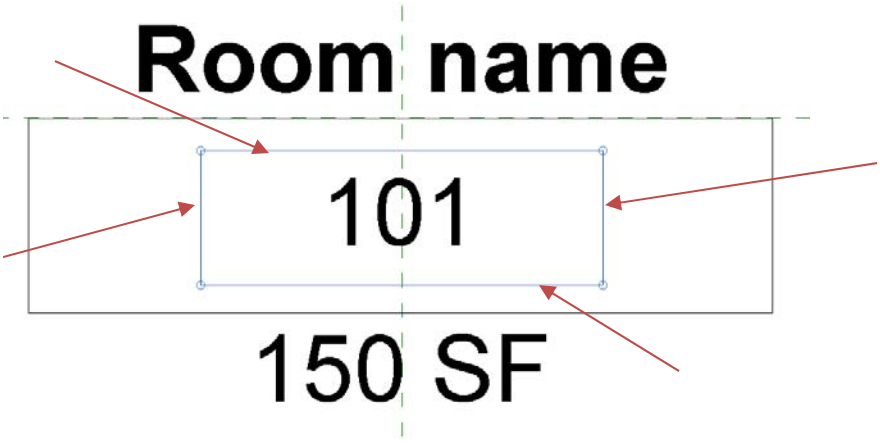
8. Select **Duplicate**, and then rename the label to **Room Number**:



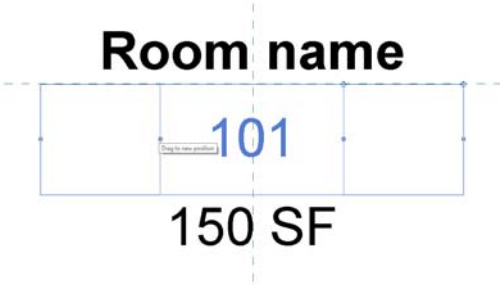
9. Click **OK**, In the **Graphics** section, select the **Show Border** option:



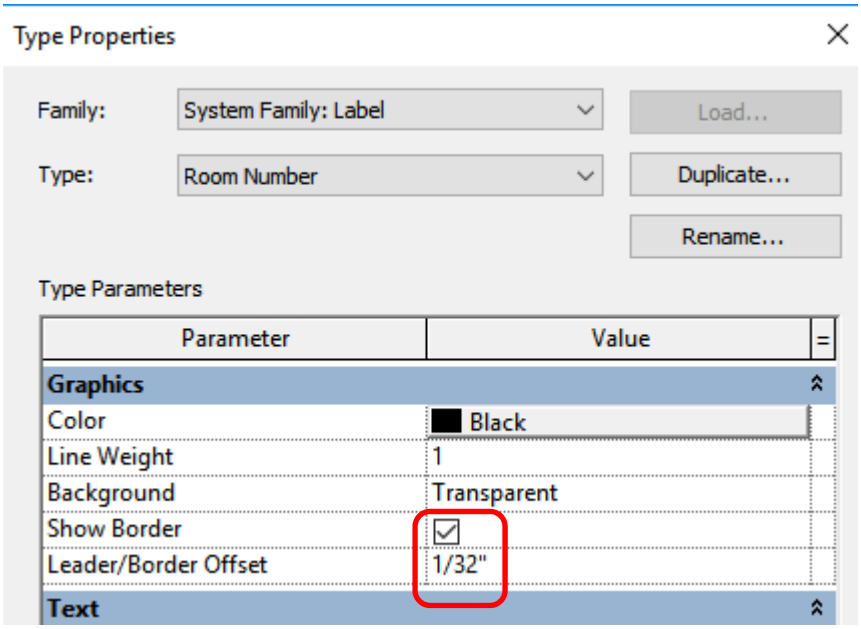
10. Leave all other options as is, and click **OK**. In the view, select the original **lines** used to define the box, and **delete** them from the family:



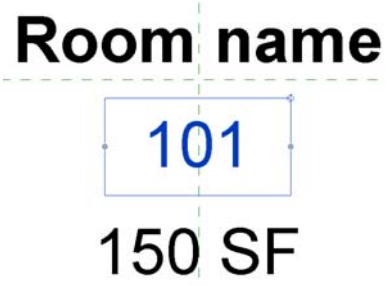
11. To change the **offset** around the label again, select it, and use the grips to change the width:



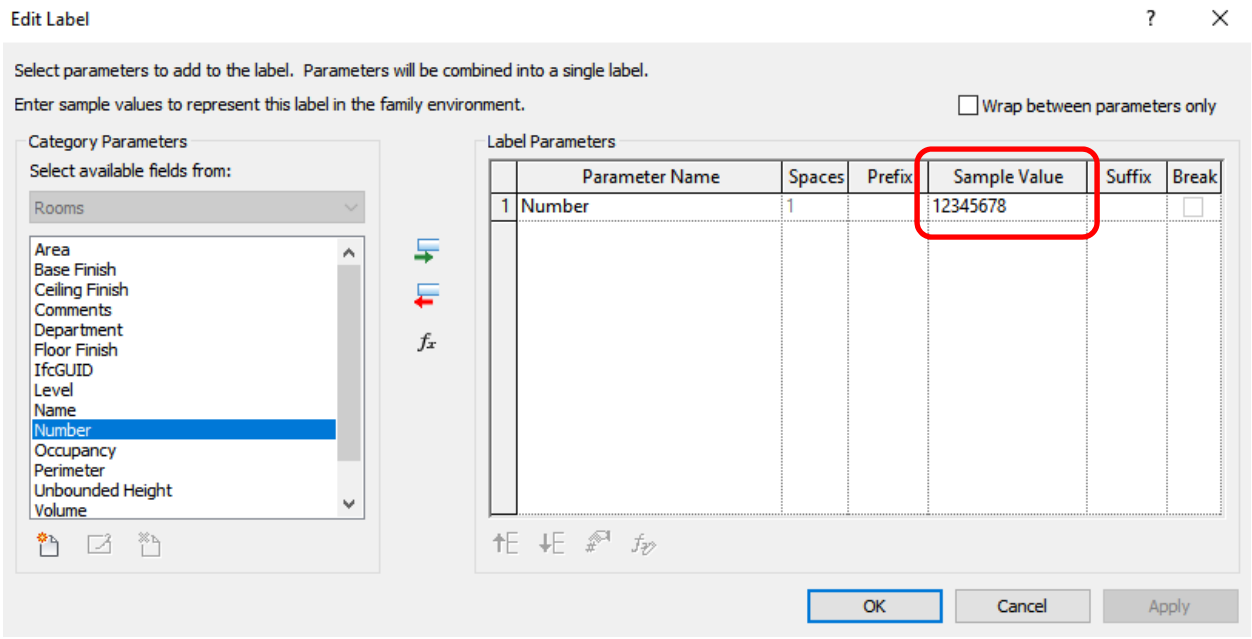
12. After resizing the width, edit the **label type** again. This time, change the **Leader/Border Offset** to **1/32”**, and click **OK** to see the changes:



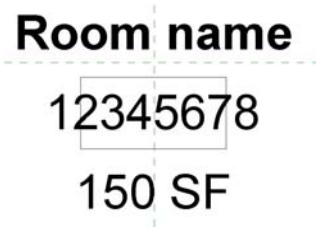
13. The border is now closer to the top and bottom of the text:



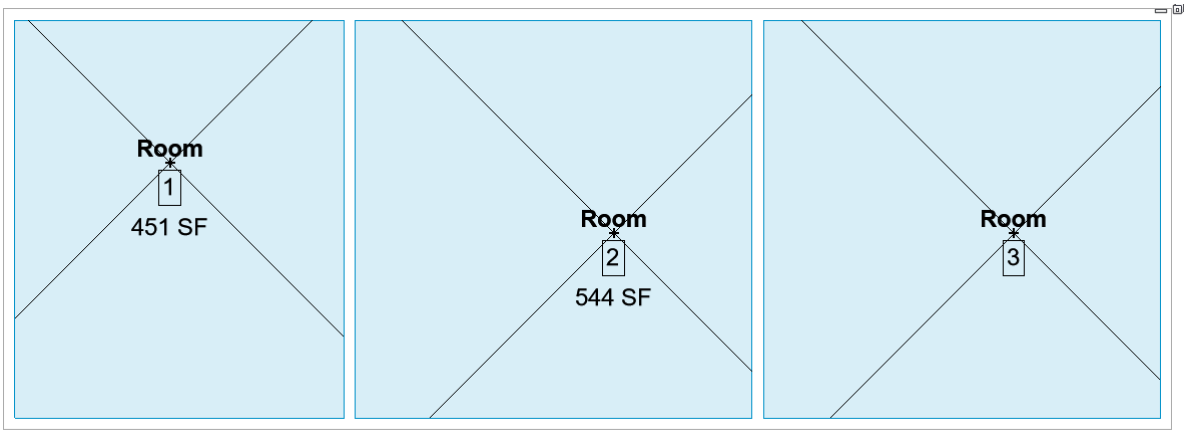
14. Select the **Edit Label** tool to run the final test. Change the **Sample Value** to a longer number, such as **12345678**:



15. Click **OK**, and review the changes. You may still need to stretch the box to allow for the **largest size** of text that will appear:



16. Once the parameters are assigned, **save** the family, and then load it into the project. Open the **Check Room Tags Here** floor plan, and note the changes to the box:



17. If you change the room number, the box will expand to cover the extra characters:

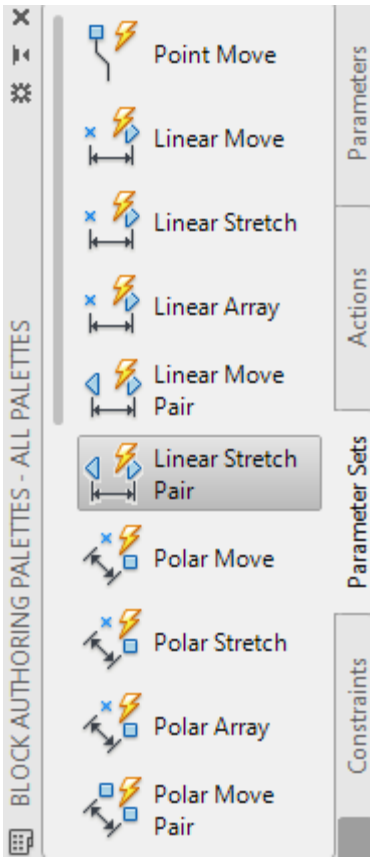
In this case, you are using a feature associated with a text type to control additional graphics, saving several CAD steps.

AutoCAD – Adding Linear Stretch Parameter to a Block

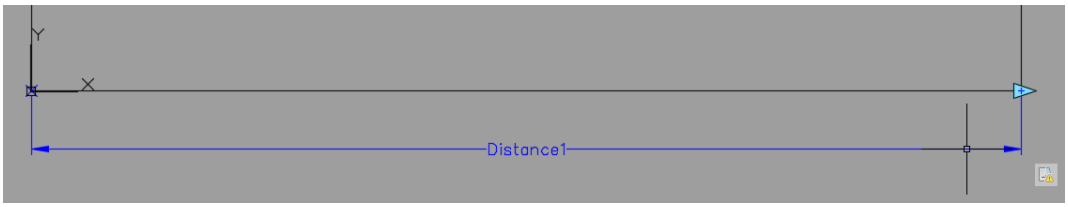
To get similar behavior in an AutoCAD dynamic block that is used as a tag, you can add a linear stretch parameter. This gives you the ability to resize a box using grips without having to edit a global type value.

1. In this case, you want to combine an action with a parameter. You can also use Parameter Sets, which combine an action and parameter as one action. From the current drawing, select the **Room Tag Sample** block as shown. Right click and select **Block Editor**:

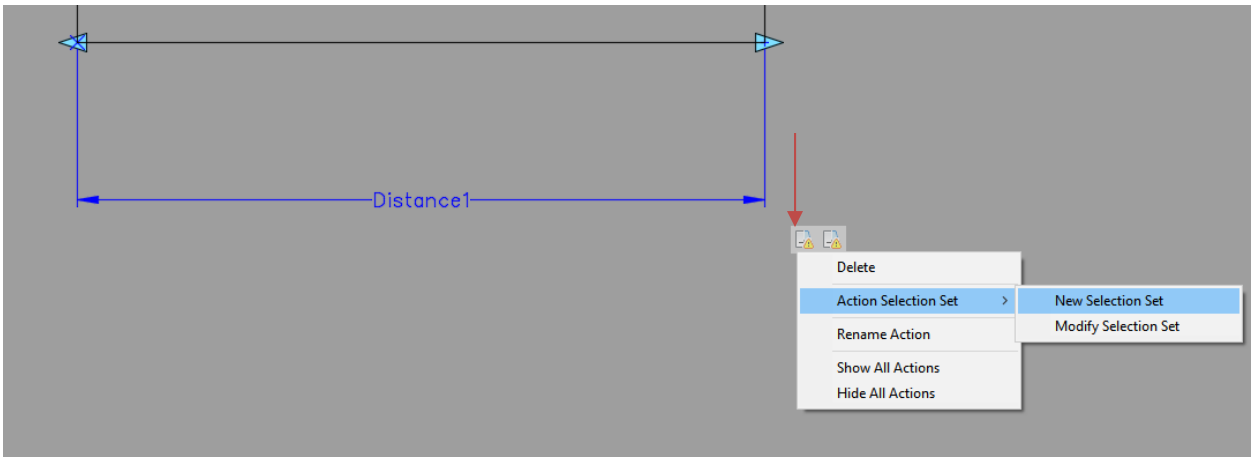
2. From the **Block Authoring** palette, **Parameters Sets** tab, click **Linear Stretch Pair**:



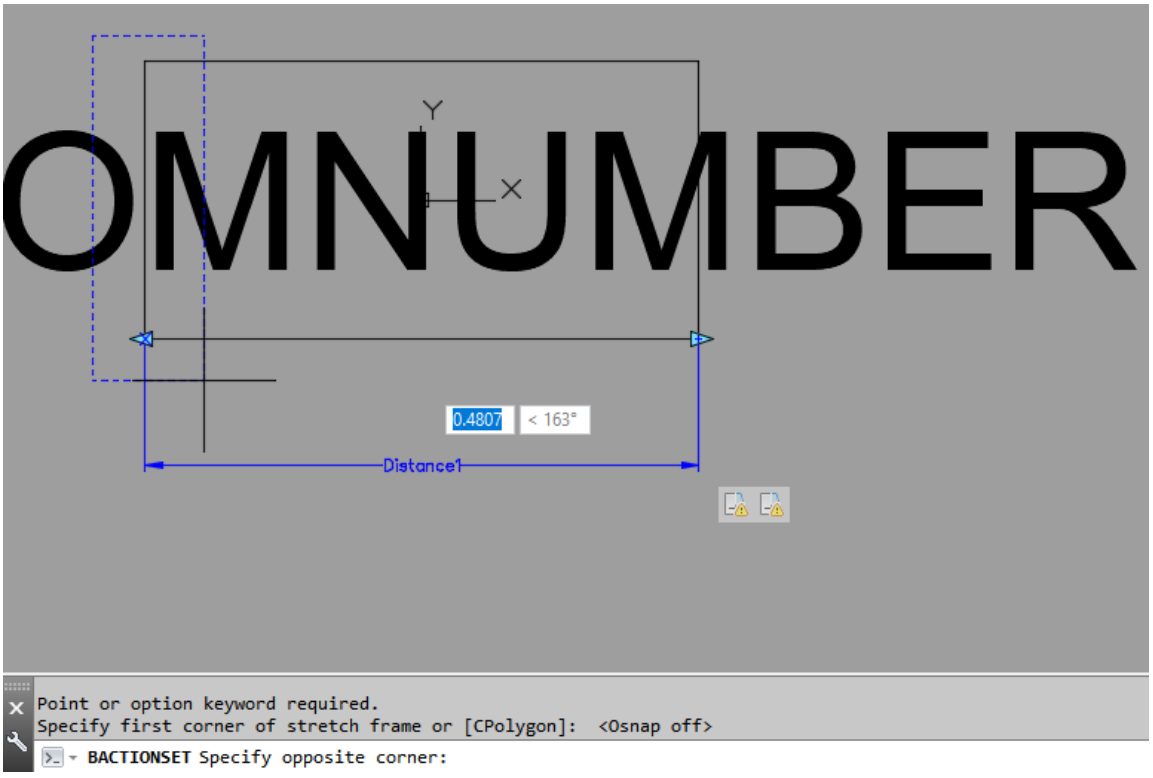
3. In the block, snap the **endpoints** along the bottom of the rectangle, from **lower left** to **lower right**, to add the parameter:



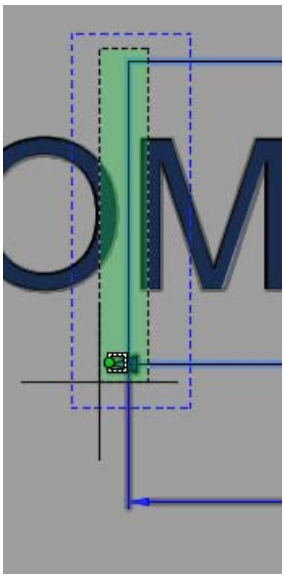
4. Next, select the **action** icons. The first should be named **Stretch** – right click on the parameters, and choose **Action Selection Set > New Selection Set**:



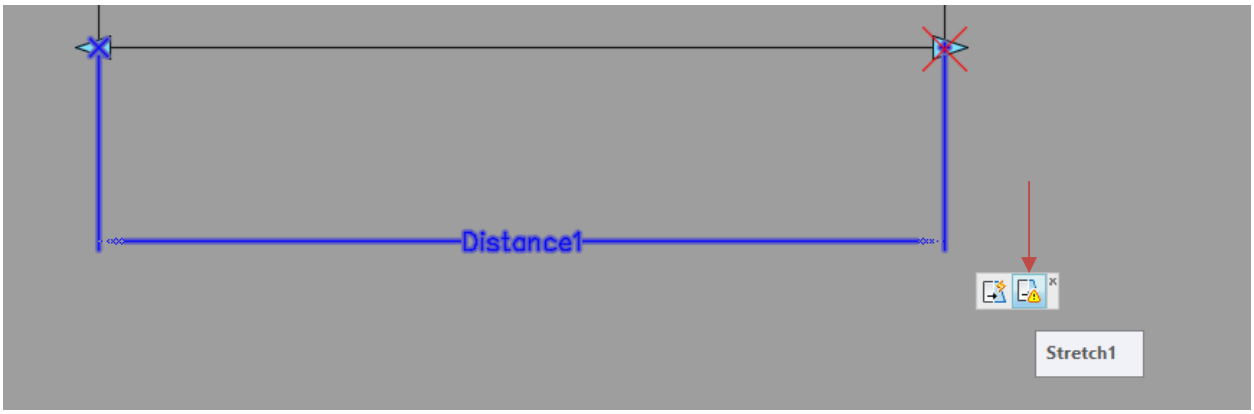
5. The first step lets you use a window to create a frame that is used to define the stretch area. Use a crossing window to define a box that encloses the **left side of the rectangle** (tip – turn off object snaps at this point to make it easier to define the box):



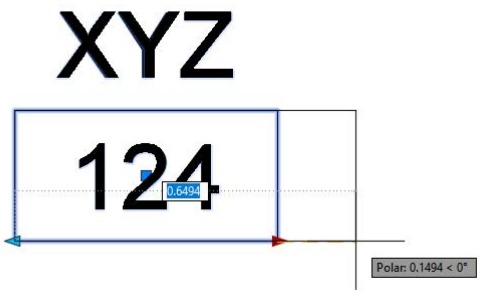
6. The next step prompts you to **select objects** – since there is only one item, you can select it, but if there are multiple lines, you may use the **crossing window selection**:



7. Press **enter** to complete the selection, and the stretch behavior is completed:



8. Repeat the steps for the right side of the box and the **Stretch1** action. When finished, Click **Close Block Editor** to exit, saving the changes. You can now select the block, and use the **grip** to stretch the box around the tag, and change its size:



While the workflow is different that Revit, the results can be the same. You can edit the box as needed to make sure the text does not overwrite the box.

Matching Revit Type Behavior with Dynamic Block Visibility and Lookup Tables

This exercise helps the user understand one of the key functions of a Revit family – the inclusion of nested types within a family, where you have similar parts – such as a variety of receptacle types – but can represent them with just a single family, by controlling the visibility of linework. The same behavior can be included in an AutoCAD dynamic block, when lookup tables are used.

Concepts – Types Versus Lookup Table

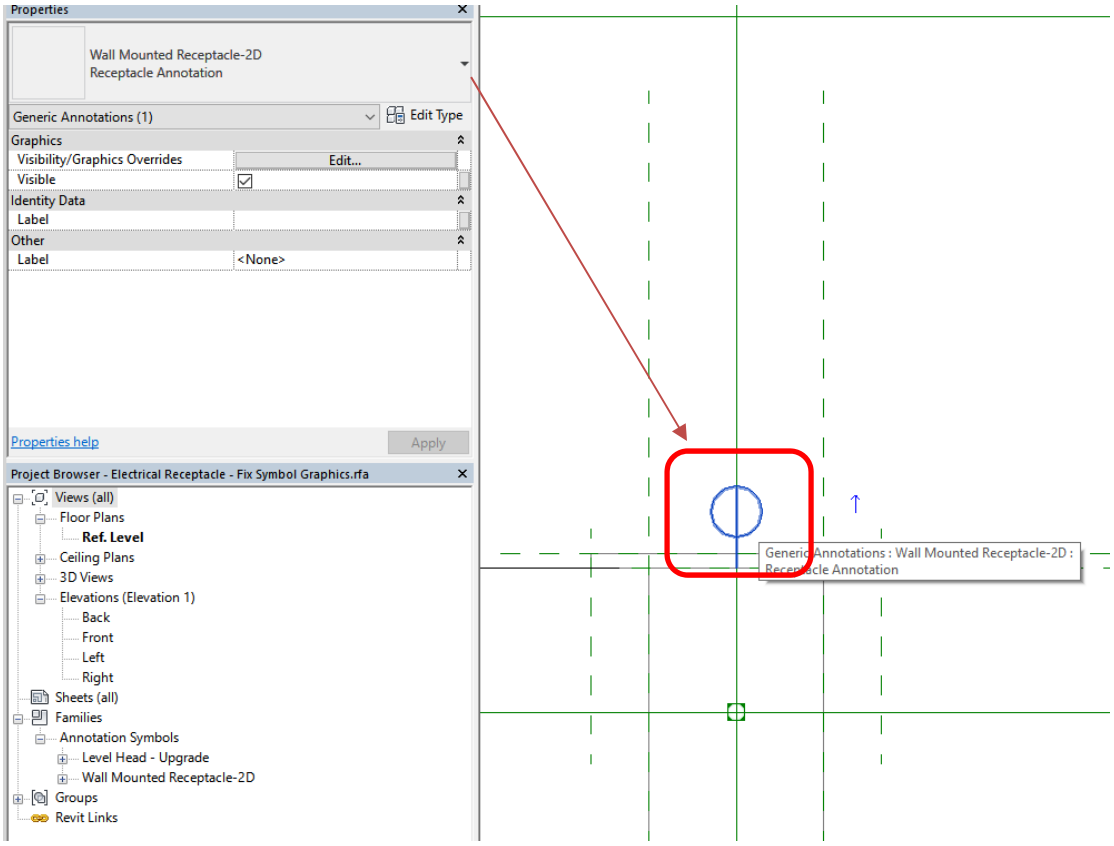
One of the key features of Revit is its ability to create many parametric and symbol family types into one family object. Receptacles, which are used throughout this lesson, are a great example of an item that covers both types of items. Since the receptacle can't be seen well at a large scale, a symbol is used. But there are several representations of how the symbol is used, such as quadraplex, single pole and more. While the box used to house the receptacle may stay the same, you can add nested annotations, or linework with visibility settings enabled, to display the different types.

AutoCAD dynamic blocks can have the same behavior, and save a tremendous amount of space in terms of your library files and drawings. Understanding how the behavior is defined makes the workflow more consistent between applications.

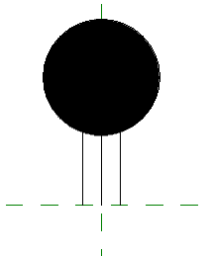
Revit – Adding Yes/No Graphic Controls to Nested Symbols

For you to control what symbols appear with different types in a family, you need to include visibility controls for the linework in the symbol file. This example includes a nested annotation family, where the linework is already assigned to the items in that family, so begin by reviewing these settings.

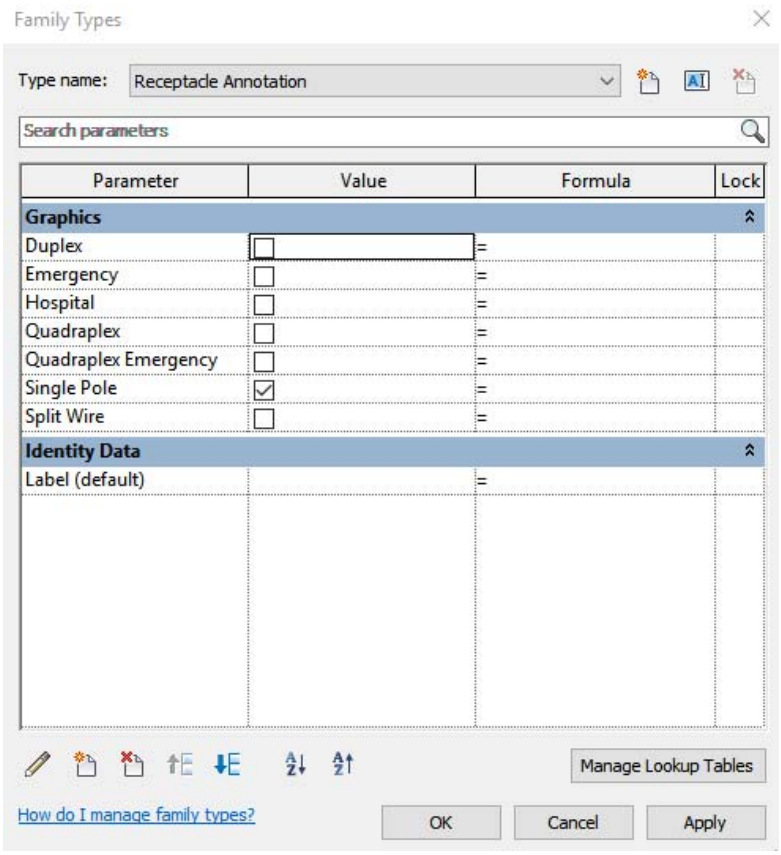
1. From Revit, open the **Electrical Receptacle – Fix Symbol Graphics.rfa** file. Once the file is open, locate the **2D symbol** in the **Floor Plan – Ref Level** view:



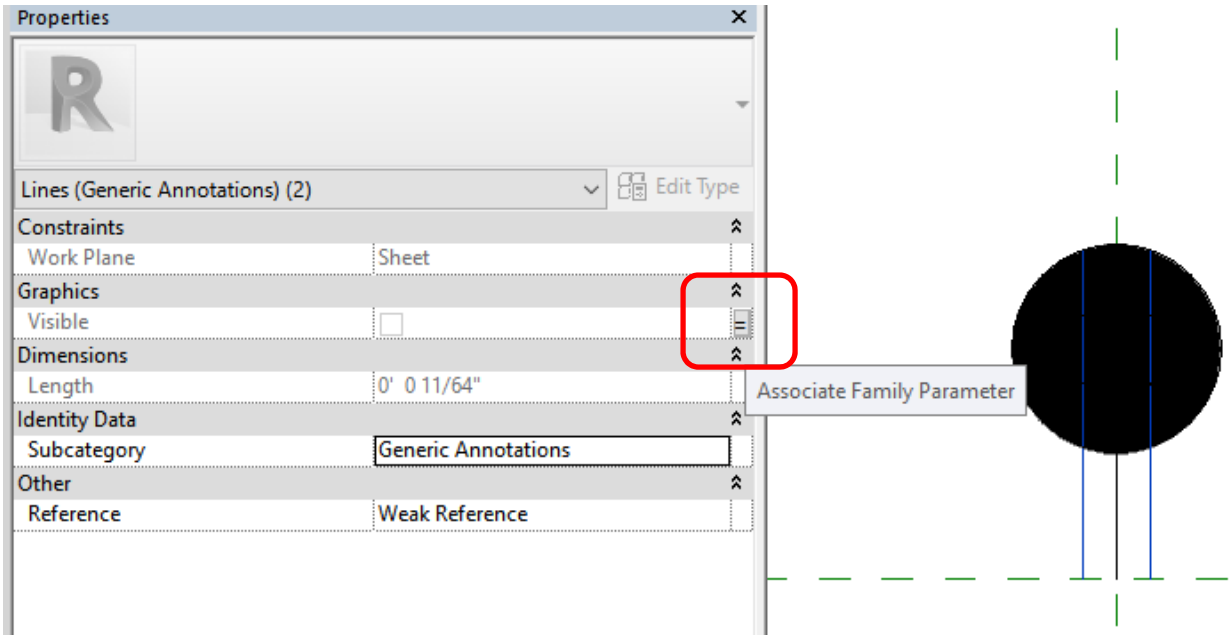
2. While the symbol is selected, choose **Edit Family** from the **ribbon** to open the family. You will several lines and fill patterns in the view:



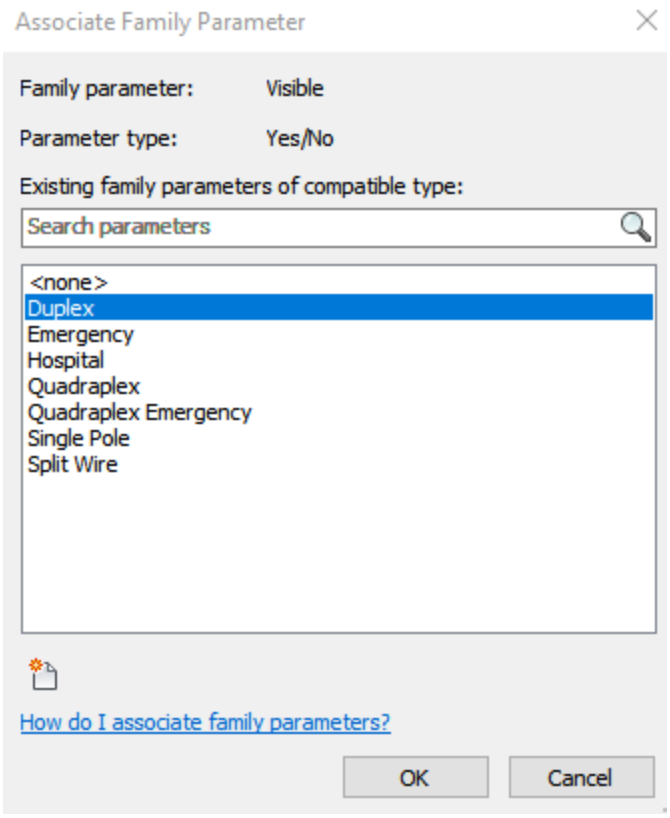
3. From the ribbon, **Create** tab, **Properties** panel, click **Family Types**. The dialog will appear – review the list of predefined visibility parameters:



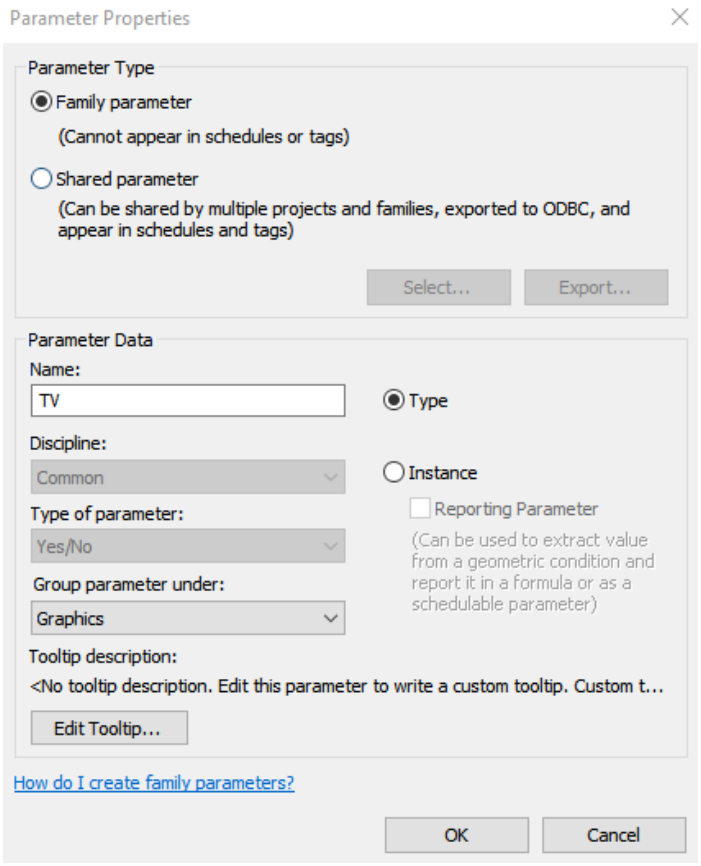
4. The lines in the view are controlled by these parameters. Click **OK** to close the dialog. If you select a line, the **properties palette** will indicate that the **Visible** parameter is using an **associated family parameter**:



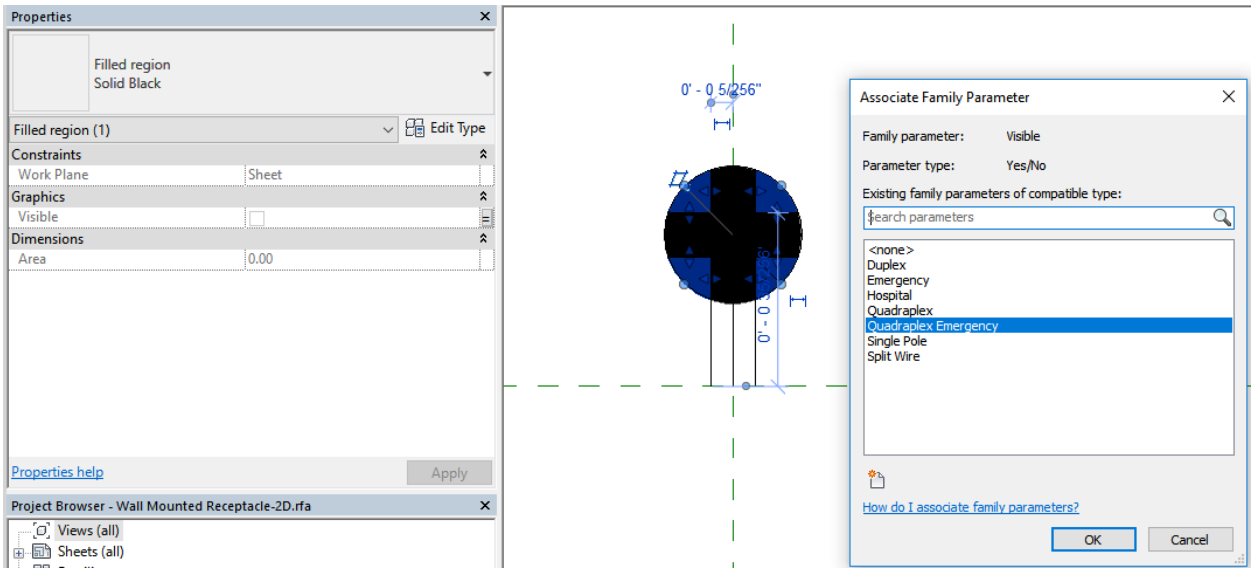
5. If you select the **link**, the **Associate Family Parameter** dialog will appear. The lines are assigned to the **Duplex Parameter**:



6. It's easy to do this – simply **select** the lines you want to control, and then **associate** them with the **visibility** parameter. You can define these with the **New** icon in the lower right corner, creating them as **family parameters**:

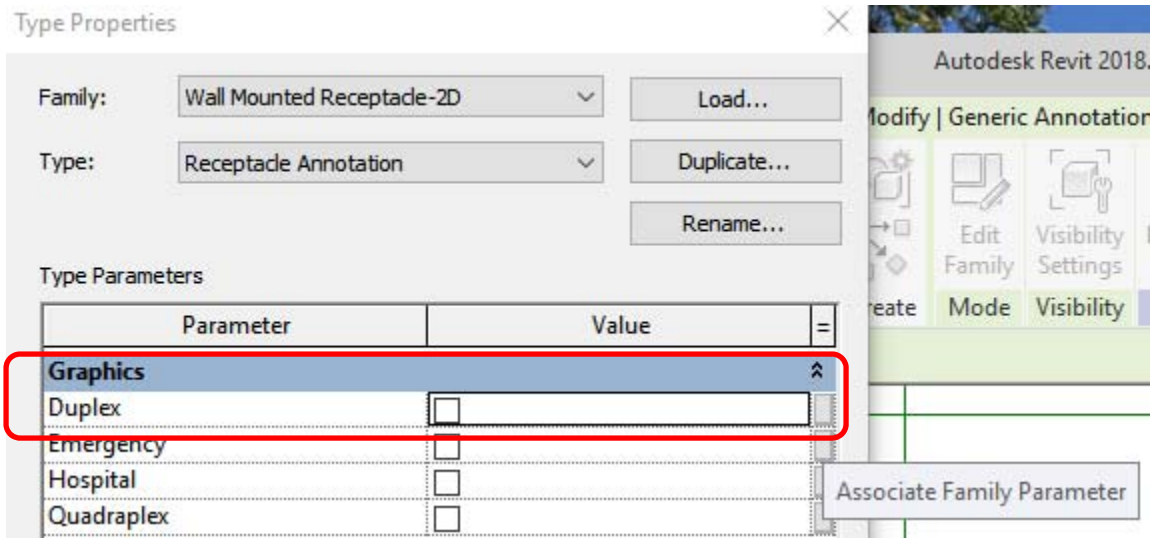


7. It is recommend using **Type** for all of these, since you will be placing multiple examples of each type throughout the project. Click **Cancel** to exit this dialog. As you select other items, such as the **outside circle** or **fill pattern**, you can see what items are associated with the parameter:

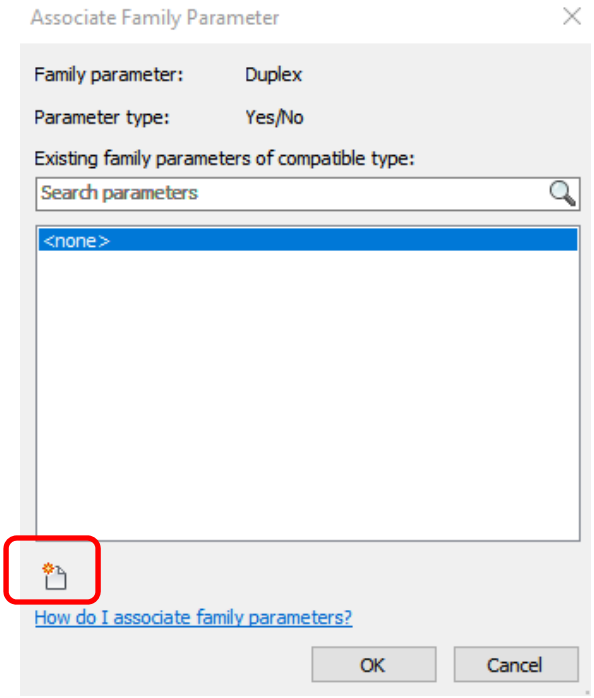


Here's a crucial point – you can have different/multiple visibility options turned on in a family, to help control what you see. **Close** this family, and return to the **Electrical Receptacle – Fix Symbol Graphics**.rfa file.

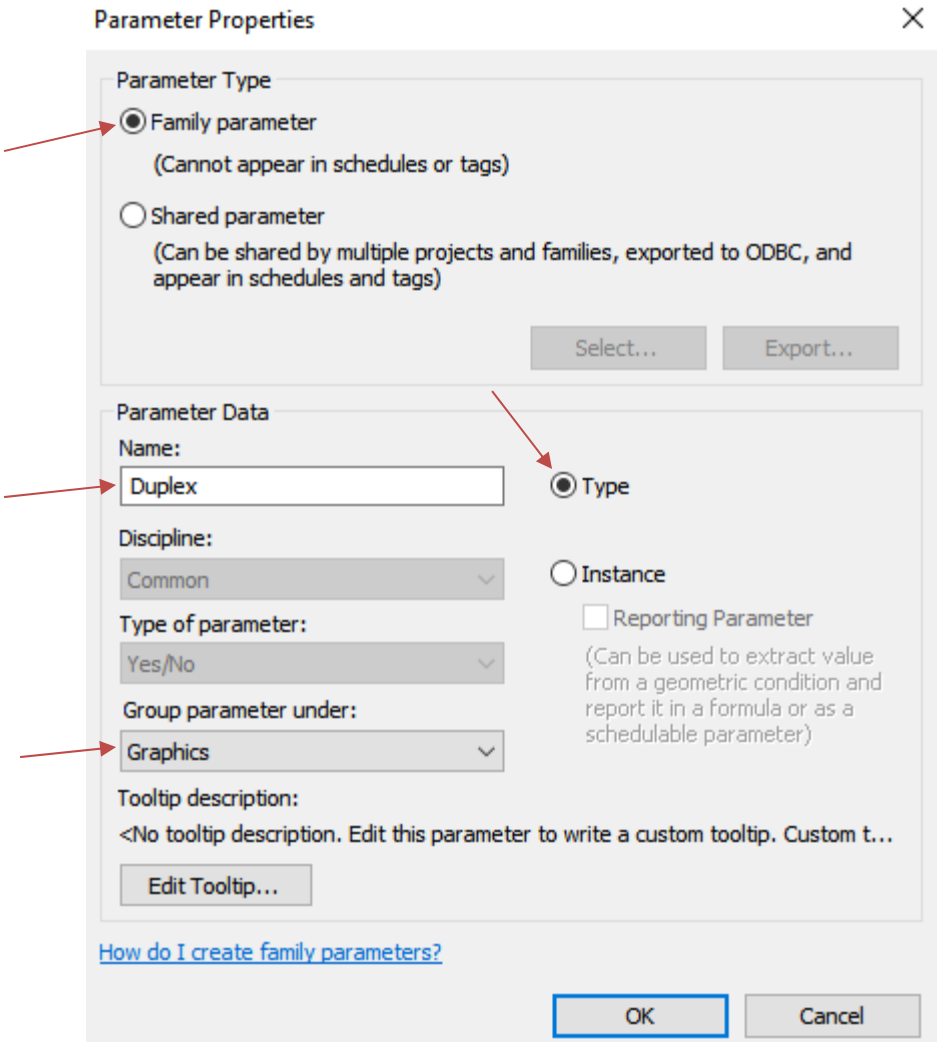
8. Select the **symbol family** again, and from the **properties palette**, click **Edit Type**. For the assign visibility parameters to be used with the type families in this file, you must also "expose" these settings using the **Associate Family Parameter** tool:



9. Use **Duplex** as the first example. After selecting the **associate** tool, use the **New Parameter** tool to add the **Duplex** visibility parameter, using the same settings as in the nested annotation family:

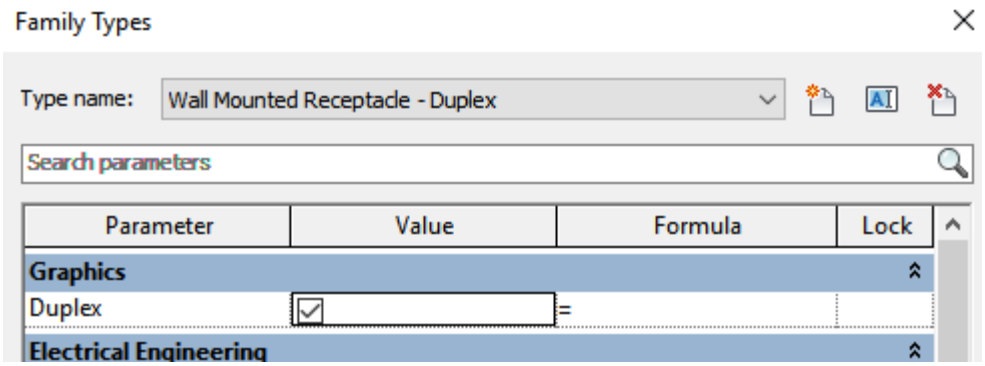


10. When the dialog appears, make sure you select **Family** for the parameter type, and type the **name** of the parameter exactly the way you want it to appear. Use the **type** value, and make sure you edit the **parameter group** to use the **Graphics** category:

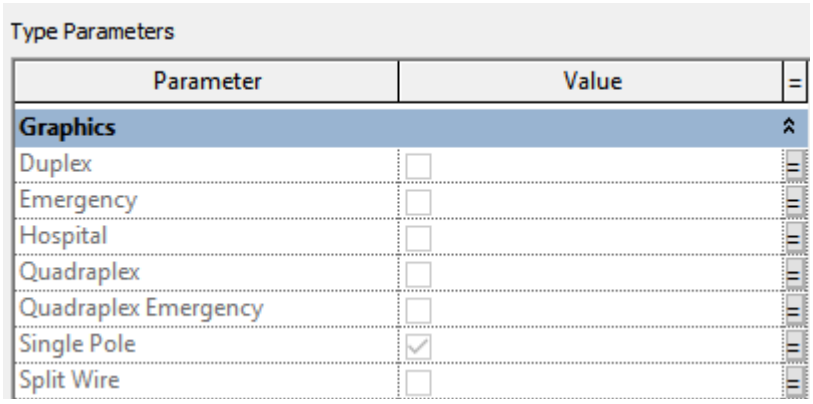


11. Click **Ok** twice to exit both dialogs. The **Duplex Parameter** is now greyed out, meaning it can now be controlled by the family type properties. Click **OK** to close this dialog.

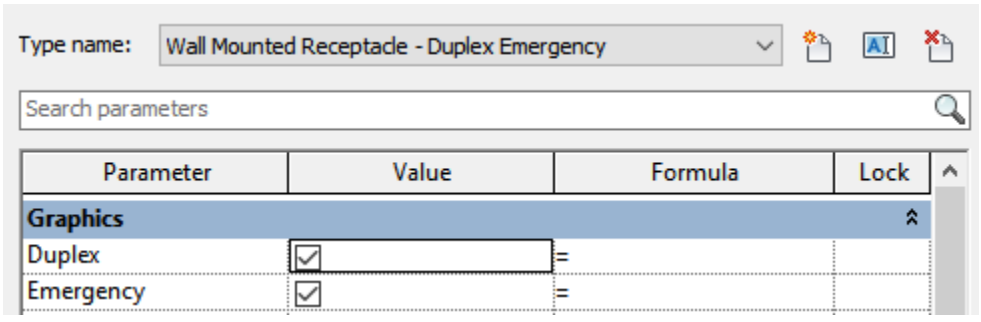
12. From the ribbon, **Create** tab, **Properties** panel, click **Family Types**. Set the family to use the **Wall Mounted Receptacle**, and check the **Duplex** option. This symbol will now appear in plan views for this family.



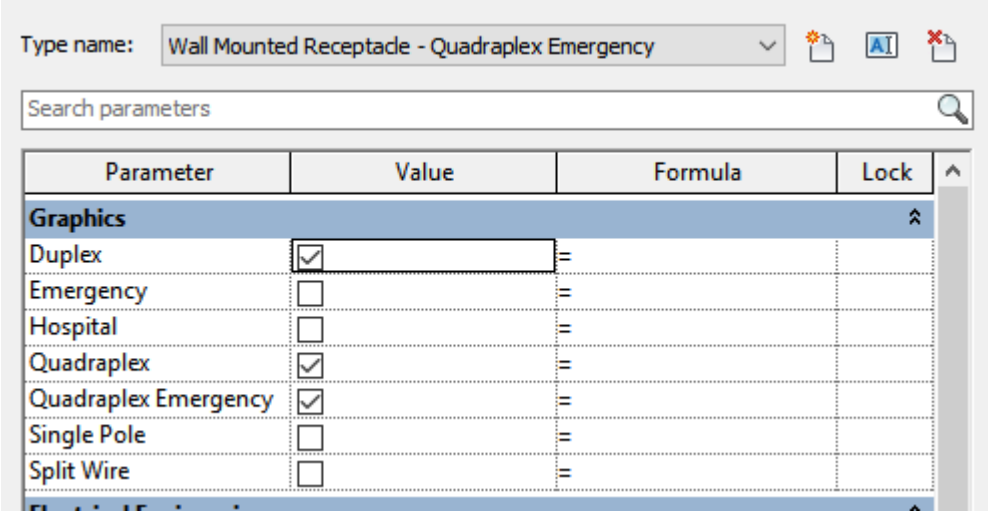
13. Return to the annotation symbol, and **associate** the rest of the types, so that they are all **exposed**:



14. After completing this step, edit the other receptacle **types** in the family. For example, the **Duplex Emergency** type could include two items checked – **Duplex** and **Emergency**. In this case, the emergency graphics are just that part of the symbol, so both are used to create one graphic representation of the fixture:



15. Other can include multiple combinations:



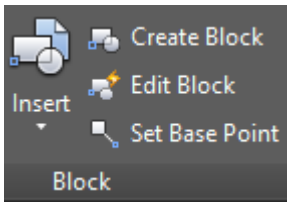
The point is to reuse as much of the linework as possible, instead of creating individual symbol families for each representation of the fixture.

The project includes a completed example, **Electrical Receptacle - Wall Mounted - Non-hosted**, that you can review. Place some examples into a power plan or coordination plan to see the differences. It's a great example of how annotative items can be combined into one file, saving you time and space in your project.

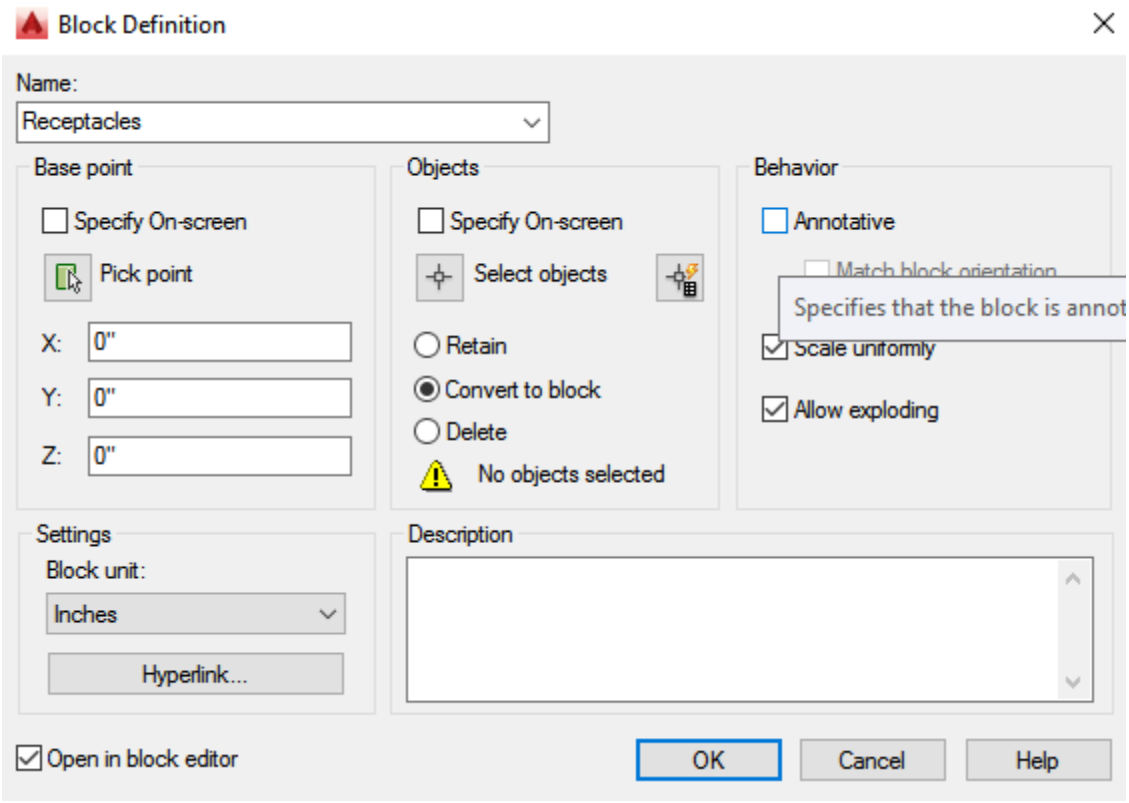
AutoCAD - Creating a Lookup Table of Blocks

In this exercise, you will need to open the drawing **Receptacle – Wall Mounted.dwg**.

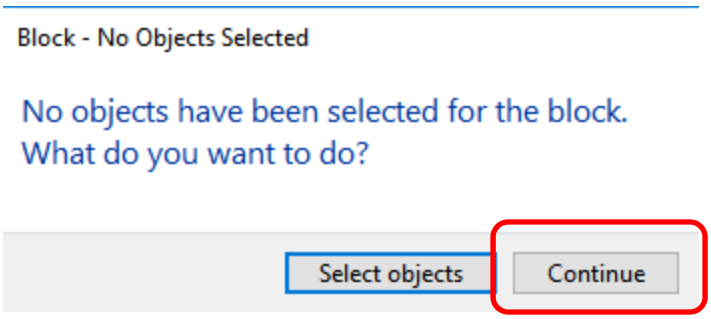
1. From the **Home** (or **Insert**) tab, **Block** Panel, choose **Create Block**:



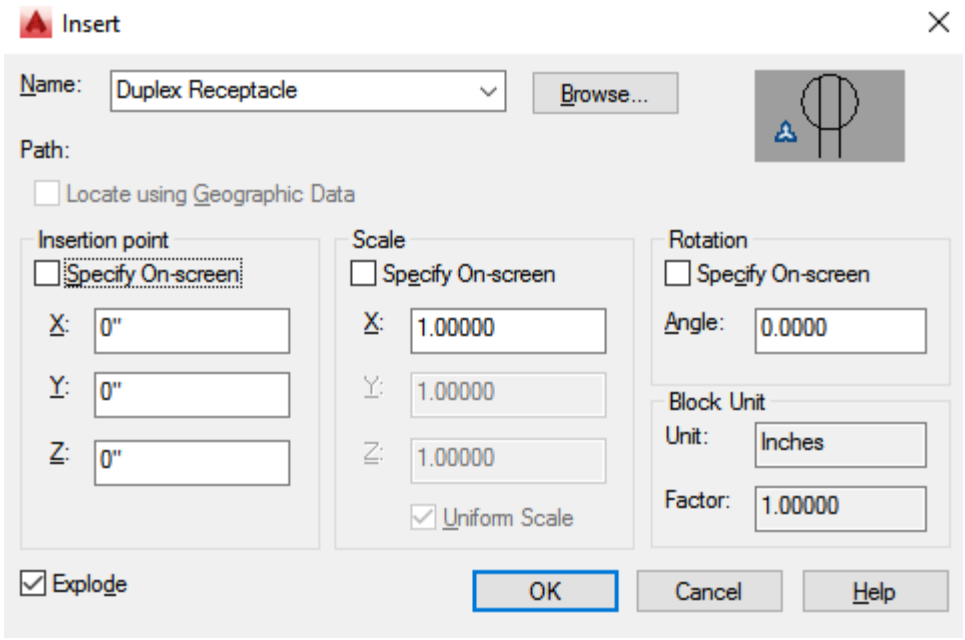
2. When the dialog appears, name the new block **Receptacles**. Make sure the **base point** is set to **0,0,0**, and **do not select** any objects.



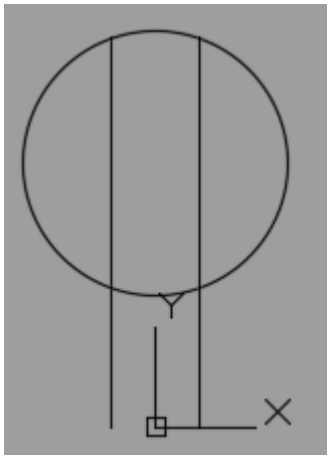
3. Make sure the **Open in Block Editor** option is selected, and then click **OK**. You will get a warning about “block - no object selected” – click **Continue**:



4. Once the block is defined, you use the **Insert** command to load the block examples, and convert them using the **Visibility** parameters. From the command line, type **INSERT** and press **ENTER**. The **Insert Dialog** will appear:

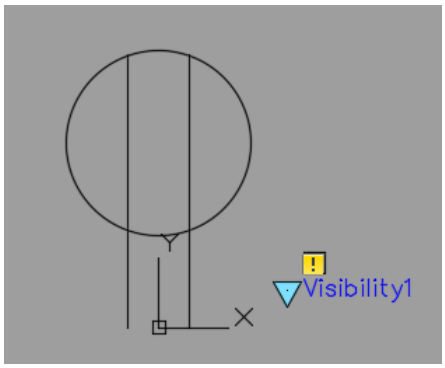


5. A block that is already loaded into the drawing is displayed. Make sure the **insertion point** is set to **0,0,0** and the **Explode** option is selected. This prevents nested blocks from being used. Click **OK**, and the linework representing the receptacle will be added:

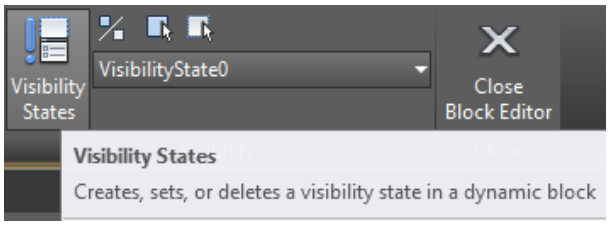


6. From the **Block Authoring** palette, **Parameters** tab, click **Visibility**:

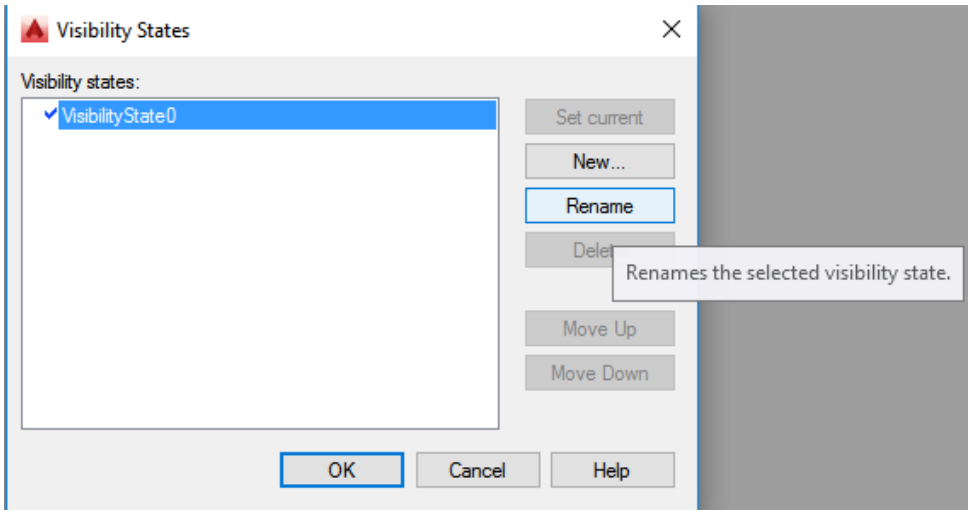
7. Place the **parameter** adjacent to the linework:



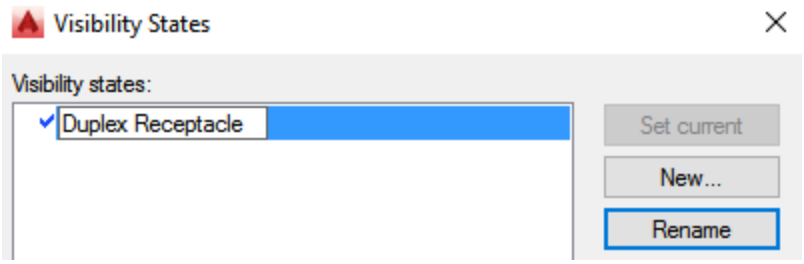
8. The **parameter location** sets the location of the **drop-down list grip**. Now that the parameter is added, go to the **Block Editor** tab, **Visibility** panel on the ribbon, and click **Visibility States**:



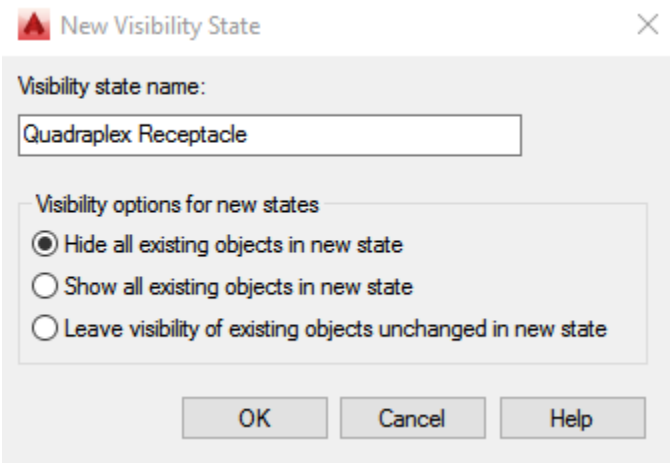
9. The **Visibility States** dialog will appear – select **Rename** to define the duplex receptacle:



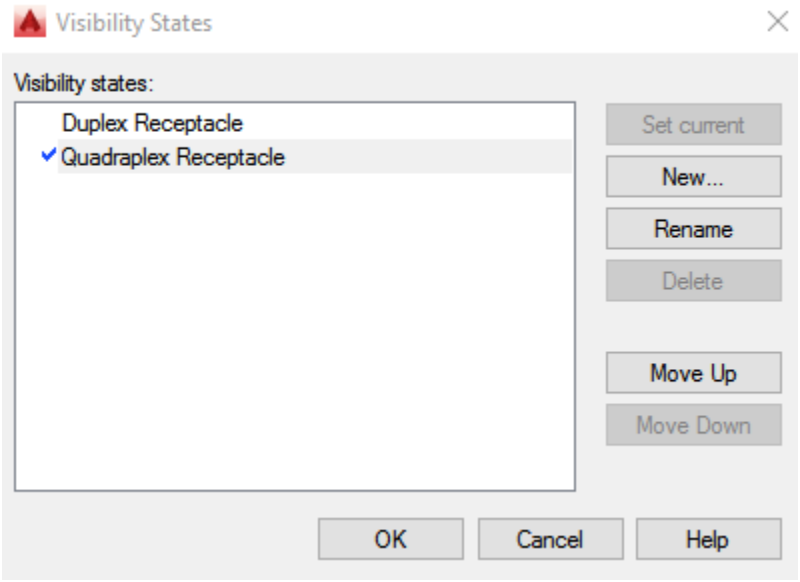
10. The item will be available for editing – name the state **Duplex Receptacle**:



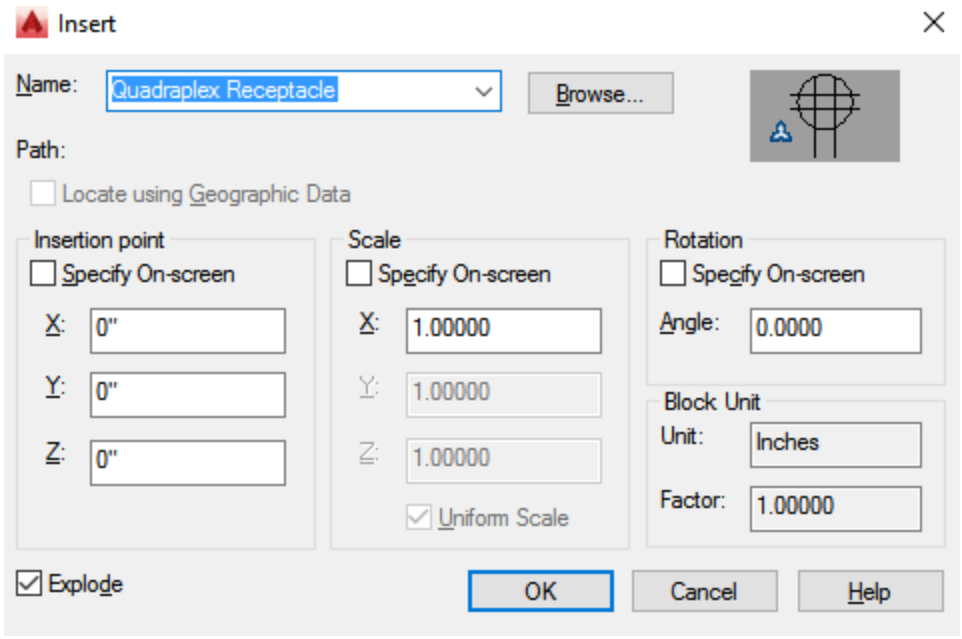
11. Once this first item is renamed, add the remaining states as needed, for the remainder of symbols you want included in the block. Click **New** and then add the name **Quadraplex Receptacle** – make sure you select **Hide all existing objects in new state**, so that only the new quadraplex receptacle will remain visible for the next step:



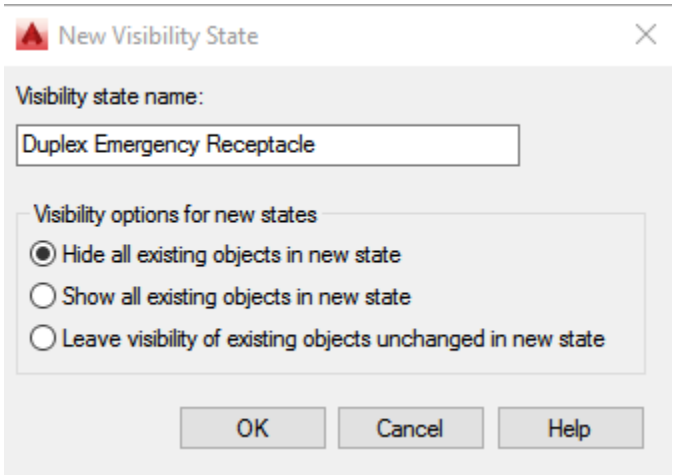
12. Click **OK**, and the new state will appear with a **check mark** beside it.



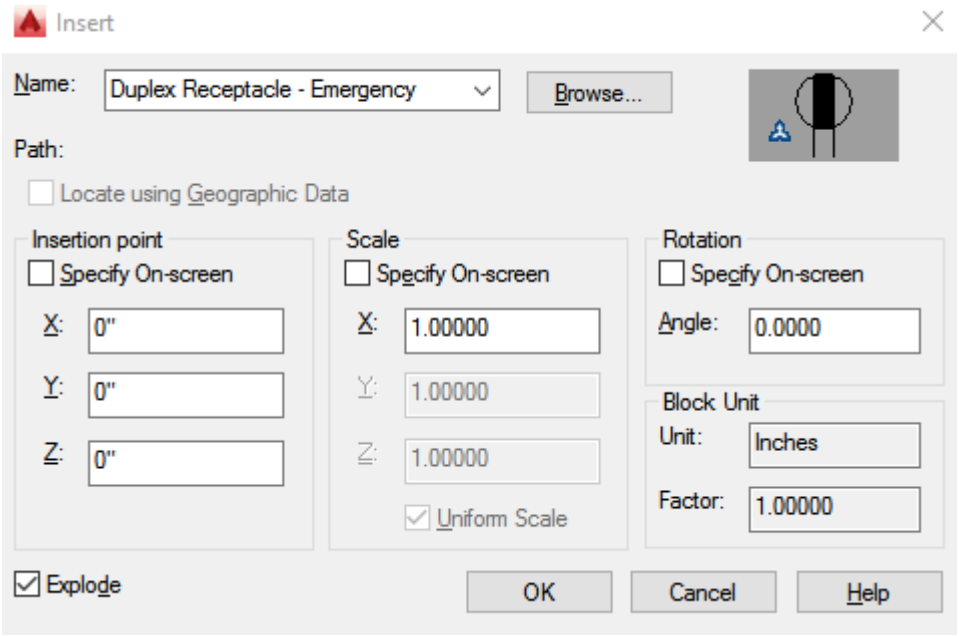
13. Click **OK** to close the dialog. The original lines and circle representing the duplex receptacle will not be visible. Repeat the **insert** command to place the **Quadraplex Receptacle**, with the **Explode** option selected:



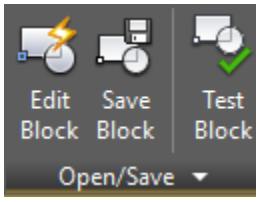
14. The linework for the quadraplex receptacle will appear, and be used for that state. Next, from the ribbon, click on the **Visibility States** tool again, and then click **New** again. Add the state **Duplex Emergency Receptacle**, using the **Hide All option in the new state**:



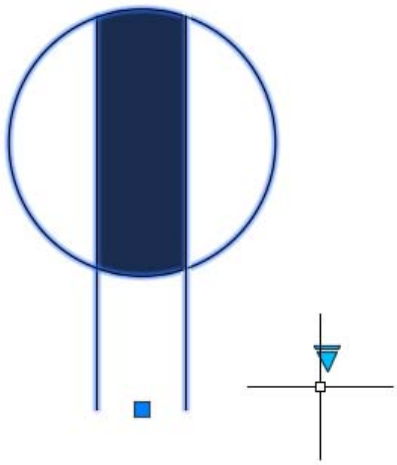
15. Click **OK**, and **OK** to close the Visibility States dialog. Repeat the **insert** command to place the **Duplex Receptacle – Emergency** block, with the **Explode** option selected (*note – the defining block name used to create the linework does NOT have to match the visibility state*):



16. Click **OK**, and the last symbol in the block is defined. From the ribbon, **Block Editor** tab, **Open Save** panel, click **Save Block**, and then use the **Close Block Editor** tool to return to the drawing (save the changes if you are prompted):



17. Save the current drawing. From the command line, type **INSERT** and press **ENTER**. Select the block **RECEPTACLES** and place an example in the drawing.



You can now use the grip to swap the symbol in the view.

Conclusions

It pays to make sure that you create consistent and repeatable tools across both software platforms. The more you can emulate behavior from AutoCAD to Revit, the easier it is for the user to retain their skills, and become more productive with both applications. Get your annotation and symbol content nailed down first, and then you can move on to bigger and better designs.