Bentley Institute

Practice Workbook

This workbook is designed for use in Live instructor-led training and for OnDemand selfstudy.
The explanations and demonstrations are provided by the instructor in the classroom, or in the OnDemand eLectures of this course available on the Bentley LEARN Server (*learn.bentley.com*).

This practice workbook is formatted for on-screen viewing using a PDF reader.
It is also available as a PDF document in the dataset for this course.

Using Parametric Solids, Constraints, Variables and Variants: Putting it all Together with Parametric Modeling

This workbook contains exercises that will assist experienced 2D MicroStation users become familiar with 3D tools and concepts, specifically parametric solids, 2D and 3D constraints, variables, variants and parametric cells within the MicroStation CONNECT Edition. The old saying, 'It's never done until it's done' comes to mind when working with 3D designs. Models need to be altered and edited throughout the design process as quickly and easily as possible. In the past, you may have recreated models rather than edit an existing model as needed. Now, with the MicroStation CONNECT Edition, learn to create reusable models and cells that can be dynamically adjusted and sized with parametric dimensions and constraints.

TRNC03557-1/0001

Description and Objectives

Learning 3 Dimensional (3D) drawing with MicroStation CONNECT Edition is an extension of your knowledge of 2 Dimensional (2D) drawing. If you can draw in 2D, then all that is needed is to learn how to place 3D Primitives, Manipulate 3D Geometry and Features, add 2D geometry, apply constraints, Extrude or Revolve, Create and apply variables and variants, sounds easy right ? It is easier to do than what it sounds. In this example a tank will be created by creating a 2D profile drawn in a 3D model, constrained geometrically and dimensionally then revolved to create the tank. Also a Traffic Signal Light Post will be created as a 3D Parametric model, constrained both geometrically and dimensionally, created as a parametric cell , along with several variants.

Course Description

This workbook contains exercises that will provide the basic skills required to use:

Skills Taught

- Dimensional Constraints
- Geometric Constraints
- Solid by Revolution
- Feature Solids>Shell
- Variable Definition
- Variants
- Solid Slab and Cone
- Parametrics with Solids
- Coincident Constraint
- Concentric Constraint
- 3D Dimensional Constraint
- 2D Dimensional Constraint

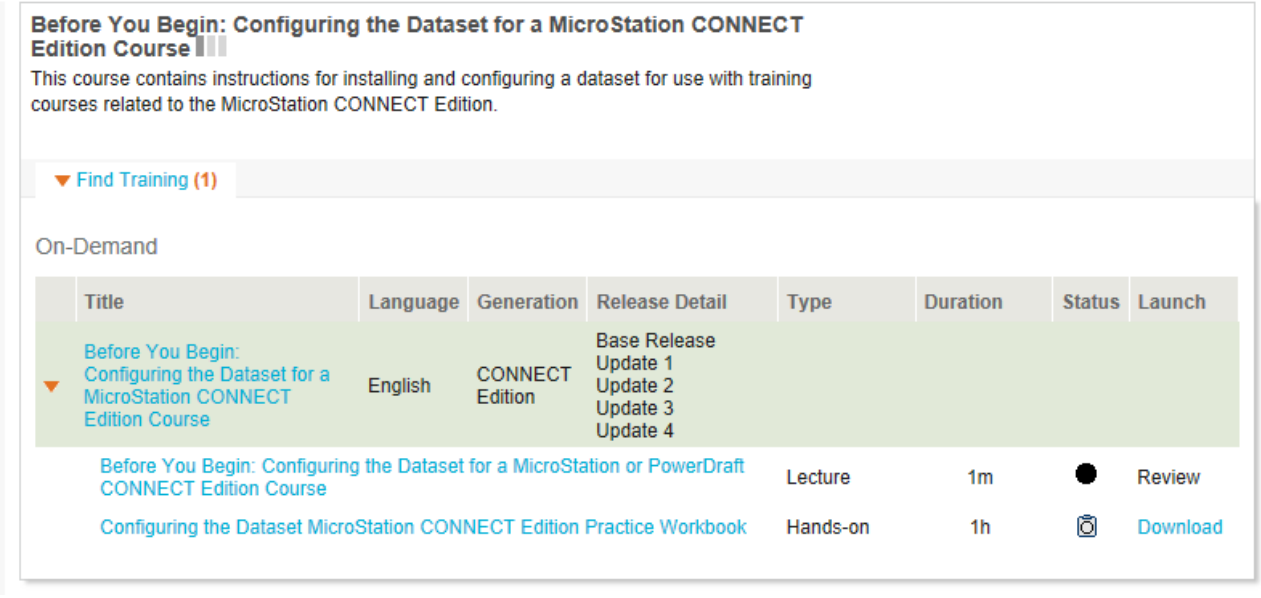
- Solid by Extrusion
- Solid by Sphere
- Place Parametric Cell

Note that the exercises contained in this workbook are designed to use a custom WorkSpace, BentleyCONNECTTraining, and several WorkSets including the *WorkSet* **QuickStartForCONNECT**. The default installation folder for the datasets is:

C:\BentleyCONNECTTraining\WorkSets

For instructions on how to configure a training dataset, please visit **Learn.Bentley.com** and view the following training course:

Before You Begin: Configuring a Dataset for a MicroStation CONNECT Edition Course



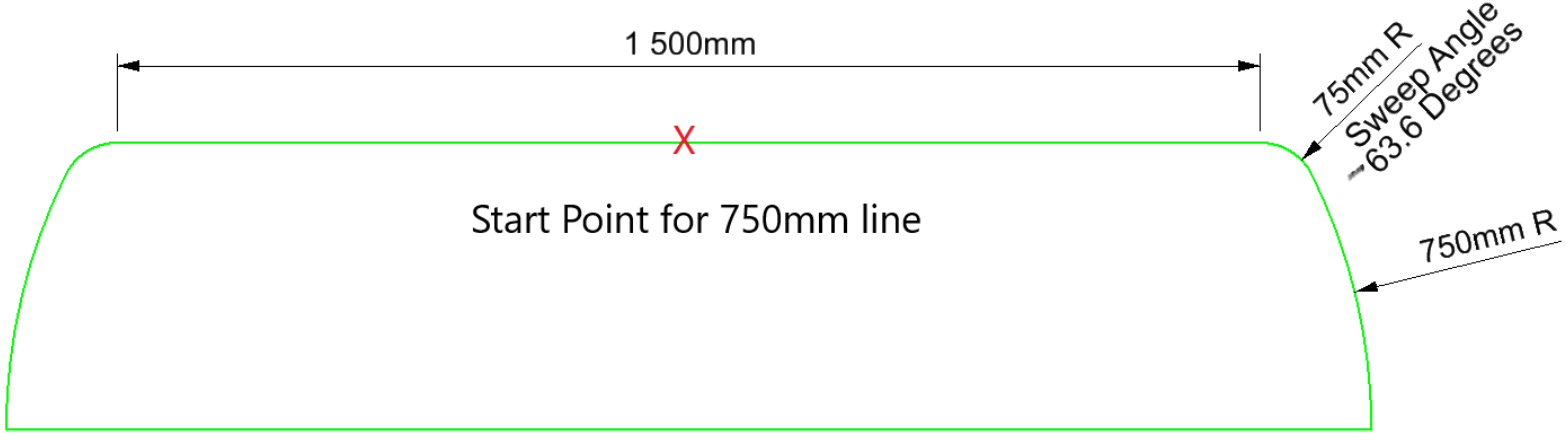
Creating the Tank

In this section, apply constraints and variables to a 2D profile and revolve it to create a 3D Parametric Pressure Vessel "Tank". The "Tank" will be created based on design criteria applied through constraints both dimensional and geometric. Open the MicroStation design file:

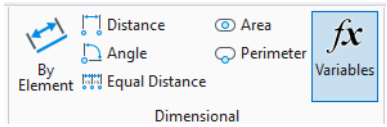
- **Tank Created from 2D to 3D.dgn**

---

1. Set the **Active Level** to *PressureVesselHull* and select the **Place Smartline** tool.
   - Provide a datapoint anywhere in the *Top* view. Using AccuDraw draw a horizontal line, *750mm* to the right.
   - Without hitting Reset draw, <~> (tilde) a tangent Arc with a radius of 75mm whose sweep angle is precisely -63.6 degrees.
   - Follow this with another tangent arc defined by a radius of 750mm whose ending point is aligned with the global X (horizontal) axis (Use <V> to Rotate the compass relative to the View).
   - Complete the closed shape profile as shown, note that the geometry must be closed. Remember Create Region or Create Complex Shape can be applied after it is drawn. Geometric and dimensional constraints will be added to the profile next.



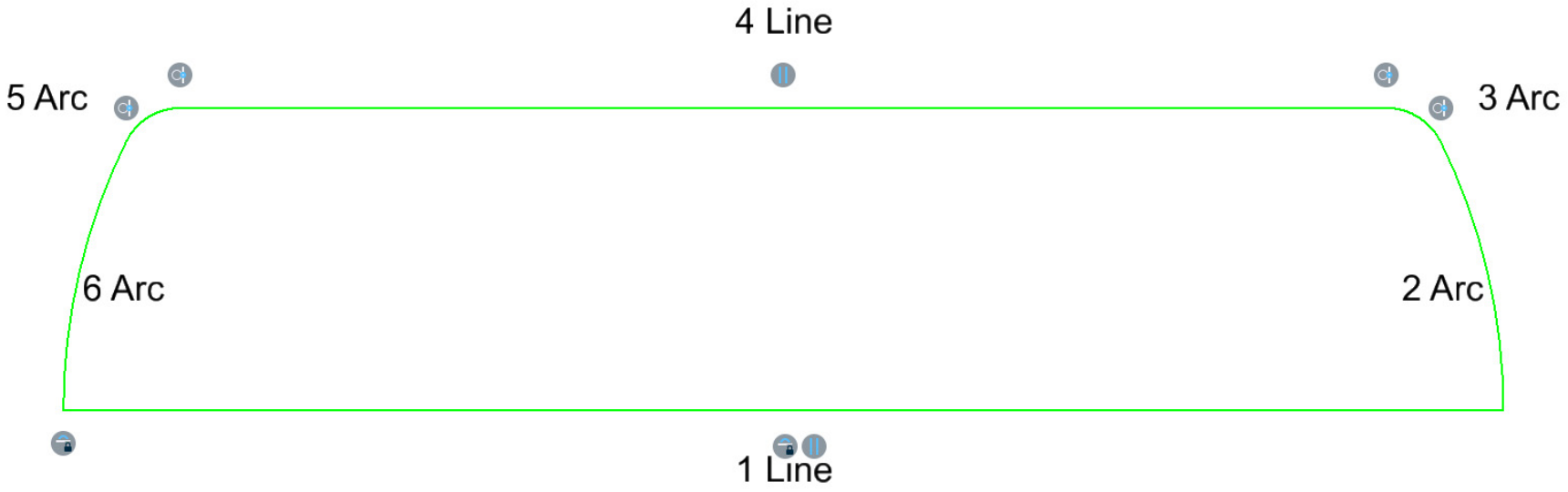
2. Pick **Drawing Workflow>Constraints Ribbon Tab>Dimensional>Variables**.



Select *Local Variables*, then **New**. Create the following five local variables based on the table below. Create the Length variable first.

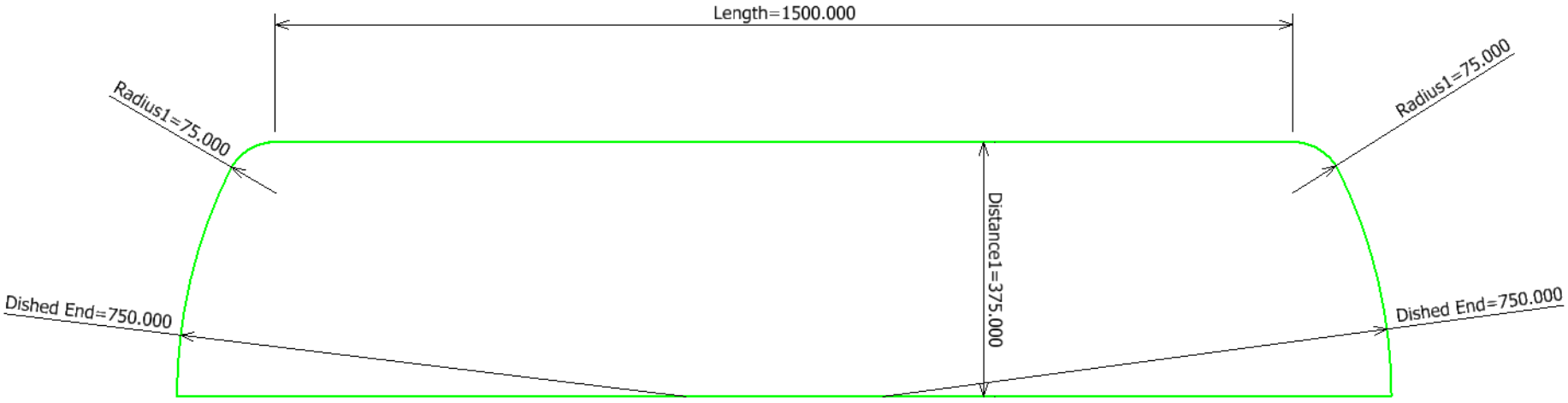
Variable Name	Expression	Resultant Value
Length	(None)	1500mm
Dished End	Length / 2	750mm
Distance1	Dished End / 2	375mm
Radius1	Distance1 * 2 / 10	75mm
Tank Thickness	Radius 1 / 5	15mm

3. Apply the following *Geometric 2D Constraints* to the geometry, using the table below for reference.



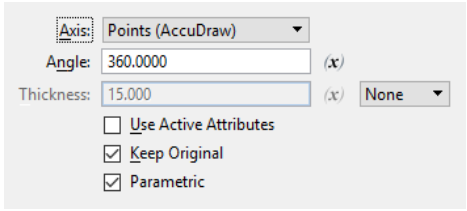
Element	Constraint
1 Line	Fixed (Fixes Left Point)
1 Line	Fixed (Locks Angle)
2 Arc	Tangent to 3 Arc
3 Arc	Tangent to 4 Line
4 Line	Parallel to 1 Line
4 Line	Tangent to 5 Arc
5 Arc	Tangent to Arc 6

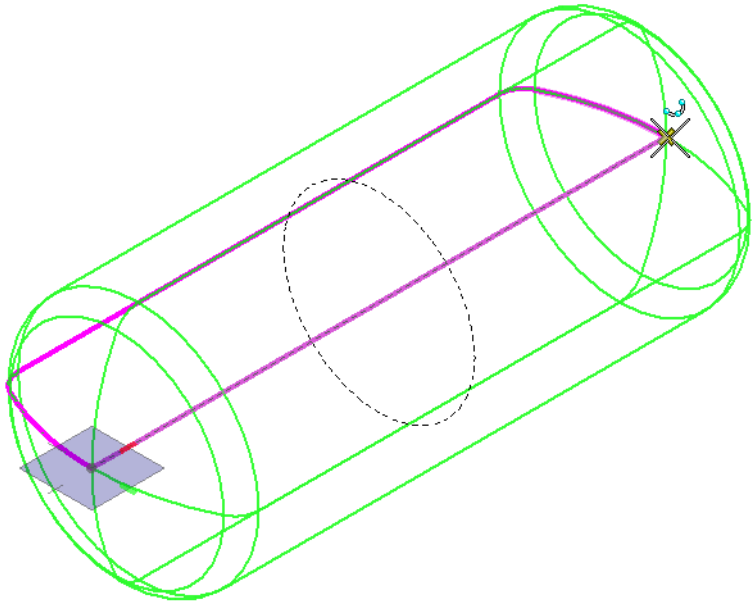
4. Next, apply the following *Dimensional Constraints* to the geometry, use the image below as reference.



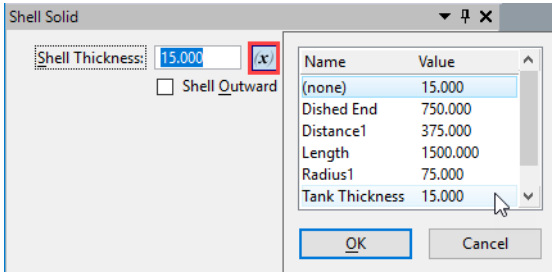
Note: The values of the variables will be applied to the geometry when the dimensional constraints are placed. For example, the vertical dimension is not the exact value of Distance1 (375) until the dimensional constraint is placed.

5. Changing the view orientation to the Isometric View and from the *Solids Ribbon Tab* on the *Modeling Workflow*, pick **Revolve** from the *Create Solids Ribbon Group*. Set *Solid by Revolution* as shown and pick the Solid, defining the Axis of Revolution with two datapoints.





6. From the *Solids>Features>Shell,* setting the **Shell Thickness** to the variable **Tank Thickness** and select the new Solid (careful not to select the profile) and accept to shell the Solid.



7. Next, change the value of the *Length* variable (this is the only variable that can be edited, all other variables are derived from the value of this variable).

   Length: **2250mm**

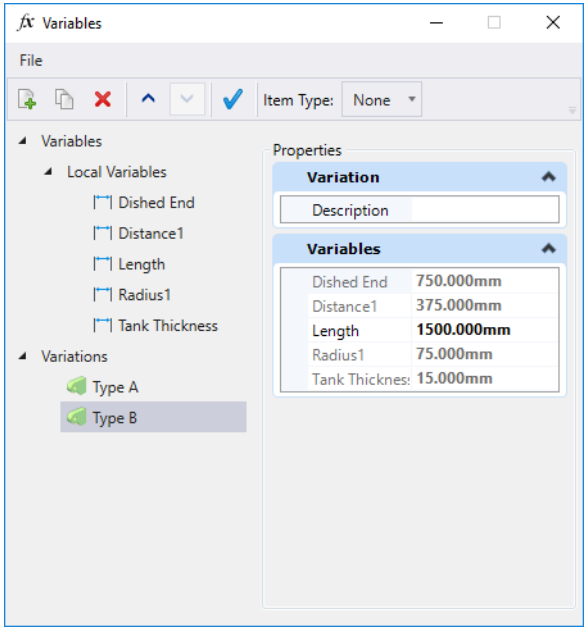
Note the geometry updates with this change in value.

Note: To review, all variables other than *Length* are based on other variables and expressions. The variables allow the design intent to be maintained.

Variable Name	Expression	Resultant Value
Dished End	Length/2	1125mm
Distance1	Dished End/2	562.5mm
Radius1	Distance1*2/10	112.5mm
Tank Thickness	Radius1/5	22.5mm
Length	(None)	2250mm

8. Create a **New** variation from the *Variables* dialog, **Type A**, based on the current settings.

   Next create a **New** variation **Type B** with one change to the variable *Length*, set this for *Type B* to **1500mm**.



9. **Apply** the variations of *Type A* to the active model, followed by *Type B*.

Creating a Parametric Traffic Signal Light Pole for Visualization Purposes

In this section you will create a traffic and signal lighting pole as pictured. To create the traffic signal light pole and mast arm, various tools will be used including Slab, Variables, Cones, Constraints, Extrusions and Sphere. All used to create a Parametric Traffic Signal Light Pole.

---

1. Open the design file **TrafficSignalLight.dgn** and open the model *TrafficSignalLight*. Begin by creating the base of the Traffic Signal Light.
2. Set **Active Level** to *Base*

- From the **Constraints Ribbon Tab>Dimensional Ribbon Group>Variables**, create the following new variables:

  BaseSize, *Distance*, *Instance*, *Visible*, *2.25*

  BaseWidth, *Distance*, *Instance*, *Hidden*, Equal to *BaseSize*

  BaseLength, *Distance*, *Instance*, *Hidden*, Equal to *BaseWidth*

  BaseHeight, *Distance*, *Instance*, *Hidden*, Equal to *BaseWidth*

Note: If the Width, Length, Height were always equal, such as in our example, individual variables are not required. W, L, H could be defined as BaseSize. However in our design we may want the freedom to change the individual values of W, L, H.

- Pick *Modeling Workflow>Solids Ribbon Tab>Primitives Ribbon Group>Slab*

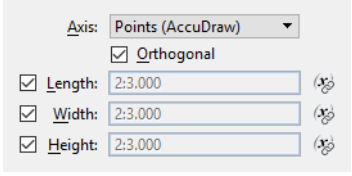
  Axis: *Points (AccuDraw)*

  Orthogonal *On*

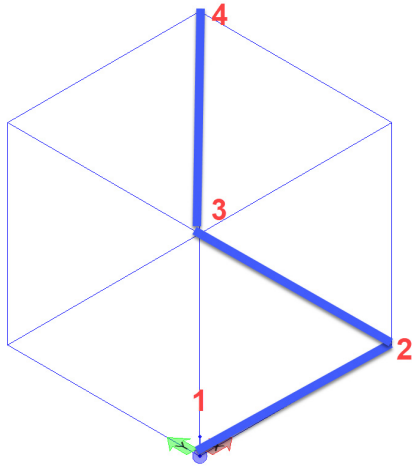
  Length set to variable *BaseLength*

  Width set to variable *BaseLength*

  Height set to variable *BaseLength*



- Place the start point on the ACS triad at 0,0,0, using AccuDraw, set the drawing plane to be <T> Top. Provide a data point in the Positive X direction to create the *Length*, another data point in the Positive Y direction to define the *Width* and finally the *Height* in the Positive Z direction



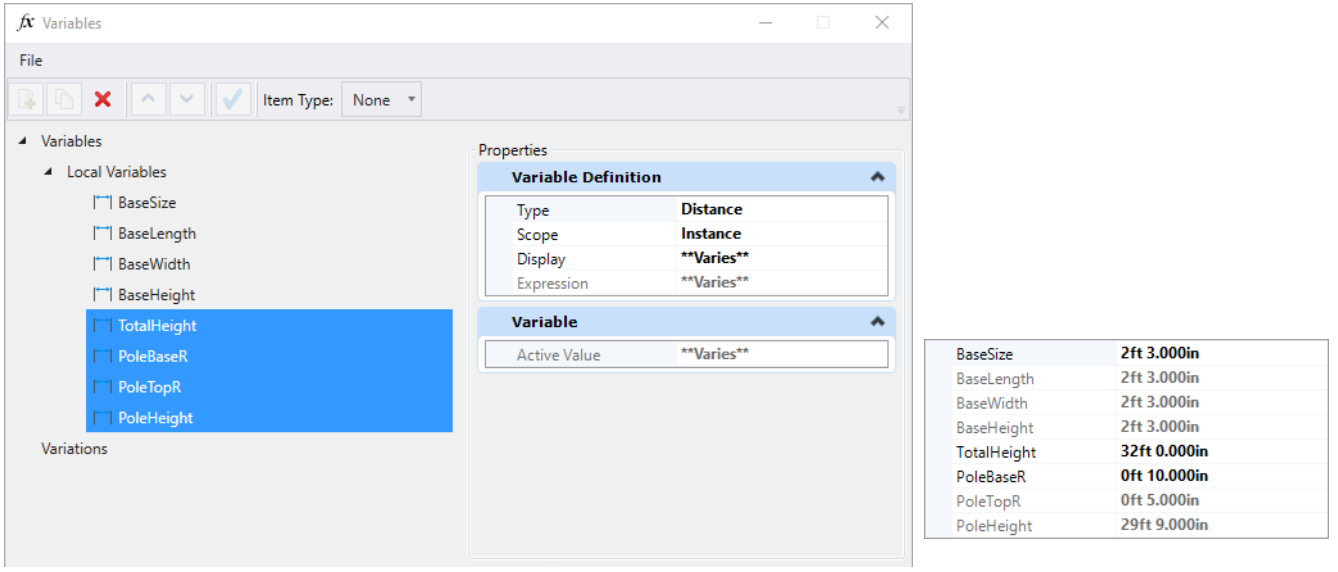
3. Next create the Pole, centered on the base. Set *Active Level* to **Pole** and Create the following variables:

   TotalHeight, *Distance*, *Instance*, *Visible*, *32 feet*

   PoleBaseR, *Distance*, *Instance*, *Visible*, *:10 or 10 inches*

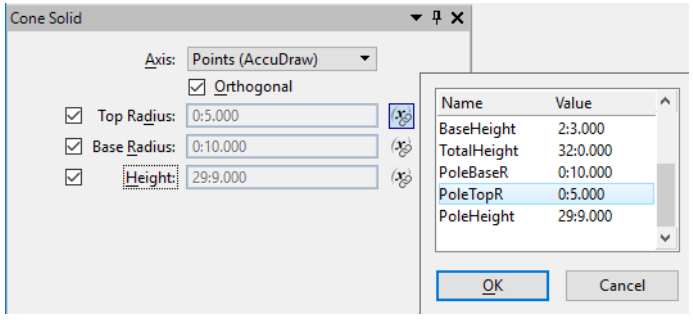
   PoleTopR, *Distance*, *Instance*, *Hidden*, *PoleBaseR / 2*

   PoleHeight, *Distance*, *Instance*, *Hidden*, *TotalHeight - BaseHeight*



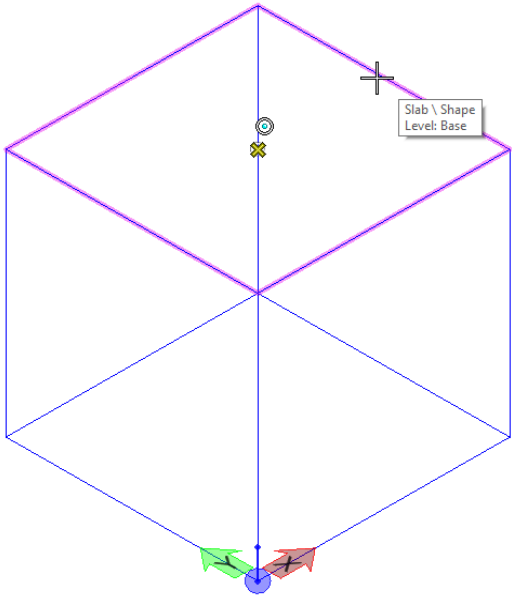
4. *Modeling Workflow>Solids Ribbon Tab>Primitives Ribbon Group>Cone*

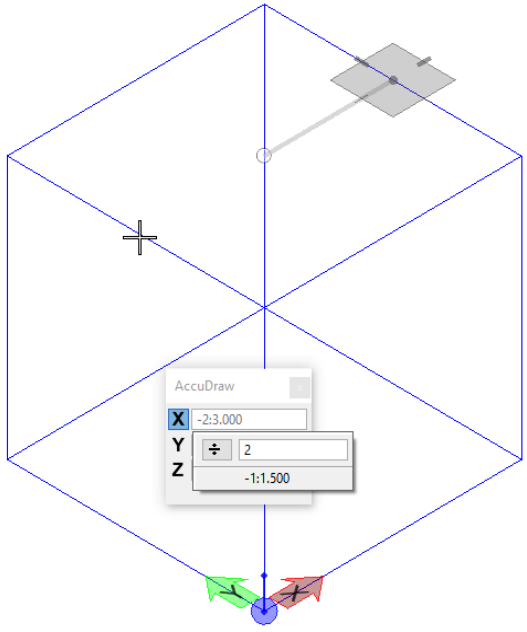
   Set the *Top Radius* to equal **PoleTopR**, *Base Radius* to equal **PoleBaseR** and *Height* to equal **PoleHeight**.

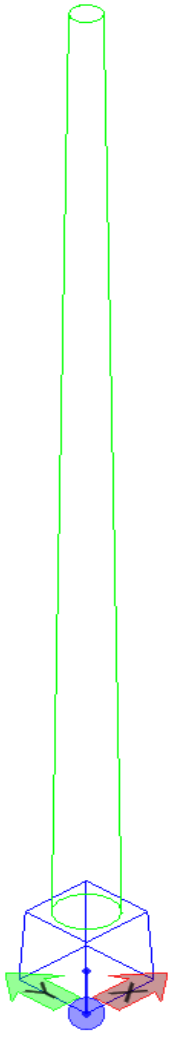


   Using AccuDraw locate the *Center* of the top face of the slab, provide a data point for the base center point, and a direction for the Height.

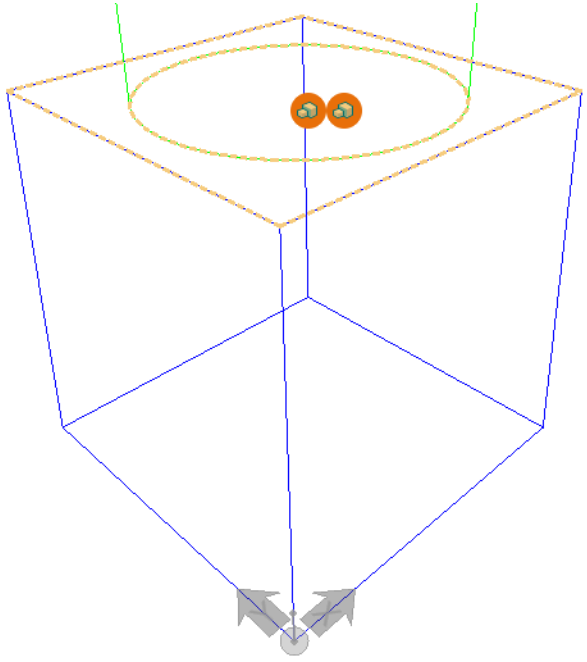
Hint: Use <O> for Origin, and <C> for Center or <T> for Top and divide the distance across the top face by 2.







5. Next, create a 3D Coincident Constraint (**Constraints Ribbon Tab>3D**) between the bottom face of the pole to the top face of the Base.

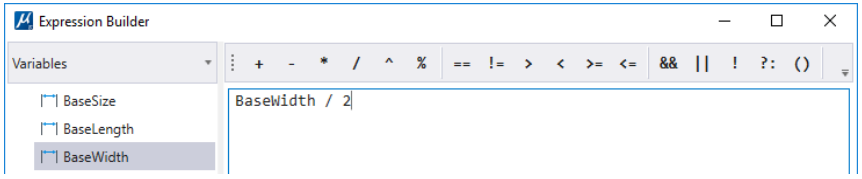


6. Next, create variables that will be used with the 3D Dimension tool. Currently if the Base is moved along the Global Z axis the Pole will adjust accordingly, however if the Pole is moved in the Global X or Y axis, the Pole and Base are disconnected. These dimensional variables will be used to keep the pole centered on the base. It is based on the variables for *BaseLength* and *BaseWidth* and divides them by 2, therefore always keeping the center of the Pole centered on the Base, regardless of the size of the base.

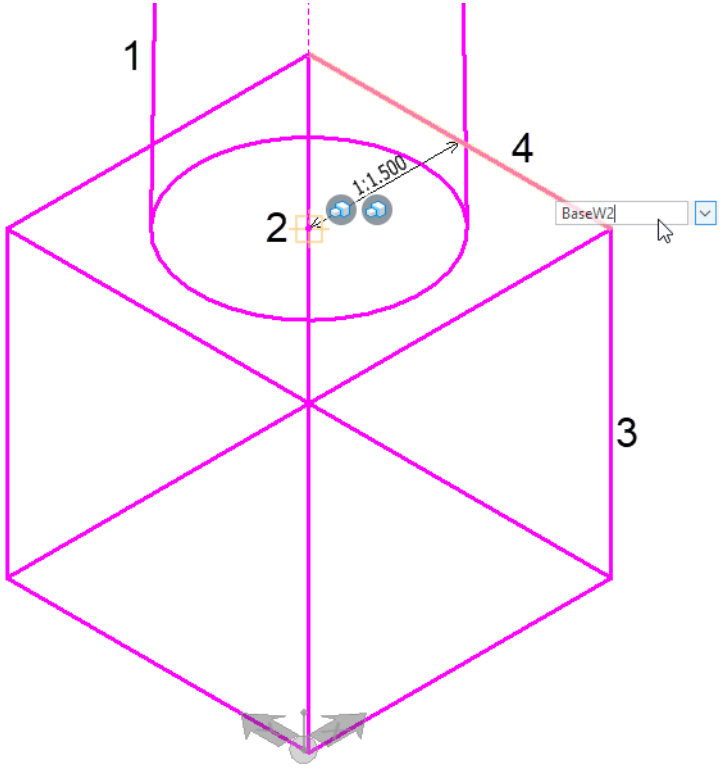
BaseL2, *Distance*, *Instance*, *Hidden*, *BaseLength/2*

BaseW2, *Distance*, *Instance*, *Hidden*, *BaseWidth/2*

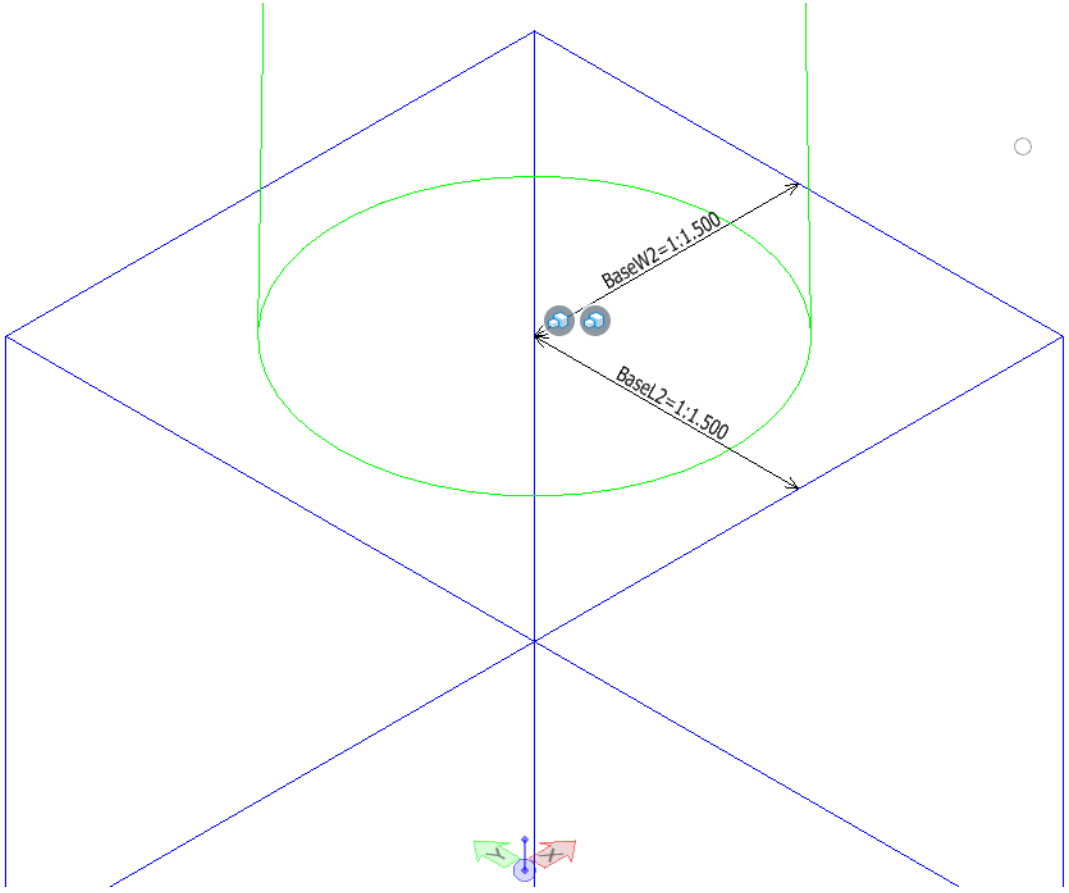
- Pick **Modeling Workflow>Constraints Ribbon Tab>3D>3D Dimension**.



- Pick the Pole followed by the center point of the pole base, followed by the slab and then the right edge (ISO view) this will need to be assigned the variable of *BaseW2* by selecting **BaseW2** from the dropdown.



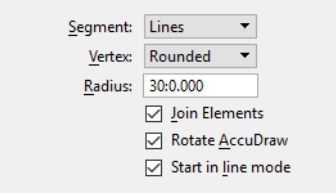
- Repeat by assigning the variable **BaseL2** to the Y direction as shown.

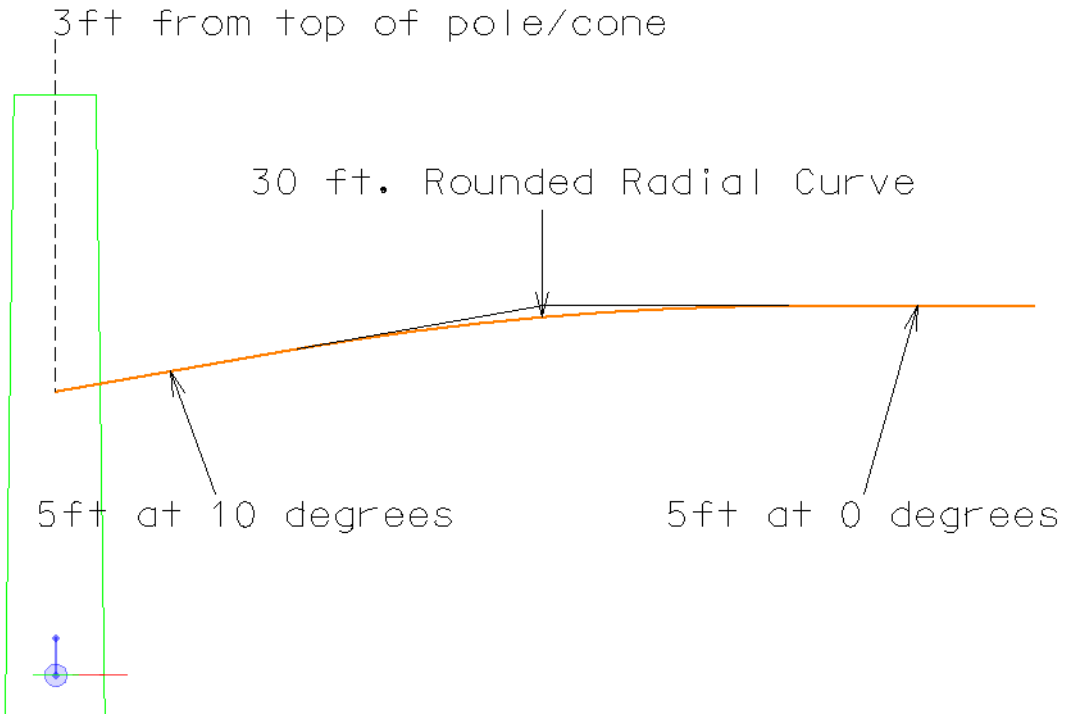


7. Next use **Place Smartline** to draw the centerline for the Light Arm Profile.

   Set **Active Level** to *LightArm* and pick **Place Smartline** from **Modeling>Home Ribbon Tab>Placement Ribbon Group>Place Smartline**

Moving the mouse to the center point of the top face of the Pole, and pick <O> for Origin in AccuDraw. Moving the mouse in a direction down the pole, type 3ft, and a data point at that location to set the location for the start point. The light arm profile is 5’ at 10 degrees, with a 30’ rounded vertex in Smartline, then along the X axis (0 degrees), 5 ft.

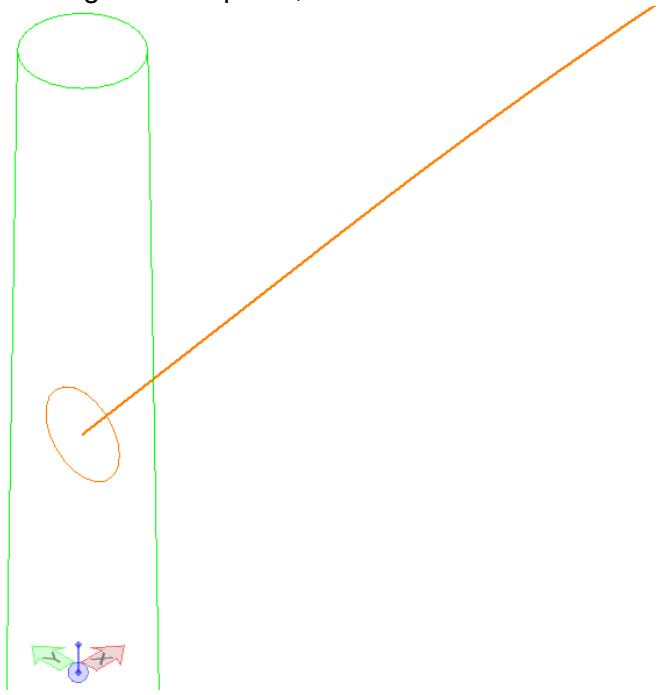




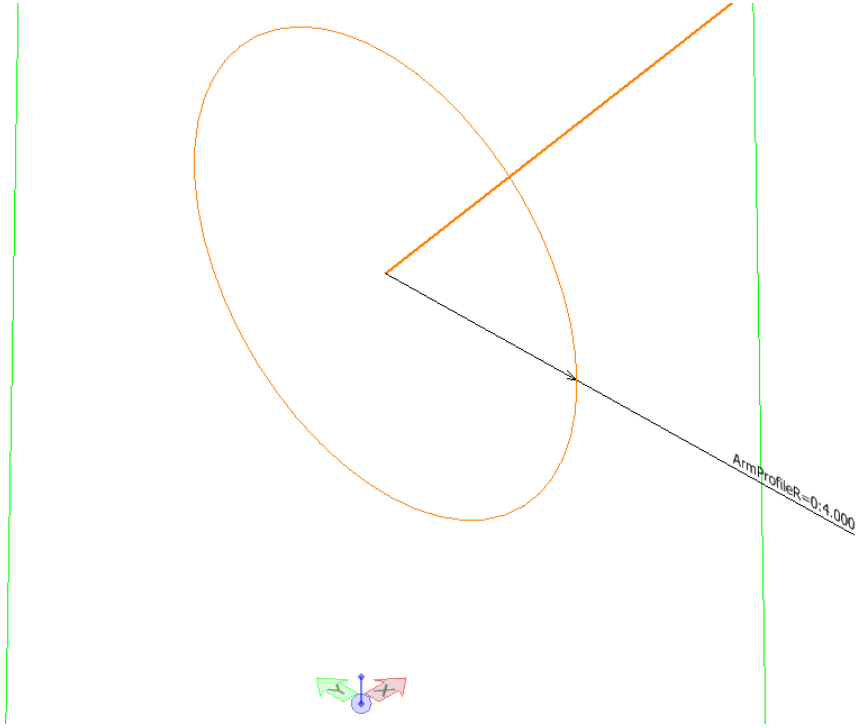
8. Create the following variable: **ArmProfileR**, *Distance*, *Instance*, *Visible*, *:4*.

   Ensure the **Active Level** is set to *LightArm*.

Next using Place Circle draw a circle along the Side plane, with a radius of :4 inches.

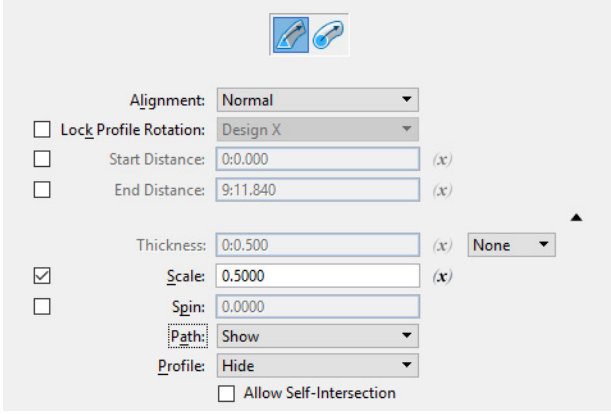


Using **Constraints Ribbon Tab>Dimensional>By Element** , place a dimensional constraint on the circle using the variable *ArmProfileR*.



Note: This is a different type of *Dimensional Constraint* than that used for *BaseW2* and *BaseL2*. Those dimensional constraints were placed on 3D geometry. And while the model is 3D (with the geometry in general also 3D), the circular profile is planar 2D geometry, requiring the use of the correct dimensional constraint.

9. Next extrude the circle using **Solid by Extrusion Along Path**, pick **Modeling Workflow>Solids Ribbon Tab>Create Solids Ribbon Group>Extrude Along.** Set the options as shown (the extrusion will scale down by half over the distance).



Select the path element, then pick the profile, followed by snapping to the beginning of the path element, accepting the extrusion with a data point to create. And note that a variable could have been created and selected for the option of *Scale*. Here it is “hard-coded” to .50.

10. Create the following variable: **ArmProfileHeight**, *Distance*, *Instance*, *Visible*, *3ft*.

11. Earlier a 2D Dimension was added to the Light Arm Profile. Now a 3D Dimensional Constraint will be added to define the Light Arm's location as 3 feet below the top face of the pole. Pick **Modeling Workflow>Constraints Ribbon Tab>3D >3D Dimension**. Pick the first (stationary) element (the pole), and then select the top face of the pole. Next, identify the second (moveable) element (the arm), followed by selecting the center of the starting profile of the extrusion. Complete the 3D constraint by accepting the placement with a data point, assigning the value of **ArmProfileHeight** to it.

TrafficArmHeight, *Distance*, *Instance*, *Hidden*, *Total Height-19*

TrafficArmLength, *Distance*, *Instance*, *Visible*, *45ft*.

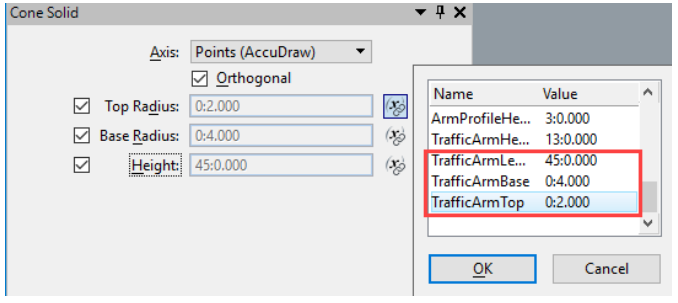
TrafficArmBase, *Distance*, *Instance*, *Visible*, *:4*

TrafficArmTop, *Distance*, *Instance*, *Hidden*, *TrafficArmBase/2*

Set **Active Level** to *TrafficArm*

Next, a Traffic Signal Arm is needed. A Traffic Signal Arm will be added and constrained so that the signal arm will be no more than 19' above the existing signal base, regardless of the overall total height for the traffic signal light.

From the **Modeling Workflow>Solids Ribbon Tab>Primitives Ribbon Group>Cone**.

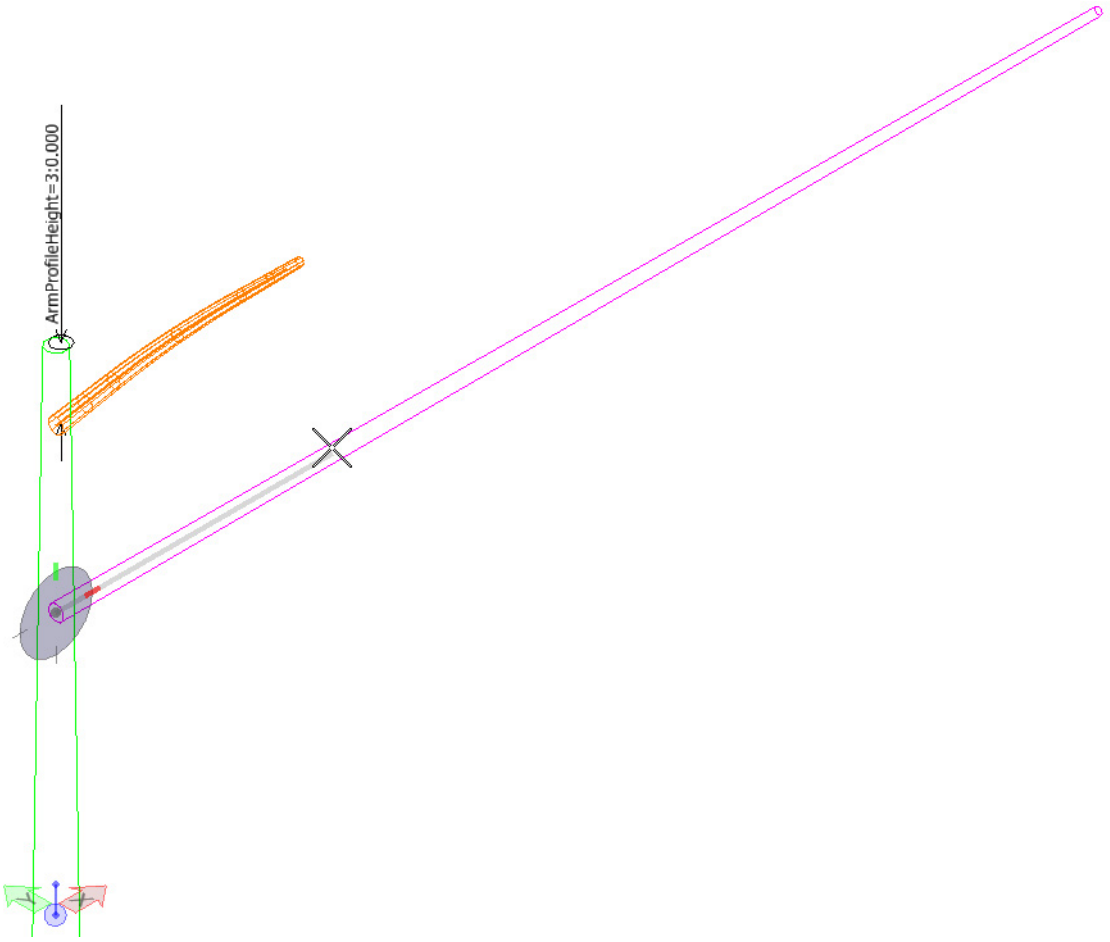


Set the options as shown

- Axis : Points (AccuDraw)
- Orthogonal On
- Top Radius set to **TrafficArmTop**
- Base Radius set to **TrafficArmBase**
- Height set to **TrafficArmLength**

Move the mouse to the center point of the top face of the Pole and using AccuDraw press <O> for Origin followed by <F> for Front. Moving the mouse in a downward direction, type 10ft, and a data point at that location to define the start point of the cone. Moving the mouse in a positive X direction (of the AccuDraw compass), provide a data point to place the Traffic Arm.

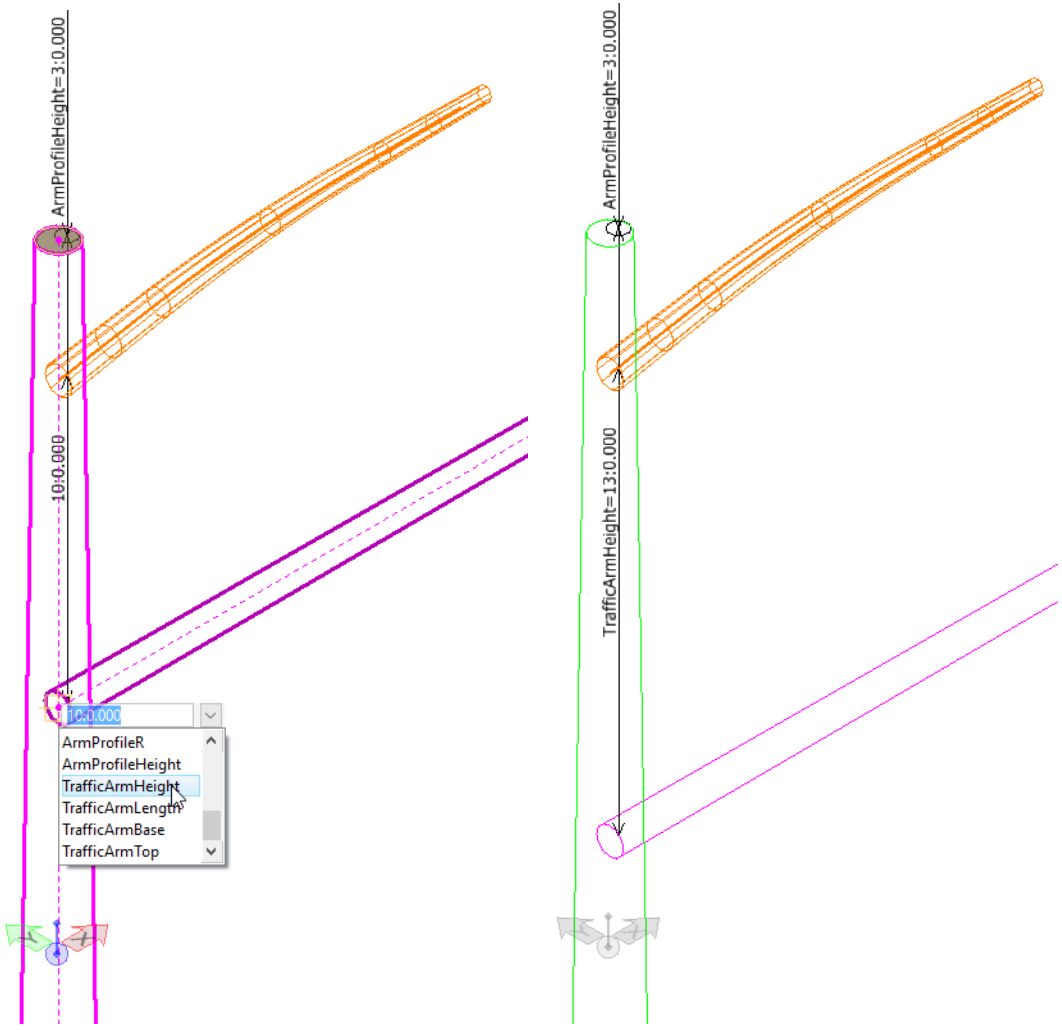
The Cone representing the Traffic Arm is 45 ‘ in length, with a base radius of 4” that tapers to 2”. The Traffic Arm is 10’ below the top face of the pole (currently it is hardcoded at this location), however a 3D Dimension variable will be placed in the coming steps.



12. Note the variable, **TrafficArmHeight**. The value of this variable is calculated by subtracting 19’ from the value of *TotalHeight* (currently set to 32’). This allows us to maintain the height above the existing ground to be maintained (note the active value is set to 13’). The *TrafficArm* is currently positioned at 10’ from the top face of the pole. When the 3D Dimension variable is applied , the *TrafficArm* will adjust.

Next add a 3D Dimension Constraint to define the Traffic Arm's start location at 13 feet below the top face of the pole. Pick **Modeling Workflow>Constraints Ribbon Tab>3D >3D Dimension**. Pick the first (stationary) element, (the pole), then select the top face of the pole.

Next identify the second (moveable) element, (the Traffic arm), followed by the center of the base of the traffic arm. Complete the 3D Constraint by accepting the placement with a data point, assigning the value of *TrafficArmHeight* (13') to it.



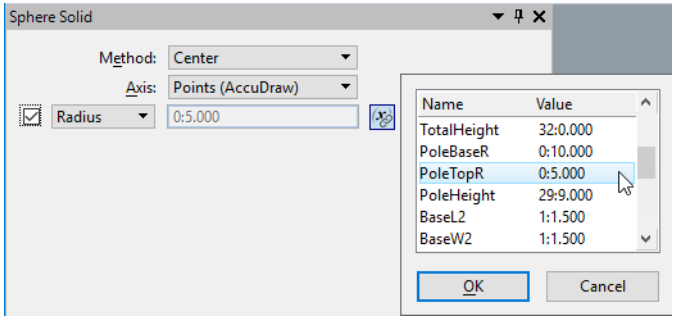
13. More decorative than anything, a "ball" style finial top for the Pole will be added.

    Set **Active Level** to *PoleTop* then review the following variable that was created earlier:

PoleTopR, *Distance*, *Instance*, *Hidden*, *PoleBaseR/2*

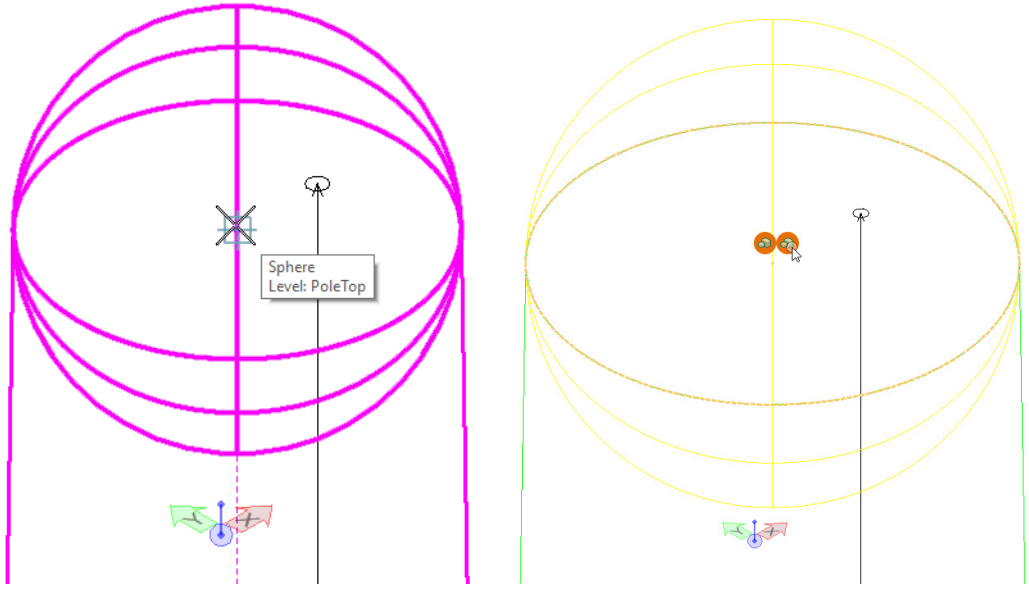
This variable uses the value of the base of the pole, then divides it by 2 to derive the top radius.

Next pick, **Modeling Workflow>Solids Ribbon Tab>Primitives Ribbon Group>Sphere**. Set the options as shown, the most important being set the Radius to the value of the variable *PoleTopR*. Snapping to the center of the top face of the pole and accepting to define the center point. Define the axis of the sphere by picking the edge of the top face of the pole.

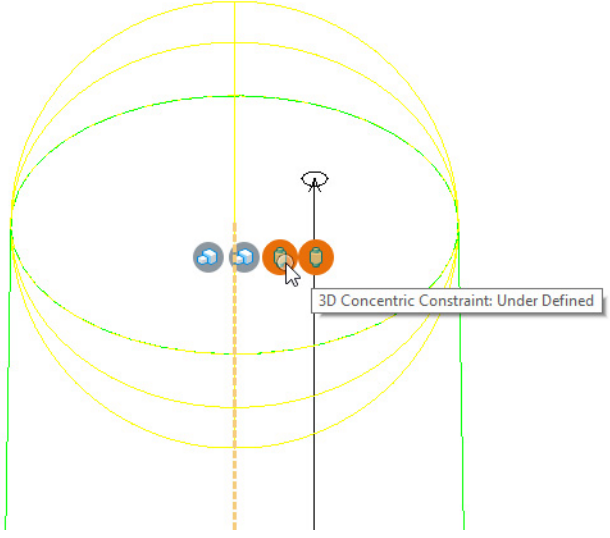


14. If the pole was moved either up or down along its axis, you will note the pole, the base, the traffic arm and light arm all move in concert, however the sphere does not. It will however update its radius based on the size of the pole base, but there are no constraints between the pole and sphere. Pick **Modeling Workflow>Constraints Ribbon Tab>3D Ribbon Group>Coincident**. Pick the first (stationary element)

solid, the pole, then select pick the top face of the pole. Next identify the second (moveable) solid, the sphere, followed by the center of the sphere and accept to place.



15. Next, add a **Concentric Constraint** to the pole and sphere.

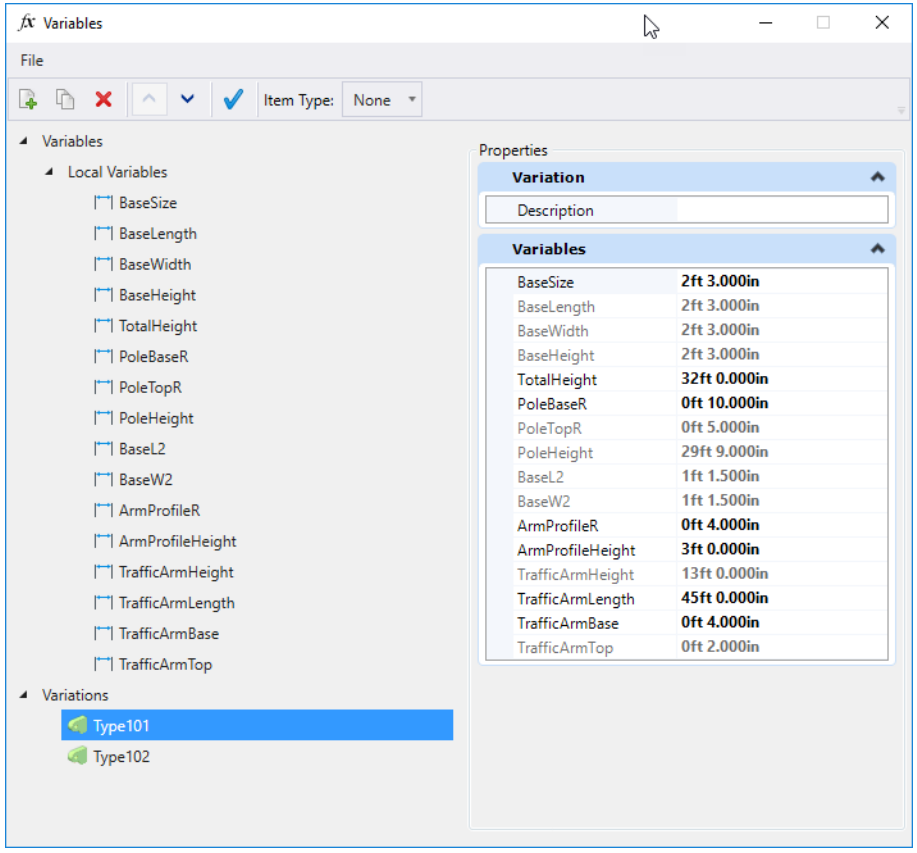


Creating Variants for Use as Parametric Cells

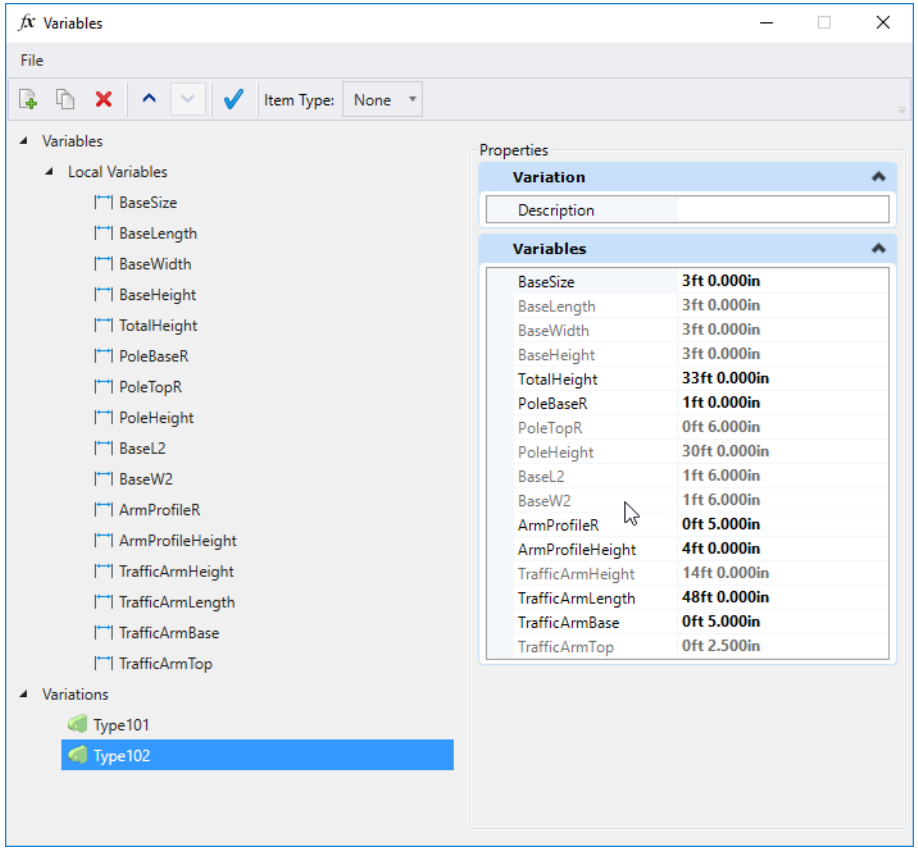
In the last exercise, there are many variables created to define for the Traffic Sign Light Pole. These variable capture our design intent. Rather than adjust all the variables individually to create different instances of a parametric cell, groups of variables can be preserved and named. This allows for easily creating different variants of the geometry and variables.

---

1. Continue in *TrafficSignalLight.dgn*.
2. **Drawing** or **Modeling Workflow>Constraints Ribbon Tab>Dimensional Ribbon Group>Variables**, pick **Variations** and then **New**, name the new variation *Type 101*. Double check the value of the variables shown.

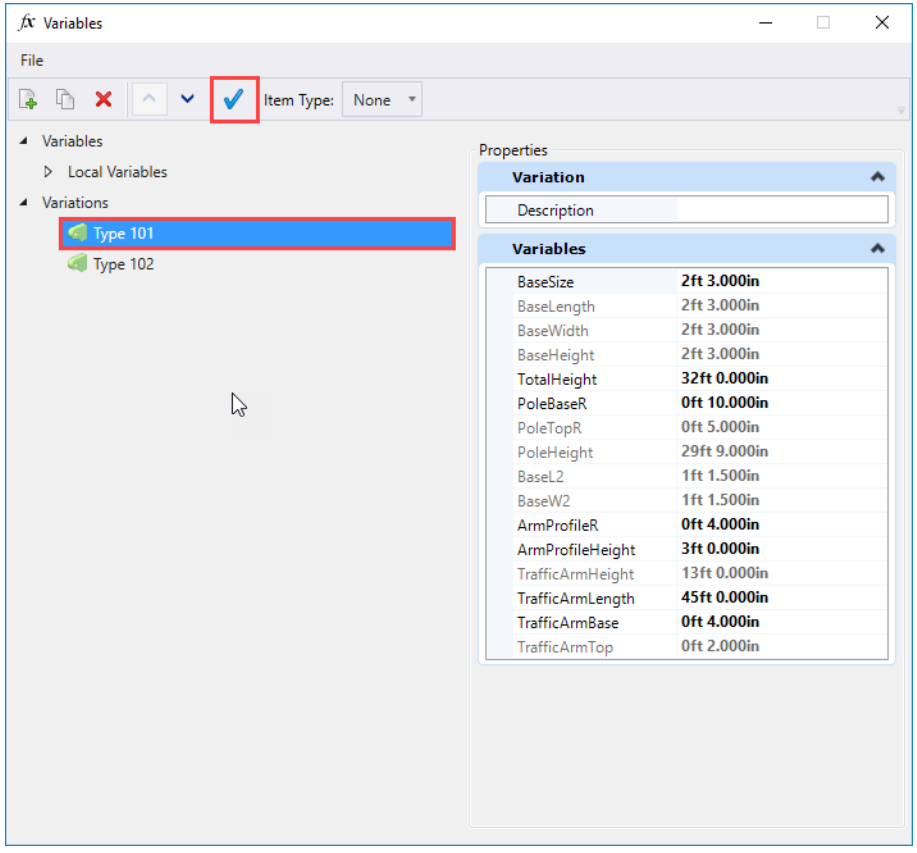


3. From **Variations** and then *New*, name the new variation *Type 102*. Modify the variables as shown.



4. To apply the variations to the model, select **Type 102** and pick **Apply Variables Value to Model** and note the changes to the geometry.

Next **Apply... Type 101** to the model.



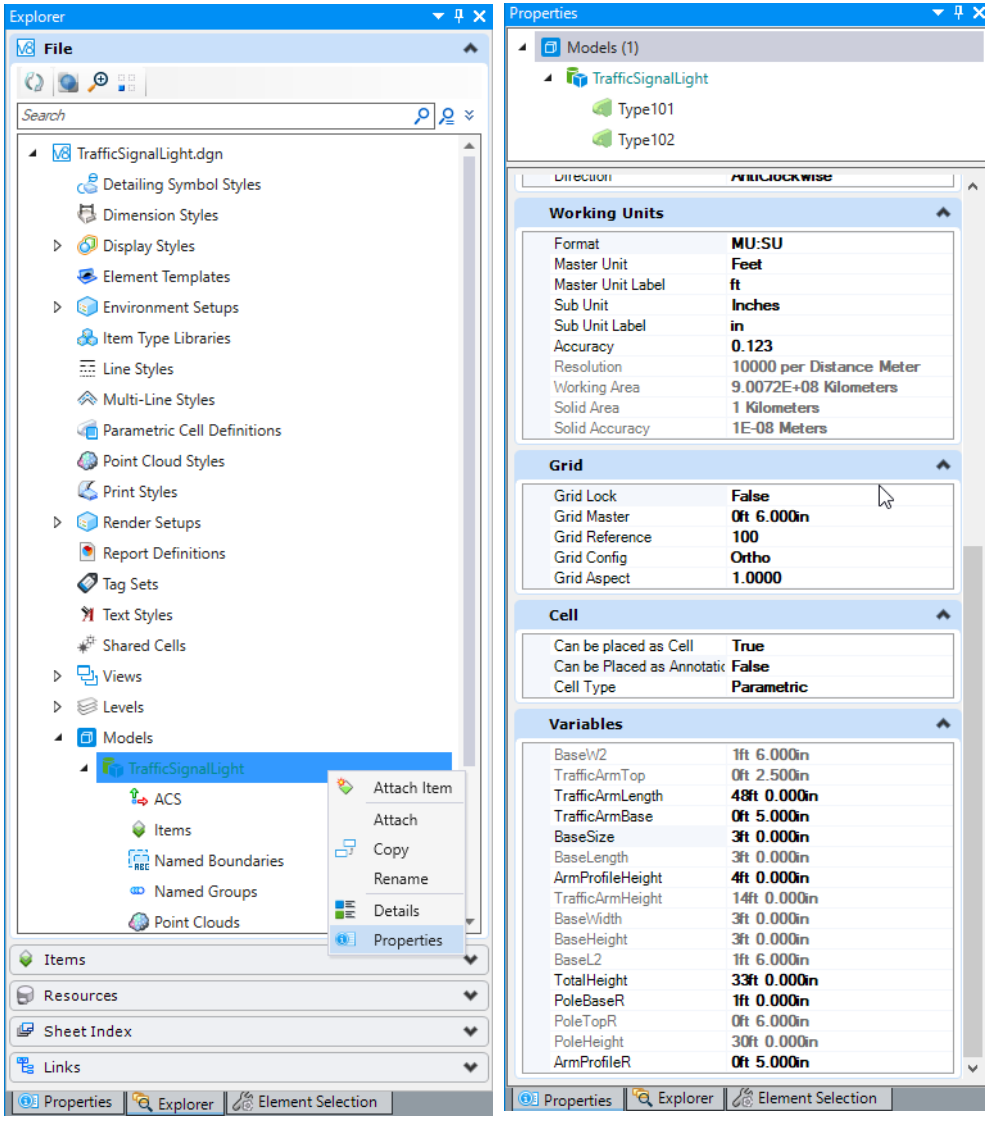
Creating the Parametric Cell

Parametric modeling enables you to reuse your parametric content by placing parametric models as Parametric Cells. Parametric cells are special cells that have been created using variables and equations, grouped in variations, to define their parameters. One parametric cell can have multiple variations, one of which you select when placing the cell.

Parametric Cells are similar to Shared Cells. The first time a parametric model is placed as a cell within a given design file, a local cell definition is created in the design file. All cells placed within that file then refer to that cell definition. This causes parametric cells to be much more efficient in terms of file size as compared to regular cells. To enable the placement of a model as a parametric cell, you must set the model's **Can be placed as Cell** property to *True* and its **Cell Type** to *Parametric*, in the model's Properties dialog.

1. Continuing in **TrafficSignalLight.dgn**, start *Explorer*, and navigate to **Models**. Pick the model called **TrafficSignalLight**. Right press on the Model name and select **Properties**. Ensure that the **Cell Type** is set to *Parametric* and that **Can be placed as Cell** is set to *True*.

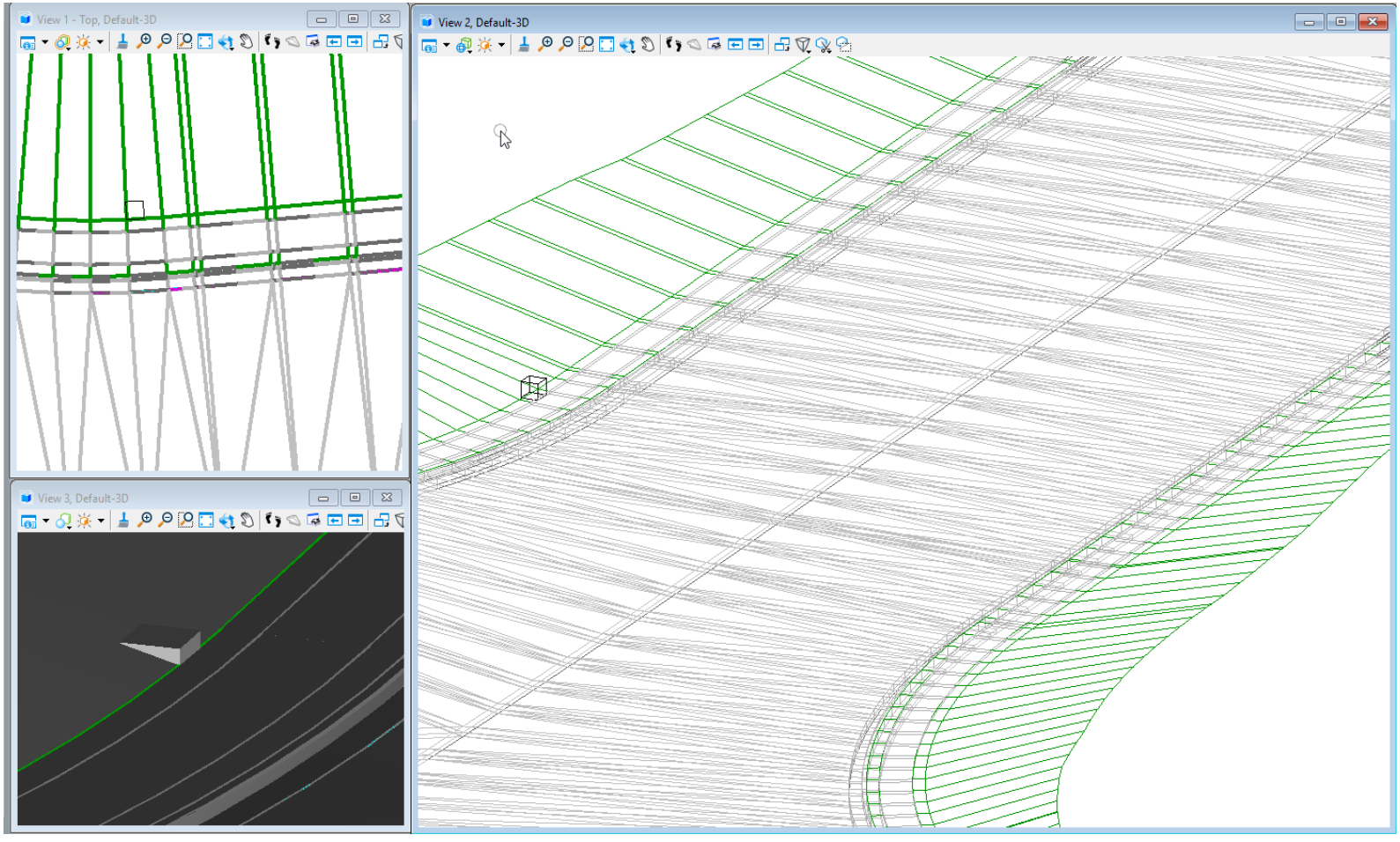
   Since the geometry was created with the base bottom at 0,0,0 and since a Cell is a Model in MicroStation, that model can be placed as a cell.



Placing the Cell for Visualization Purposes

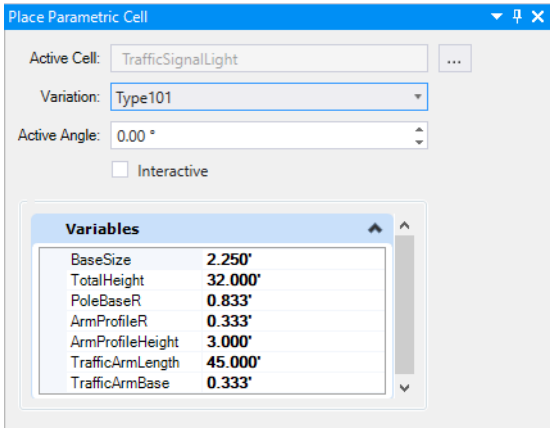
The Traffic Signal Light Pole was created for use in Visualization. A Traffic Signal Light Pole will now be placed on a concrete footing using the variant created during the last exercises.

1. Open the file **Int-Complete.dgn**. This file will be where the Traffic Signal Light Pole will be placed.
2. Setup views similar to as shown, or pick the **View Group**, *Place Signal Light Pole*.

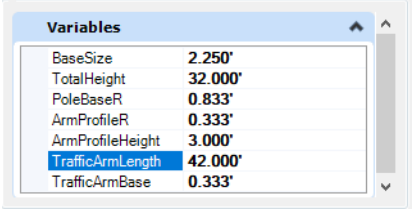


3. Pick the **Drawing Workflow>Home Ribbon Tab>Placement Ribbon Group>Place Active Cell**.

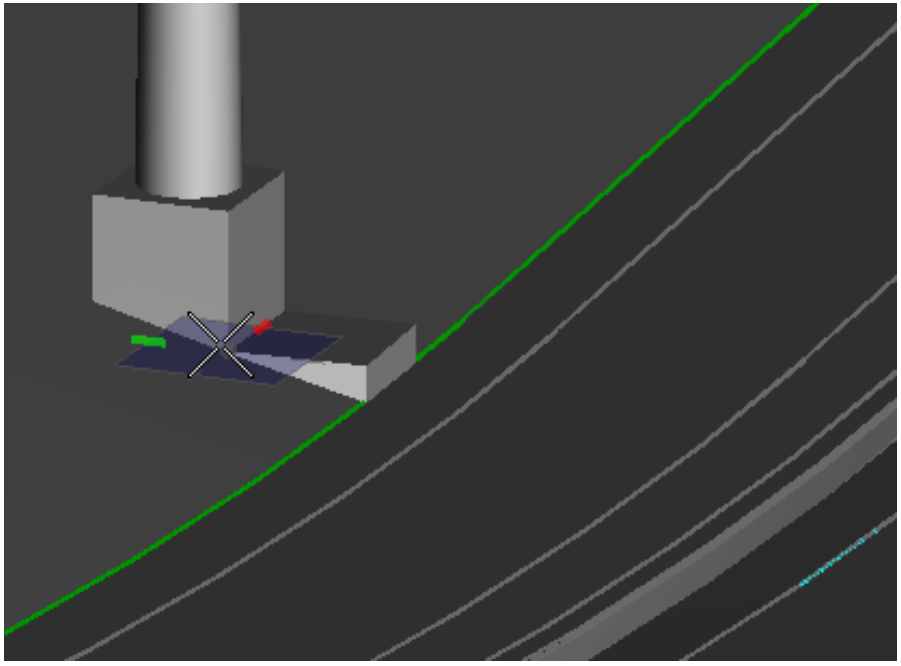
   Pick Browse and navigate to the file (File>Attach File), *TrafficSignalLight.dgn*. Double click the Cell, *TrafficSignalLight*, select the **Variation** *Type 101*.



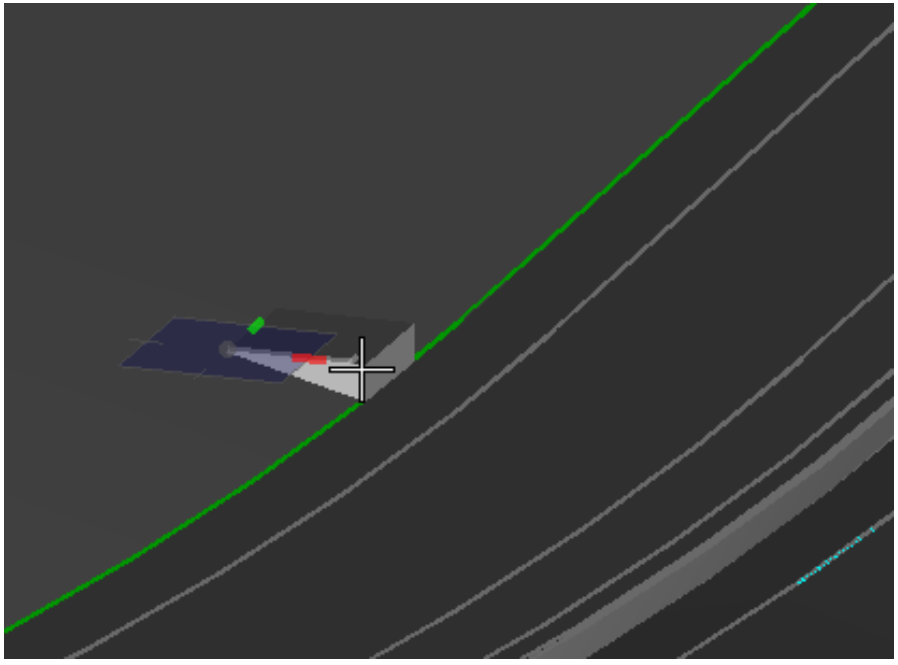
4. Define the variable *TrafficArmLength* to be *42 ft.*



5. Moving the cursor, use AccuDraw to define the **<O>** origin as the back edge of the concrete footing. After the Origin is set, rotate the compass to the Top view using **<T>**.



While snapped (with <O>) to the corner of the concrete footing, use **<RQ>** to rotate the compass relative to the right edge as shown.



Next place the Traffic Signal Light relative to the left, back corner of the concrete footing.