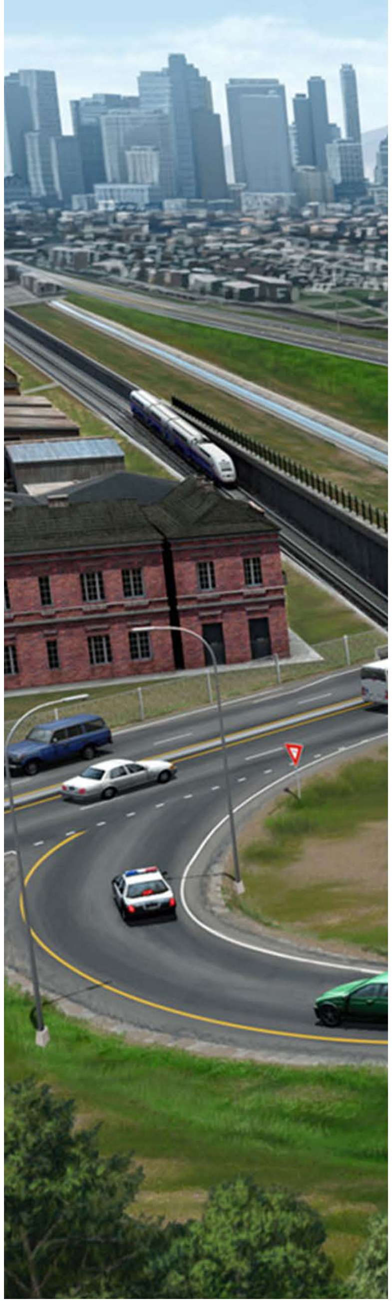


Defining Template Components and Constraints

This course is for the **2024 Release 2** *version of:*

OpenRoads Designer

OpenRail Designer

OpenSite Designer

About this Practice Workbook...

- This workbook is designed for use in Live instructor-led training and for OnDemand self study. OnDemand videos for this course are available on **learning.bentley.com**
- This PDF file includes bookmarks providing an overview of the document. Click on a bookmark to quickly jump to any section in the file.
- Imperial files are included in the dataset.
- This course workbook uses the *Training and Examples* WorkSpace and the *Training-Imperial [Training-Metric]* WorkSet delivered with the software.
- The terms “Left-click”, “Click”, “Select” and “Data” are used interchangeably to represent pressing the left mouse button. The terms “Right-click” and “Reset” are also used interchangeably to represent pressing the right mouse button. If your mouse buttons are assigned differently, such as for left-handed use, you will need to adjust accordingly.

Have a Question? Need Help?

If you have questions while taking this course, you can submit questions to the Civil Design Forum on Bentley Communities where peers and Bentley subject matter experts are available to help.

Bentley Institute

Edition: **09-01**

Course Level: **Intermediate**

Course Overview

This course teaches how to create templates and components for those complex project situations that require more than simply modifying existing templates. This training focuses on the hinge-to-hinge backbone portion of the template. You will learn how to create pavement slabs, pavement stripes, curbs, pavement widening, matching existing pavement and barriers.

Skills Taught

- Create a Pavement Slab Component
- Place Template Points Using Dynamic Settings
- Insert Template Points
- Review and Edit Template Point Properties
- Create Parametric Constraint Labels
- Create a Curb Component
- Define Shoulder Rollover Locks
- Create Vector-Offset Constraints
- Create Pavement Widening Components That Match Existing Pavement Slopes
- Create a Median Barrier
- Create Vertical Minimum and Vertical Maximum Constraints
- Define Component Display Rules.

Exercise 1 - Create a Single Lane Pavement Slab

Description

In this exercise, you will create a single lane pavement slab and define parametric controls so the width and depth of the slab can be dynamically controlled with parametric constraints during modeling.

Skills Taught

- Create Simple Component
- Dynamic Settings
- Assign Point Properties
- Assign Parametric Controls

Template Review

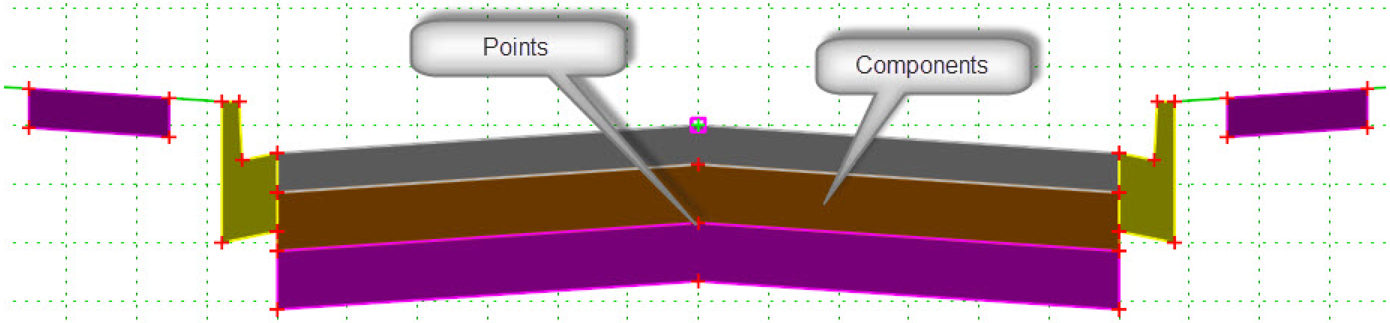
Note: Basic template terminology and techniques are introduced in the *Using and Editing Templates* course.

Typical Sections - a diagram of a slice of a cross section of the finished roadway corridor - are a common roadway project requirement document.

Templates represent this typical cross-sectional geometry. Templates consist of points and open and closed shapes called Components. Points can be “null” (standalone), but most are part of components. They are what components are made of. Rules are used to represent the engineering relationships between points. For example, an Edge of Pavement is typically constrained to a Centerline by a Width and a Slope.

Corridors represent one or more Templates being “dropped” or “extruded” along a 3D baseline to create a 3D model.

- A Point extruded along a 3D path results in a 3D linestring.
- An Open Shape extruded along a 3D path results in a Surface Mesh
- A Closed Shape extruded along a a 3D path results in a Volume



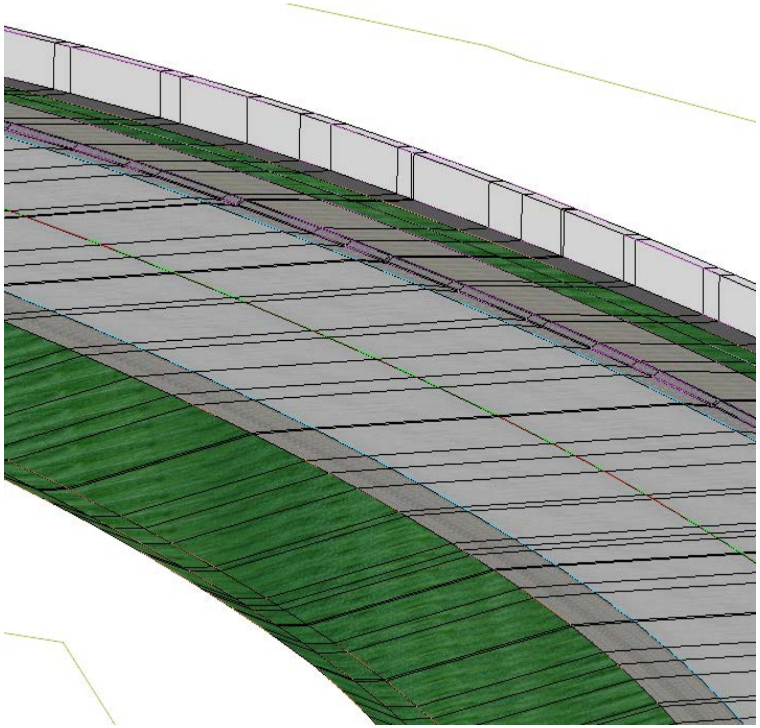
All elements have a *Feature Definition*. The lines and shapes resulting from a “Corridor Run” not only provide an accurate 3D model, their Feature Definitions provide the mechanism for BIM, Quantity Management, and Earthwork Calculations.

Feature Definitions determine a Feature’s Symbology in 2D, Profile, and 3D/Cross Section. This is true for template points and components as well.

Common Closed Components include pavement sections, curbs and sidewalks. Open Components include cut and fill slopes.

“Null points” can be used to model linear features such as utilities that are laid out relative to a baseline or other template point. The resulting 3D linestring can then be “Extracted from Graphics” to create full Subsurface Utilities data modeling.

Templates can be extruded along any 3D linestring using the Apply Linear Template tool. For example, a curb template can be applied around a traffic island. A slope treatment can be extruded along a road corridor hinge point. A pavement lane, curb and slope treatment can be extruded along a pavement saw cut line.



Apply Linear Templates is universal, quick and easy, but it has limitations and makes assumptions. For example, it only applies a single template without providing any mechanisms for adjustments.

Corridors are dramatically more powerful:

- You can “drop” multiple templates along a corridor and control how to transition between them. Consider a corridor that starts without curb and gutter but later needs them. A single corridor can have multiple cross sectional geometries and behaviors.
- Template Points can be controlled by independent geometry.
- Many more aspects of a template can be customized along a corridor. There are multiple *Corridor* courses teaching these capabilities.

Templates are stored in a Template Library File, which has the file extension *.itl*. Like anything else in most data models, no two template points or no two template components in a template can have the same name.

Open a Template Library

In this section, we will open a template library and become familiar with the create template dialog box.

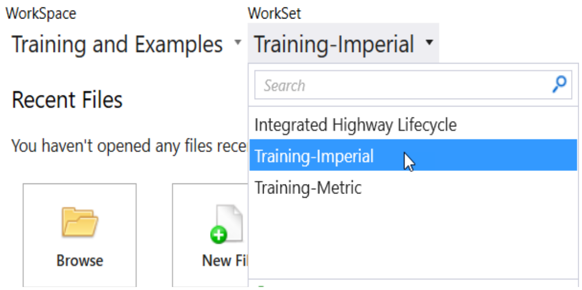
1. Start the software.
2. Set the WorkSpace and WorkSet.

   The WorkSpace and WorkSet define standards that are used by the software, and the ones used for this training are installed during the software installation.

   Typically, the WorkSpace contains organizational standards and the WorkSet contains project standards.

   a. Select **Training and Examples** from the *WorkSpace* drop-down menu.

   b. Select **Training-Imperial** *[Training-Metric]* from the *WorkSet* drop-down menu.



3. Open an existing file.

   a. Select **Browse**.

   b. Browse to *C:\Bentley Training\Defining Template Components and Constraints* or other folder where you unzipped the dataset files

   c. Select the file **Templates.dgn** *[Metric-Templates.dgn]*. and click **Open**.

   Note: If you get a message stating “Incompatible Civil Data”, this is because the training files are “aligned” to OpenSite Designer. Clicking *Yes* will align the file to the software you are using (OpenRoads Designer or OpenRail Designer). This will have zero impact for the training courses. However, note that in production, upgrading the file will make the file read-only in OpenSite Designer.

4. Verify or set the *Workflow* to **OpenRoads Modeling**.

a. Select *Corridors > Create > Template >* **Create Template**.

The *Create Template* dialog opens, displaying the default Template Library. We will be using a different, course-specific library.

b. In the *Create Template* dialog, select *File >* **Open**.

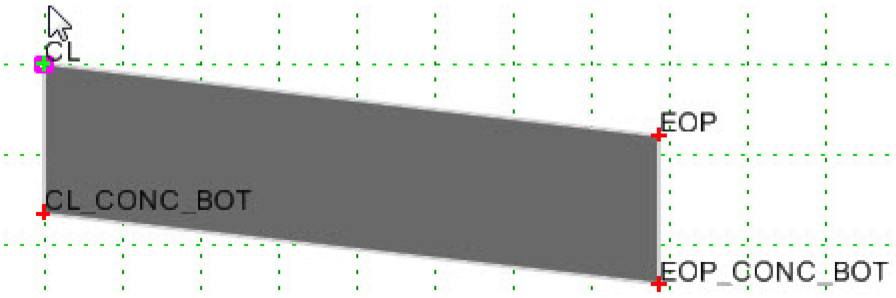
c. *Browse to* and *select* the project template library in the class data folder: **Defining Components.itl** *[Metric-Defining Components.itl]*

Note: When you open the software, the template library designated by the CIVIL_ROADWAY_TEMPLATE_LIBRARY variable is opened, NOT this course template library. If you exit the software, you will have to repeat the above steps to open the course library.

In production, this is typically a WorkSet variable so that each project points to the proper agency or project template library.

Components and Constraints

A *Component* is a set of points that define an open or closed shape. Each component, whether open or closed, can represent a different material or area of interest. In keeping with the "BIM-Ready" OpenRoads data model, the component requires a Name and Feature Definition as properties. The result of extruding a template in 3D as part of a Corridor or Applied Template is a collection of 3D BIM-ready area or volume meshes.

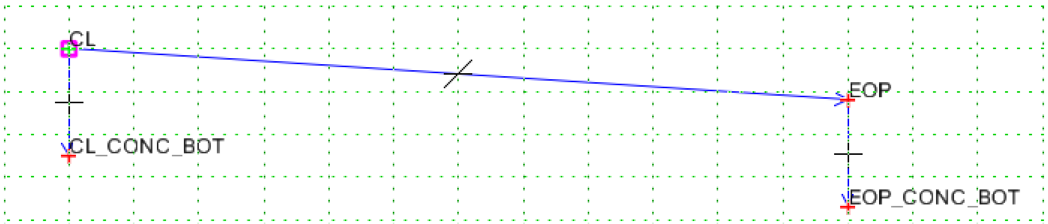


There are several Component types. A Simple Component, like the one we will create in this exercise, is a closed parallelogram defined by a Slope, Thickness, and Width. It is excellent for representing Slabs.

Constraints define the relationship between two points. The Constraint types formalize relationships we use in the real world.

In this example, the EOP point is constrained at a specified Horizontal Distance and Slope from the CL point. The CL point is the Parent of the EOP point. If the CL point moves, the EOP will also move, maintaining the defined Horizontal Distance and Slope.

The CL_CONC_BOT and EOP_CONC_BOT points are also constrained. They are each constrained to the point above them at a fixed Horizontal and Vertical Distance. Notice our naming convention: _BOT is used to indicate "subsurface" points.



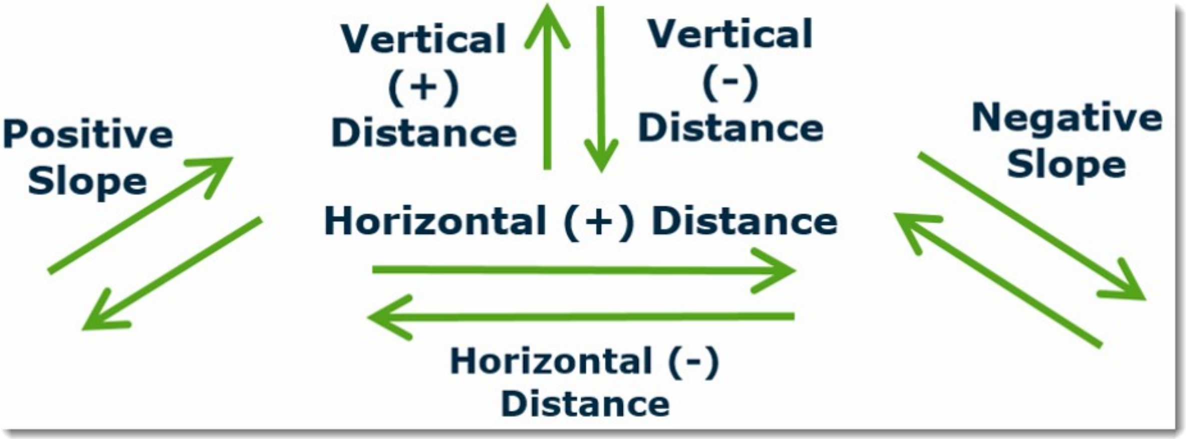
The number of constraints a point has is easily identifiable in the interface by the color of the plus sign that represents the point.

- + Red = Fully constrained point with two constraints
- + Yellow = Partially constrained point with one constraint
- + Green = Unconstrained point with zero constraints

Notice in the image above that the CL is unconstrained (green) and the EOP is fully constrained (red).

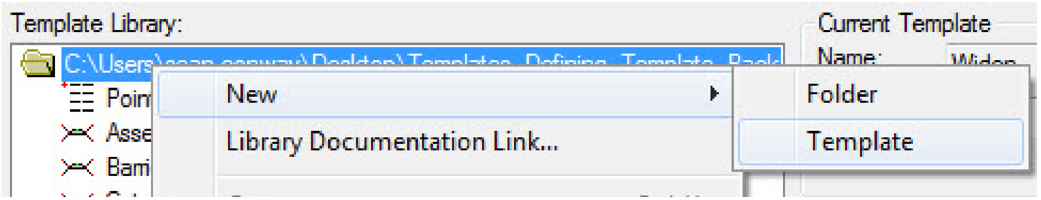
The sign of constraint value is also important.

- The distance is positive if the parent has a lower X or Y value than the child.
- The distance is negative if the parent has a higher X or Y value than the child.
- The sign of the component slope is based on the mathematical slope.

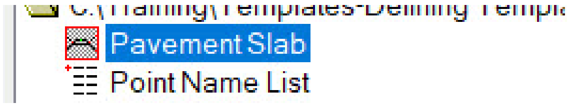


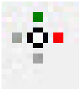
Create a Pavement Slab

1. *Right-click* on the template library object, select *New* > **Template**.

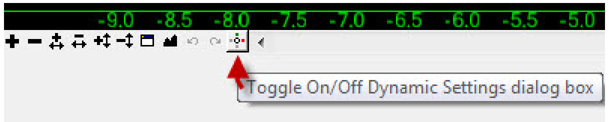


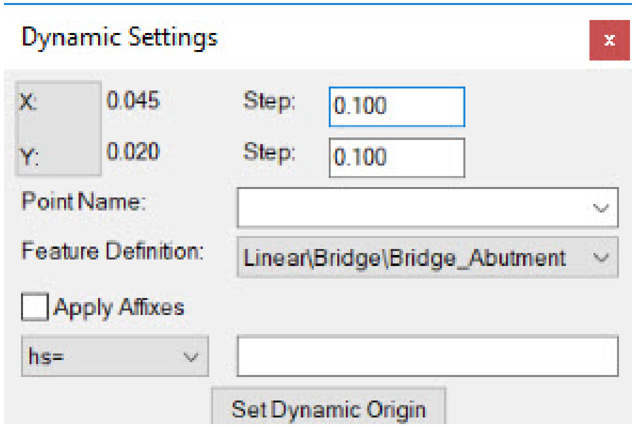
   a. Name the new template **Pavement Slab**.





   b. Open the **Dynamic Settings** dialog. This can be done from the *Tools* > **Dynamic Settings** menu or from the Dynamic Settings icon in the Edit Template view toolbar.





   c. Clear the **Apply Affixes** check box.

   d. Set both of the *X* and *Y Step* settings to **0.1** *[0.01]*.

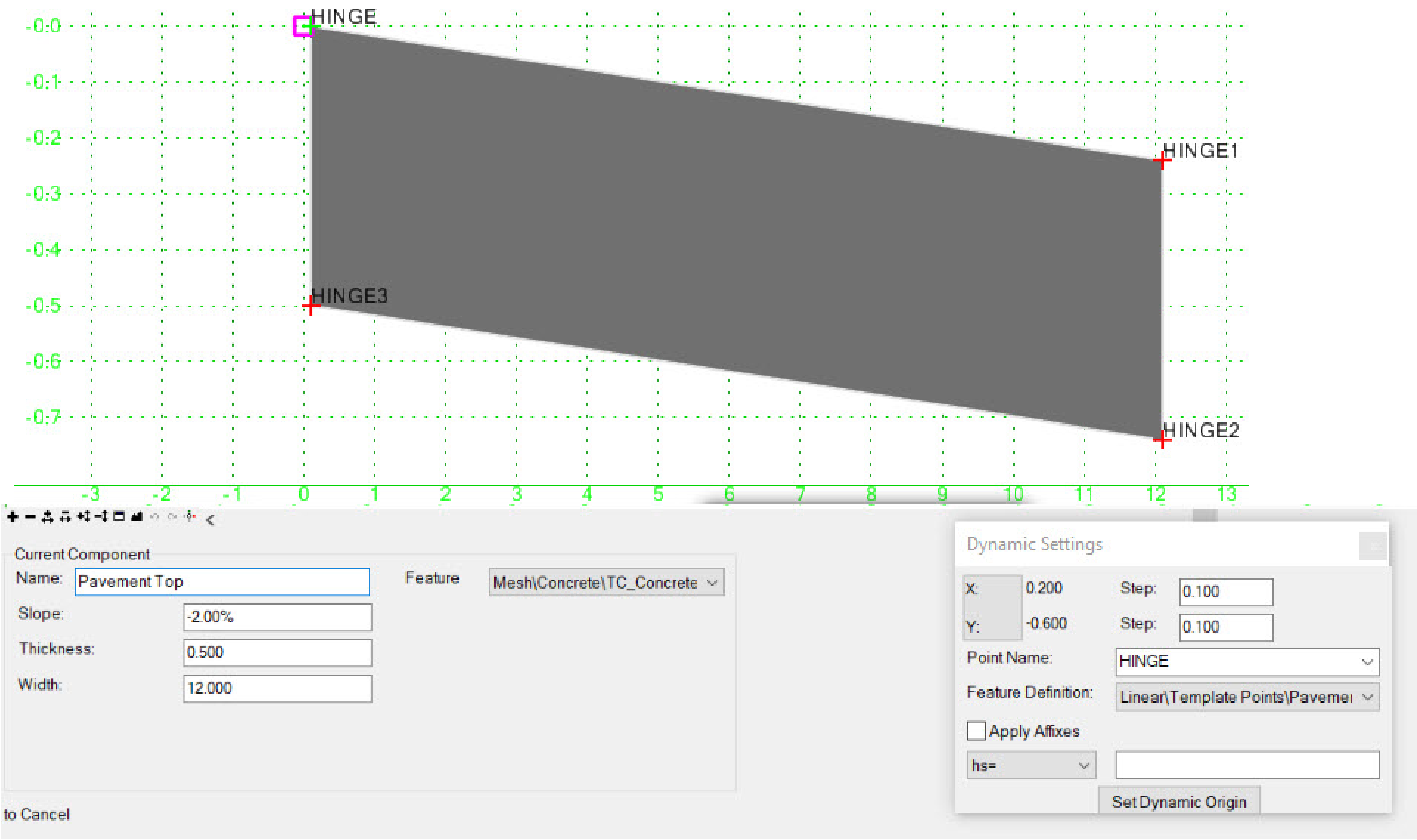
   The Step settings define a snap grid in the template dialog making it easier to precisely locate points and components.

   e. Pick a **Name** from the list. This will be used as a “seed” name during placement: <name>, <name>1, <name>2, etc.

   What you pick here is not critical as we will change them all directly after placement. It’s Good Practice, however, to not leave Feature Names blank.

   f. *Right-click* in the Edit Template window, click *Add New Component* > **Simple**.

A slab or parallelogram is drawn from the cursor at the Slope, Thickness, and Width shown in the Current Components frame. Point properties are based on the Dynamic Settings. We will manage the points after setting the Component Properties.



g. Define the following values for the Current Component at the bottom of the Create Template dialog:

- *Name*: **Pavement Top**
- *Feature*: **Mesh\Concrete\TC_Concrete Pavt**

- *Slope*: **-2.0%**
- *Thickness*: **0.5** *[0.15]*
- *Width*: **12** *[3.6]*

h. Place the Pavement at the **0, 0** point.

HINT: Use the feedback in the Dynamic Settings window to verify you are at the 0,0 point.

Note: If the point is not exactly at 0,0, you can right click on the point and select Change Template Origin.

i. **Fit** the template display if necessary to view the entire template component.

Assign Useful Properties to the Points

Using clear, intuitive point names is a recommended Best Practice. The points in the template will become linear elements in the design model. They can be used for display, calculations, and reporting. Using understandable names for the template points will result in 3D Features with understandable names. Point names can be typed or selected from the predefined list. Selecting a name from the predefined Point Name List increases conformance to standards, minimizes typing errors and accidental deviations, and automatically assigns the correct Feature Definition.

The Point Name List is stored in the template library. Names can be added to the list by right-clicking on the Point Name List object in the Library tree view and selecting Edit.

---

1. *Double-click* on the upper left point at the 0,0 origin.

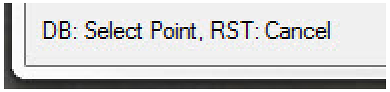
   a. In the *Point Properties* dialog, select **CL** for the *Name*. Do **not** type in CL, select it from the point pick list.

   Note that the *Feature Definition* is automatically set to *TL_Centerline*. The points in the Point List pull down have a Feature Definition assigned to them. Selecting Points from the list automatically updates the Point Feature Definition.

   b. *Click* **Apply**.

2. *Click* the **Target / Graphic Selector** button next to the Name List.

   The dialog temporarily disappears to facilitate selecting a point. The lower left corner of the Create Template dialog prompts to select a point or to right-click to return to the dialog.



   a. Select the point below the **CL** point.

   The *Point Properties* dialog returns.

   b. For the point Name select **CL_CONC_BOT**.

   c. *Click* **Apply**.

3. *Click* the **Target / Graphic Selector** button next to the Name List.

   a. Select the upper right point.

   The *Point Properties* dialog returns.

b. For the point Name select **EOP**.

c. *Click* **Apply**.

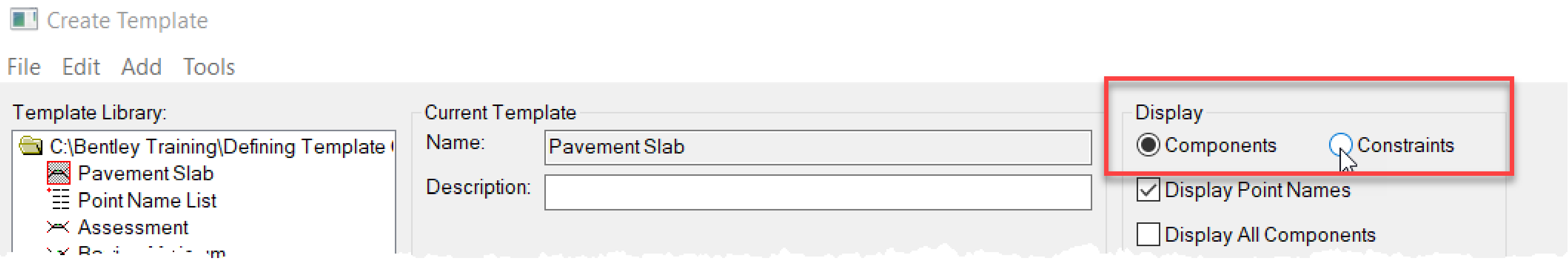
4. *Click* the **Next >** button. This is another way to navigate through the points on this component. Previously we used the Target/Graphic Selector.

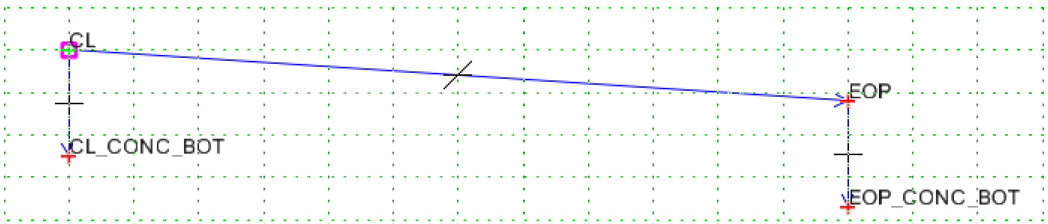
d. For the last point Name select **EOP_CONC_BOT**.

e. *Click* **Apply** and then **Close**.

Review the Constraints Graphically

5. *Click* the **Constraints** *Display* option in the upper center of the *Create Template* dialog.





This display shows the constraint relationships between the points.

The blue lines show the parent-child relationships with the arrow going from the parent to the child point. The black graphics near the middle of the blue lines indicate the types of constraints defined on the child point. A horizontal line indicates a horizontal constraint, a sloped line a sloped constraint, etc. Understanding these graphics one can quickly tell that the *CL_CONC_BOT* point is constrained from the *CL* point using a horizontal and a vertical constraint.

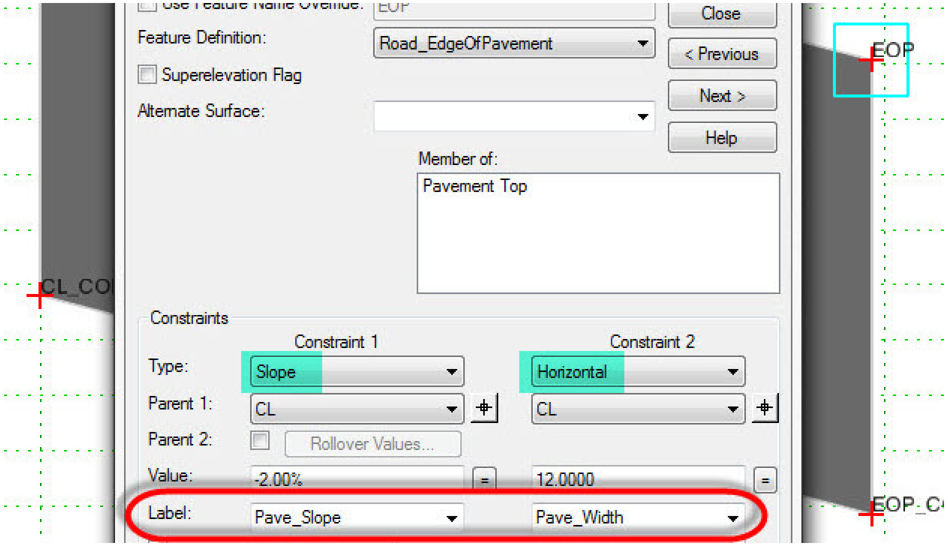
a. Return the *Display* to the **Components** option.

b. **Save** the *Template Library*

Add Parametric Labels

Constraints can have Parametric Labels which work like a variable definition so that the value of labeled constraints may be changed when the template is used. For example, the template may have a constraint label named Pave_Width that controls the width of the asphalt wearing surface. This width can change so that one template can be used for different asphalt widths, without having to edit the template. The width can even vary along the length of the design corridor.

1. *Double-click* on the **EOP** Point to edit its *Point Properties*.
   a. For the *Slope Constraint Label*, type **Pave_Slope**.
   b. For the *Horizontal Constraint Label*, type **Pave_Width**.
   c. *Click* **Apply**.



2. *Click* the **Target / Graphic Selector** button next to the Name List.

   a. *Click* the **CL_CONC_BOT** point.

   b. For the *Vertical Constraint Label*, type **Pave_Thick**.

   c. *Click* **Apply**.

3. *Click* the **Target / Graphic Selector** button next to the Name List.

   a. *Click* the **EOP_CONC_BOT** point.

   b. For the *Vertical Constraint Label*, select **Pave_Thick** from the list. Do not type the label this time. Selecting it from the list ensures it is exactly the same label you assigned to the CL_CONC_BOT point so the single label will control both points.

      NOTE: In the future exercises, we will generally skip adding parametric constraint labels solely to save time during this training. However, it is a best practice to label any points you might have a need to control dynamically each time a template is used or along the length of a corridor.

   c. *Click* **Apply**. You must click the *Apply* button to save all changes.

   d. **Close** the *Point Properties* dialog.

   e. **Save** the *Template Library*

Exercise 2 - Create a Two Lane Pavement Slab

Description

In this exercise, you will learn how to create a multi-lane pavement slab with a point at the location of the lane line in addition to the centerline and edge of pavement.

Skills Taught

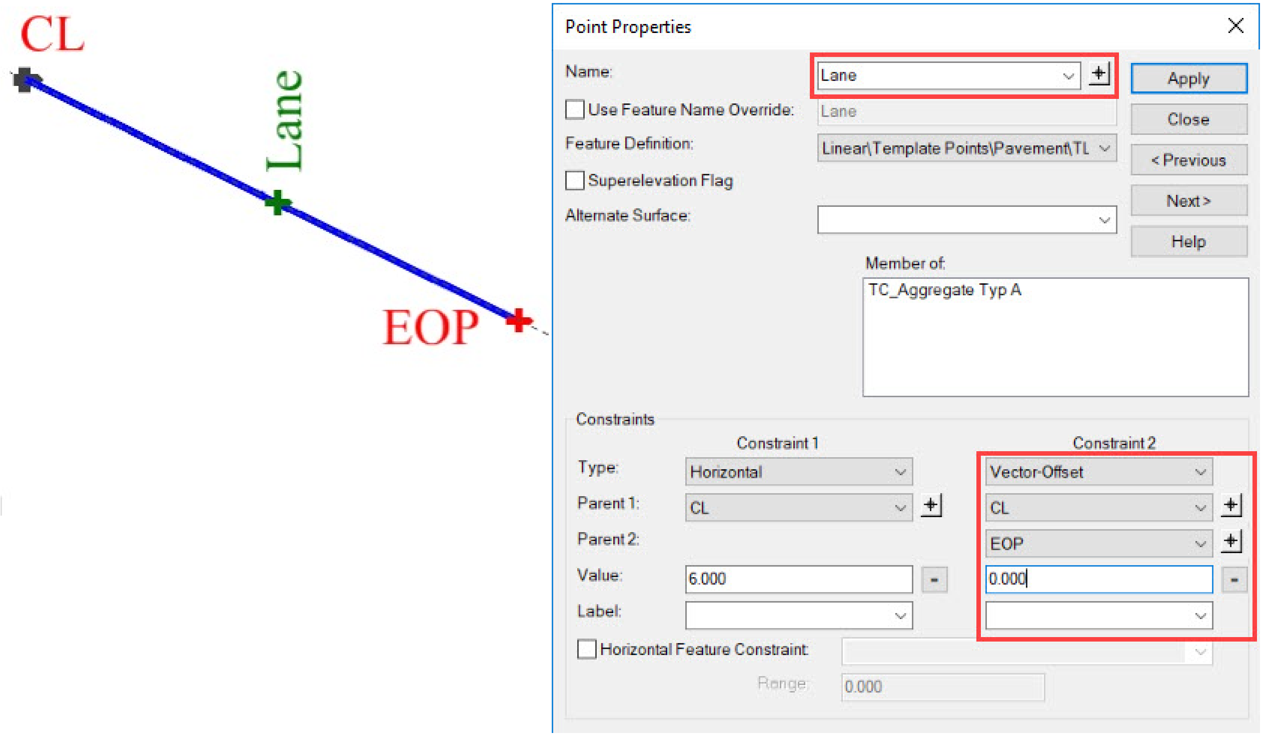
- Copy an existing template component
- Add point into template component
- Test template component
- Add Vector-Offset Constraint

Vector-Offset Constraint

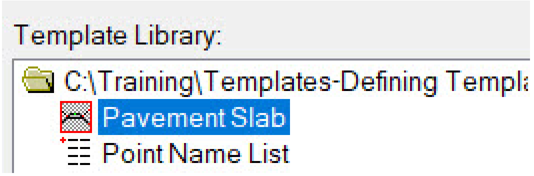
A common geometric specification for a point (i.e. Lane point) is to lie along a slope defined by two other points. For example, a lane striping point typically lies on the pavement defined by the edge of pavement (EOP) and the centerline (CL). Another example is a widened edge of pavement where the slope is required to match the slope of existing pavement.

The Vector-Offset constraint is used to define a Vector as a directional line through two points. An Offset value is specified perpendicular to the vector, positive being above the line between the first and second points, negative being below the first and second points. The vector's direction is defined from the first to the second point.

In this example (see image), the new point labeled **Lane** is created along the directional vector line between the CL and the EOP points at a zero offset.



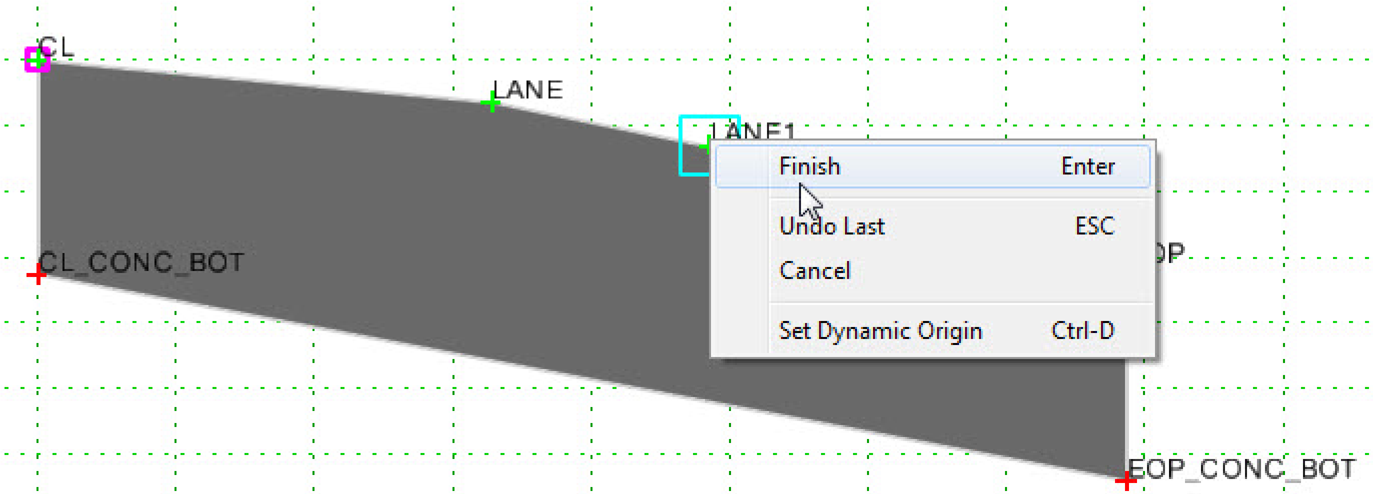
Create a Two Lane Pavement Slab



1. In the template library, *right-click* on the template **Pavement Slab** and click **Copy**.
    a. *Right-click* on it again and click **Paste**.
    b. *Double-click* the new template name to make the template Active
    c. Rename it to **Pavement - 2 Lane**.
    d. *Double-click* the **EOP** point.
    e. Change the *Horizontal Constraint Value* to **24** *[7.2]*.
    f. **Apply** the change and **Close** the *Point Properties* dialog.
    g. **Fit** the view to see the entire template.
    h. **Save** the *Template Library*

Insert a Lane Line Point (for striping)

1. *Right-click* on the line between the *CL* and the *EOP* and click **Insert Point**.
   a. Place the point near the midpoint of the top of the component (no need to be precise).
   b. *Right-click* and click **Finish**.



Next, the location of the **Lane** point needs to be constrained at the proper distance and to be in-line with the CL and EOP.

2. *Double-click* on the new lane point to open the Point Properties dialog.

   a. Set the *Name* to **LANE** by *selecting* LANE from the drop down list.

   b. Define Constraint 1. This constraint will be a horizontal distance from the centerline.

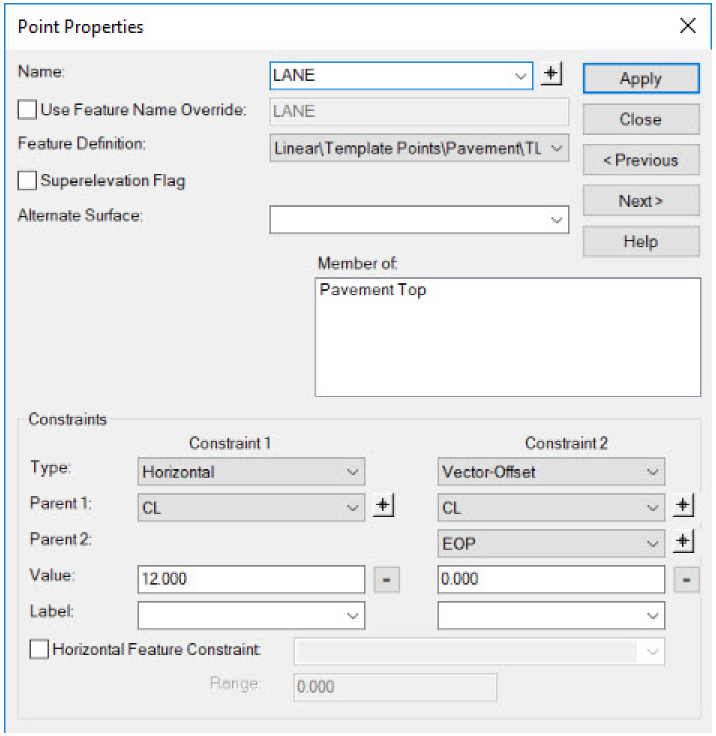
      - Set *Constraint 1 Type* to **Horizontal**.
      - Set *Constraint 1 Parent 1* to **CL**.
      - Set *Constraint 1 Value* to **12** *[3.6]*.

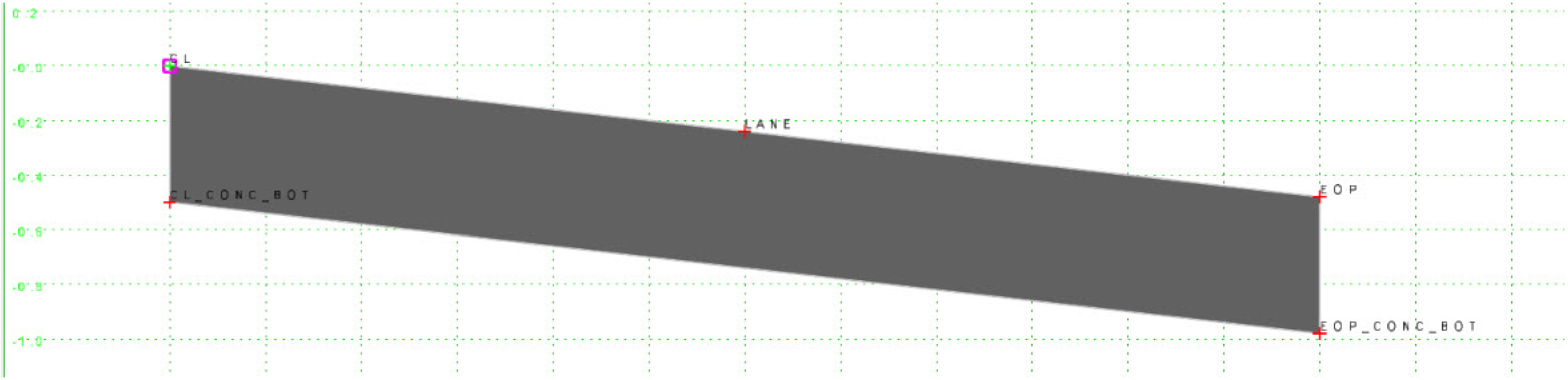
   c. Define Constraint 2. This constraint will keep the lane point on a vector between the centerline and edge of pavement.

      - Set *Constraint 2 Type* to **Vector-Offset**.
      - Set *Constraint 2 Parent 1* to **CL**.
      - Set *Constraint 2 Parent 2* to **EOP**.
      - Set *Constraint 2 Value* to **0**.

   d. *Click* **Apply**.

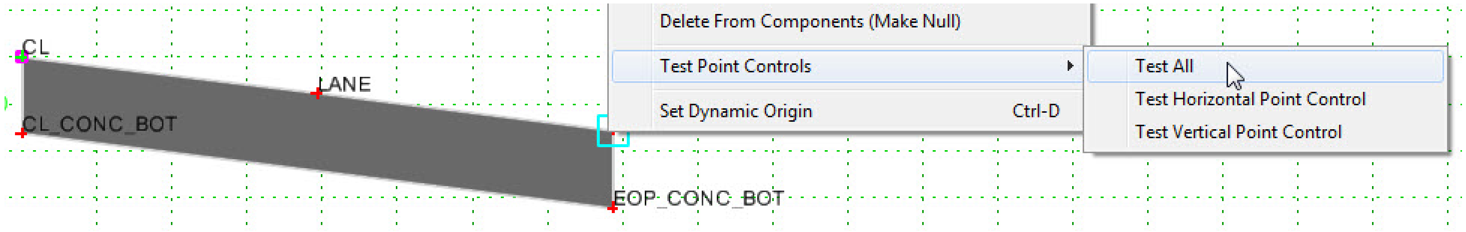
   e. **Close** the Point Properties dialog.





Test the Lane

1. *Right-click* on the **EOP** point, click *Test Point Controls* **> Test All**.



   a. Move the point around. Note the position of the point named *LANE* relative to the other points.

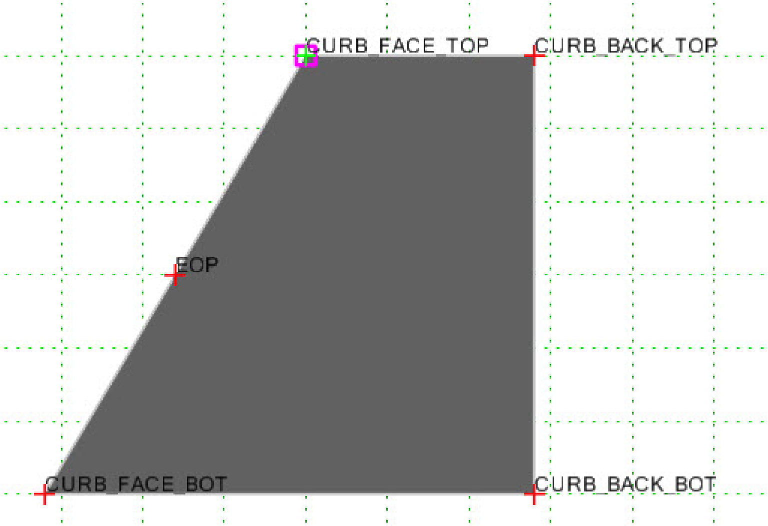
   b. *Left-click* when done.

   c. **Save** the *Template Library*

Exercise 3 - Create a Barrier Curb

Description

This exercise uses precision input to create the barrier curb shown below.



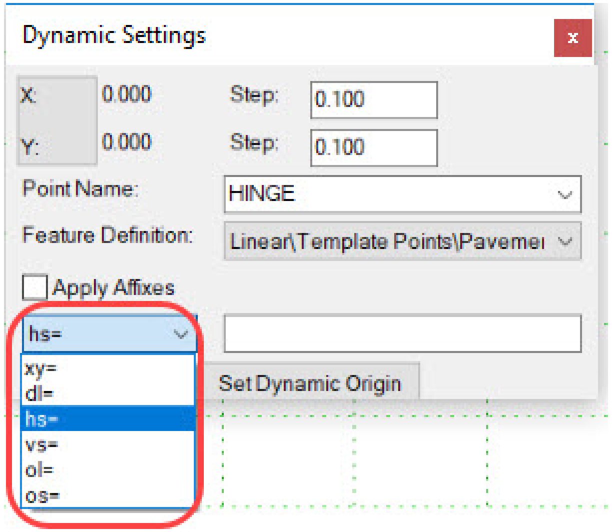
Skills Taught

- Use precision input to define geometry of template component
- Add Vertical Constraint
- Add Vector-Offset Constraint

Precision Input

When creating complex components the precision input capabilities on the Dynamic Settings dialog can be very useful to precisely locate points.

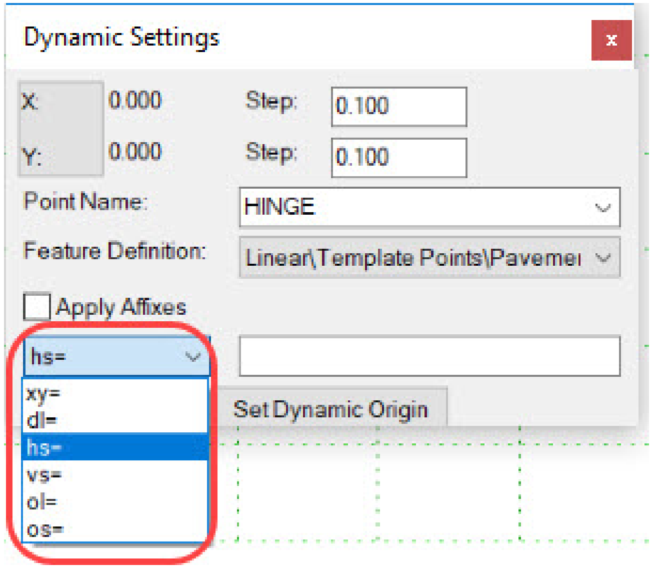
> **XY** = Absolute coordinates from template origin (0,0)
>
> **DL** = Delta coordinates from last point placed (delta X, delta Y) or the dynamic origin if it is the first point of a component.
>
> **HS** = Horizontal distance and slope from last point placed
>
> **VS** = Vertical distance and slope from last point placed
>
> **OL** = Delta coordinates from dynamic origin
>
> **OS** = Horizontal distance and slope from dynamic origin



Create a Curb

1. *Right-click* on the template library object and select *New* > **Template**.
2. Name the new template **Curb**.

   We will build a curb component with constraints. The constraints types will be defined by the Precision Input type selected:



   a. *Right-click* on the Current Template window, then select *Add New Component* > **Constrained**.

      Note: Watch the prompts in the lower left corner of the Create Template dialog.

   b. Set the properties *Current Component* as follows:

      - *Name*: **Curb**
      - *Feature*: **Mesh\Concrete\TC_Curb**

   c. Set the *Dynamic Settings* dialog as follows:

   d. *Point Name*: **CURB_FACE_TOP** (Select from the list).

Selecting the point name here causes the point to be named properly and use the proper feature definition when it is created.

e. *Apply Affixes*: Cleared

f. *Precision Input*: **xy = 0,0**.

g. Press **Enter** key.

A point named CURB_FACE_TOP is placed at the origin (0,0).

HINT: You can type in *xy=0,0* or pick *xy=* from the pick list and type in just *0,0*.

TIP: Precision input values are entered in master units such as feet or meters. You can enter other units with the proper unit specification, such as 6” instead of 0.5. Note that slopes tend to parse the input to percentage.

TIP: When creating a component using precision key-in, ESC allows you to Undo the previous point while staying in the command. You can also right-click in the view and click Undo Last.

3. Set the *Dynamic Settings* dialog as follows:

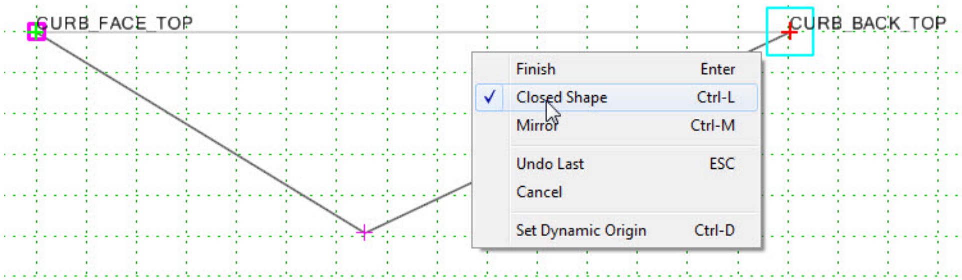
a. *Point Name*: **CURB_BACK_TOP** (Select from the list)

b. *Precision Input*: **hs= 5.5”,0** *[hs=14cm,0]*. Places point 5.5” [14cm] horizontally and at a 0 slope from the previous point.

c. Press **Enter** key.

The curb top is placed.

This component will be a closed shape. If your view shows only a line between the two points, right-click in the view and click **Closed Shape**.



4. Set the *Dynamic Settings* dialog as follows:

a. *Point Name*: **CURB_BACK_BOT** (Select from the list)

b. *Precision Input*: **dl= 0,-1** *[dl=0,-0.3]*. Places point 0 offset horizontally and -1’ [-0.3 m] vertically from the previous point.

c. Press **Enter** key.

5. Set the *Dynamic Settings* dialog as follows:

    a. *Point Name*: **CURB_FACE_BOT** (Select from the list)

    b. *Precision Input*: **dl= -1,0** *[dl=-0.3, 0]*. Places -1' [-0.3 m] horizontally and 0 offset vertically from the previous point.

    c. Press **Enter**.

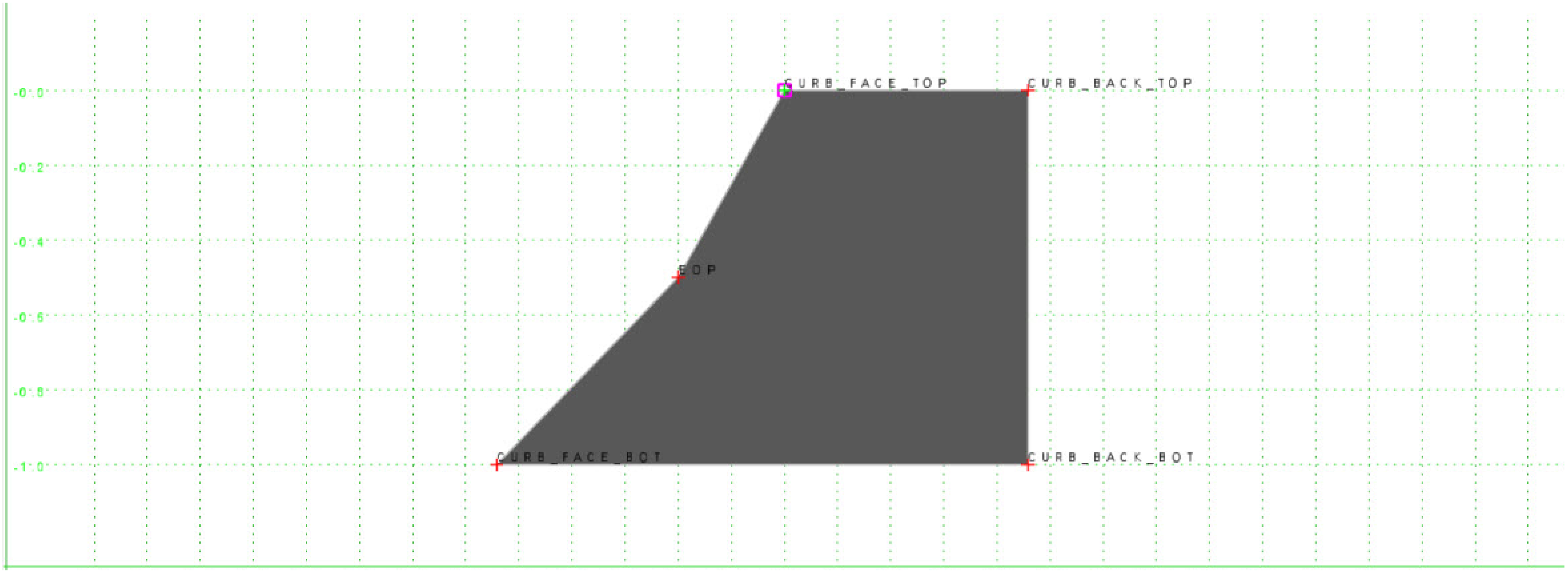
6. Set the *Dynamic Settings* dialog as follows:

    a. *Point Name*: **EOP** (Select from the list)

    b. Place the point near the midpoint of the curb face. The exact location is not important. We will correct that next.

    c. *Right-click* and select **Finish**.

    d. **Save** the *Template Library*



Define Where Pavement Connects to Curb

The EOP point represents the connection point between the pavement and the curb. Its Constraints need to be some distance below Curb Top and along the face of the curb. We will change the Constraints within the Point Properties dialog to precisely locate this point.

1. *Double-click* the **EOP** point.

   Correct the vertical position of the EOP point.

   a. Change the *Vertical Constraint* to **CURB_FACE_TOP**, either by selecting the constraint from the list or by clicking the **Target / Graphic Selector** button to the right of the of the constraint and then selecting the point.

   b. Set the *Value* to **-6”** *[-0.15]*.

   c. Click **Apply** to process the geometry change.

2. Correct the horizontal position of the EOP point.

   a. Change the *Horizontal Constraint* to a **Vector-Offset** constraint.

   b. Verify the *Parent 1* is set to **CURB_FACE_BOT**.

   c. Change the *Parent 2* point to **CURB_FACE_TOP**.

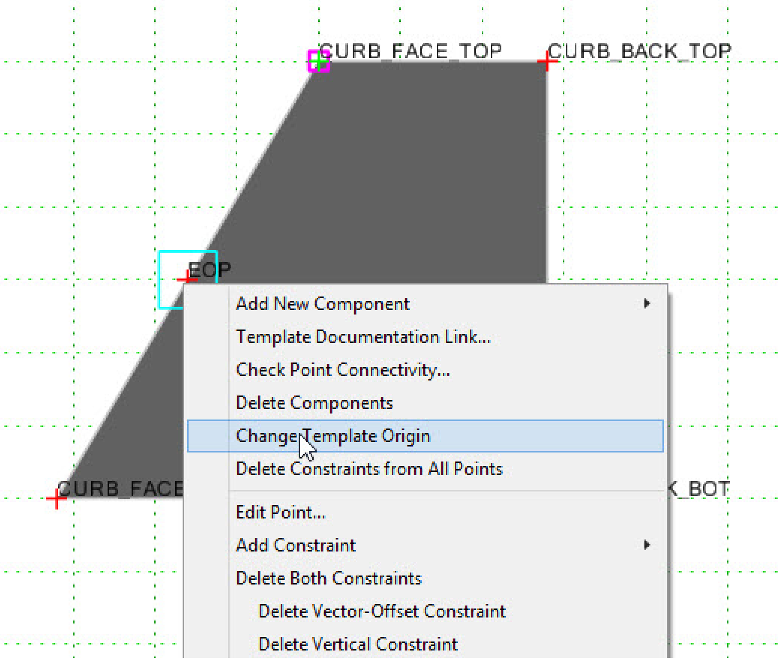
   d. Set the *Value* to **0**.

   e. Click **Apply** to process the geometry change.

   f. Close the *Point Properties* dialog.

   g. **Save** the *Template Library*

3. Finally, set the Template Origin to the **EOP** point by right-clicking on it and selecting **Change Template Origin**.



4. **Save** the *template library*.

Exercise 4 - Define Shoulder Rollover Locks

Description

This exercise teaches how to add to a Rollover Lock to a shoulder to set a maximum grade break between the shoulder and the pavement. This is a requirement for superelevation in some states.

Skills Taught

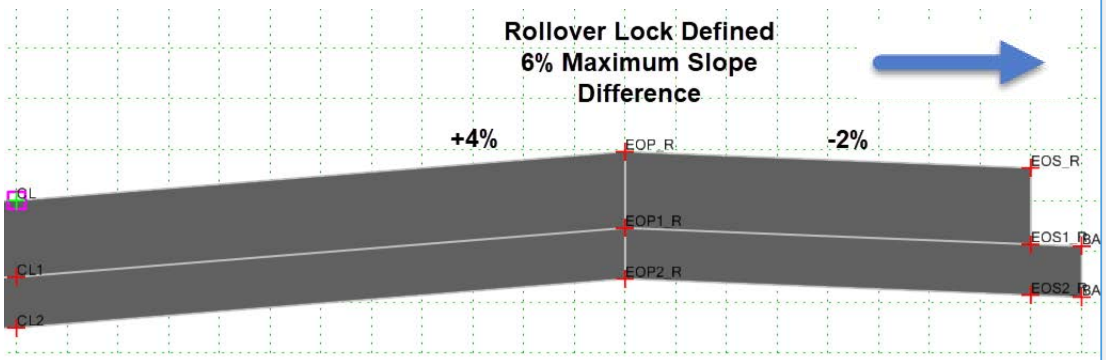
- Test Point Control
- Define Rollover Locks

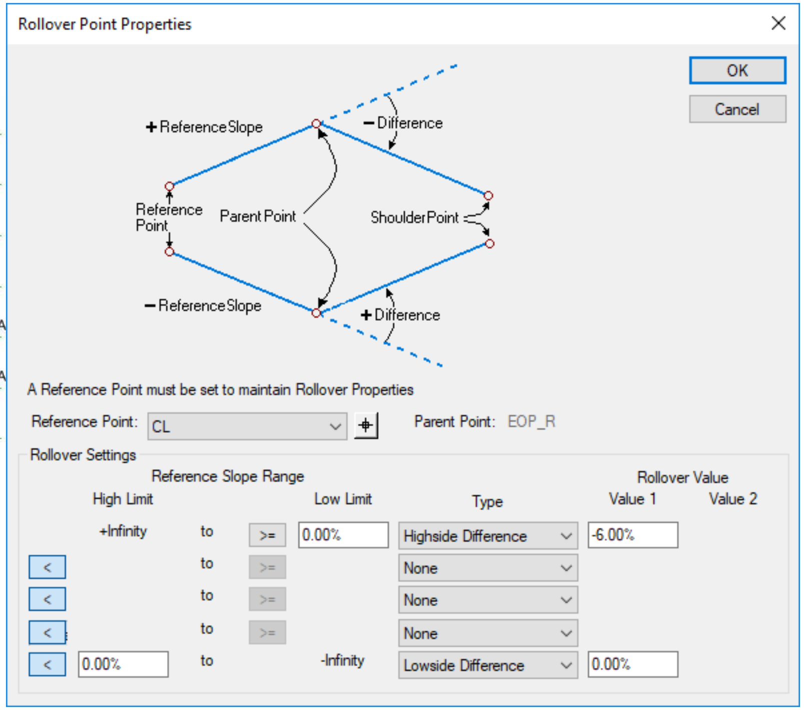
Rollover Locks

A common design requirement is to set a maximum slope difference between adjacent pavement and shoulder components. This is known as a Rollover Lock. Superelevation rotates pavement points and components when curves are encountered. Template Points within superelevated ranges are controlled by superelevation point controls that transition the pavement cross slopes as needed. Components adjacent to the superelevated pavement components can be controlled by Rollover Locks so that extreme grade breaks do not occur.

For example, consider a superelevated pavement surface that has a +4% cross slope and the adjacent shoulder cross slope is -4%. The slope difference would be 8% in this case. Without a rollover lock defined, the shoulder slope will always be defined at -4% even as the pavement cross slope changes due to superelevation (this is not desirable). Since a rollover lock has not been defined there is no way to lock the slope difference between the pavement and shoulder.

Suppose your design standard calls for a 6% maximum slope difference between the pavement and shoulder. A rollover lock could be defined with a highside difference set at -6%. This will ensure that the shoulder slope will never exceed the maximum slope difference.

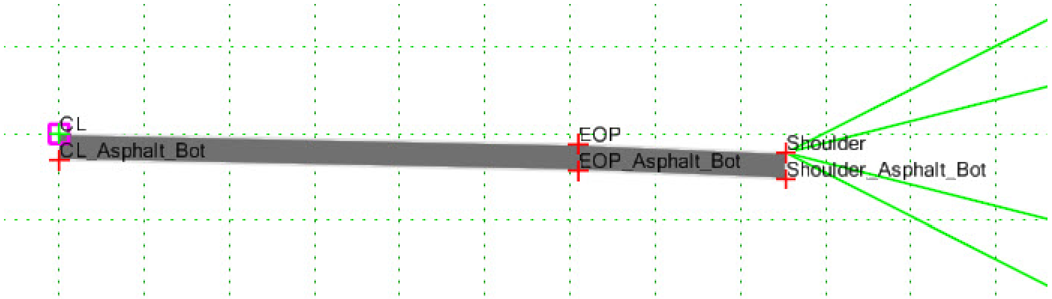




Create a Shoulder with a Rollover Lock

In this section you are going to learn how to define a shoulder rollover lock that will maintain a maximum slope difference of 7% between the pavement and the shoulder.

1. *Double-click* on the **NoHighGutter-modified** template to make it active.

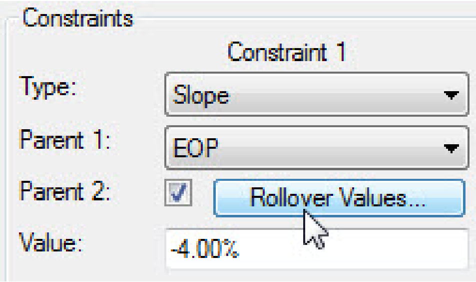


   a. Review the behavior of the template by moving the **EOP** vertically (*right-click* and select *Test Point Controls >* **Test Vertical Point Control** and *left-click* to end).

   Notice that the shoulder remains at a fixed slope no matter the slope of the main pavement section.

   b. *Double-click* on the **Shoulder** point to edit its properties.

   c. Set the check box between the *Parent 2* label and the *Rollover Values* button.



   d. Click on the **Rollover Values** button.

2. In the *Rollover Point Properties* dialog,

   a. Select **CL** for the *Reference Point*.

   b. Set the *Type* to **Highside Difference**

   c. Type **-7%** for *Value 1*.

   d. Type **0%** for the *Low Limit*

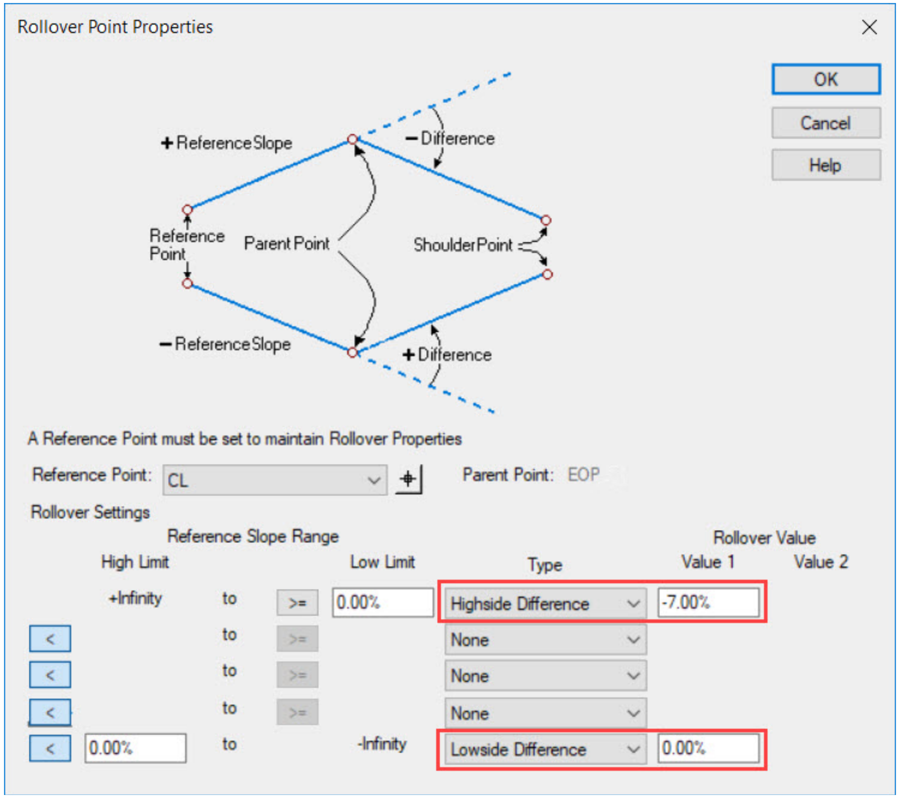
   e. At the bottom of the *Type* column, select **Lowside Difference**

   f. Set *Value 1* to **0%**.

   The reference slope is calculated between the Reference Point and the Parent Point. In this example, between the CL and the EOP.

   g. Click **OK** to accept the values and close the dialog.

   h. In the *Point Properties* dialog, click **Apply** followed by **Close**.



3. Test the template behavior by right-clicking on the **EOP** point and selecting *Test Point Controls >* **Test Vertical Point Control**.

   Notice that the shoulder slope now changes as the pavement section cross slope changes.

   - The shoulder will be at its defined slope (-4% in this example) anytime the pavement slope is between -4% and +3%.
   - When the pavement slope is greater than -4% (low side difference) the shoulder slope is locked at the same slope as the pavement.
   - When the pavement slope is greater than +3% the shoulder slope will be set based on the -7% highside difference. For example, if the pavement slope equals +4% then the shoulder slope will be -3%.

4. **Save** the *template library.*

Exercise 5 - Create a Widening Section that Matches Slope of the Existing Pavement

Description

In this exercise you will create a widening template that matches the existing pavement slope.

Skills Taught

- Use a Null point to constrain other points in the template
- Add Vector-Offset Constraint

Null Points

Null Components are single-point components. They do still create a 3D line string in the model but they have no Surface “Connection” and no component “segment connection”.

They have many uses:

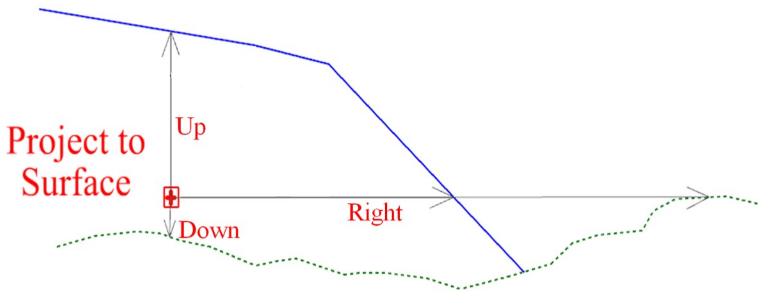
- A Temporary/Comparison point, such as a median or ROW limit
- “Trigger” Point (External Controls can move it from its “default” location)
- Surface Sample Point

Project To Surface Constraint

Tying into a surface is a common engineering need. All template points can be constrained to a surface using a Project to Surface constraint.

Engineering applications for Project to Surface include:

- “sampling” an existing surface or pavement elevation
- “sampling” two existing pavement points to compute an existing slope
- Transversely matching a surface or sub grade



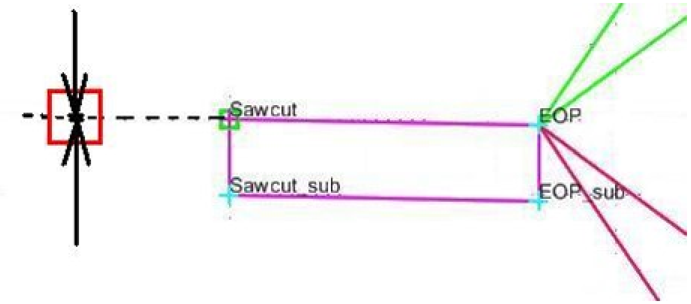
Project To Surface targets a surface in one of five directions: Up, Down, Left, Right and Any.

It is normally used in conjunction with another constraint. If the direction is Any, another constraint must be specified. A more precise name for Any would be Both. For example, if Any is constrained by a Horizontal it will search Both directions along the vertical line defined by the Horizontal constraint. Similarly, coupled with a Slope constraint, Any will search in both directions along that slope line.

Define Point to Sample Existing Pavement Slope

The location where the existing roadway will be cut and widened is defined by a geometric element, we will call the saw cut. The template you are creating will be pushed along that element to model the widening. The requirement is for this widened section to match the slope of the existing pavement.

To do this we need to sample the existing pavement to determine its slope. This will be accomplished by creating a null point 5ft [1.5m] to the left of the *Sawcut* point. This is a location near the center of the adjacent travel lane.The slope of the existing pavement will be projected from the null/sample point through the saw cut (existing edge of pavement) to the new edge of pavement.



1. *Double-click* on the **Widen** template to make it active.

   a. Set the *Dynamic Settings* dialog as follows:

   - *Point Name*: **XSample** (Type it, this point name is not in our point name list)
   - *Feature Definition*: **Linear\Template Points\DNC\TL_Draft-DNC**
   - *Apply Affixes*: Cleared

   b. *Right-click* on the Current Template window, then select *Add New Component >* **Null Point**.

   c. Place the point approximately **5** feet *[1.5 m]* to the left of the *Sawcut* point. The exact position is not important at this time.

   d. *Double-click* on the **xSample** point to open the *Point Properties* dialog.

2. Define the *Constraint 1* to establish its horizontal position.

    a. Set the *Type* to **Horizontal**.

    b. Set the *Parent 1* to the **Sawcut** point.

    c. Set the *Value* to **-5** *[-1.5]*.

    d. *Click* **Apply**.

3. Define the *Constraint 2* to establish its vertical position by projecting onto the pavement surface.

    a. Set the *Type* to **Project to Surface**.

    b. Set the *Parent 1* to **Any Direction**.

    Remember that we have already constrained the point horizontally so effectively this limits the project to surface to search vertically either above or below to find the surface.

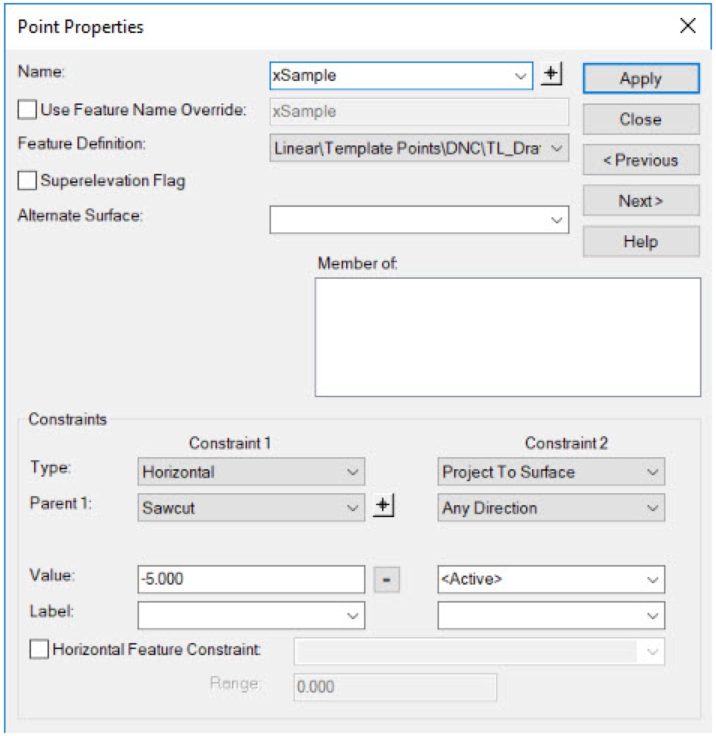
    c. Set the *Value* to **<Active>**.

    This constraint will seek the active surface, not a specifically named surface. Using the active surface makes this template flexible to be used on different projects and situations without the need to modify the template.

    d. *Click* **Apply**.

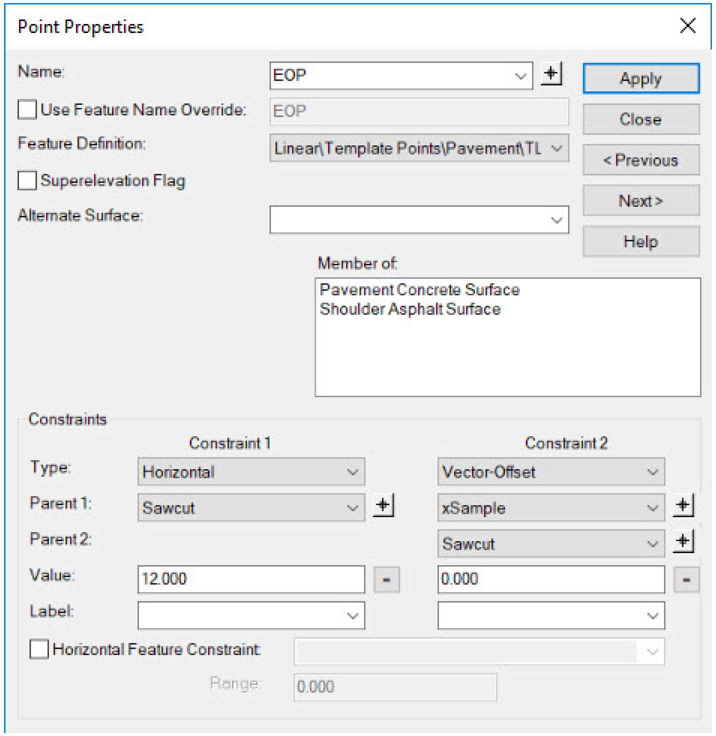
    e. **Close** the *Point Properties* dialog.

    f. **Save** the *Template Library*



Define the New Pavement Slope to Match the Existing Pavement

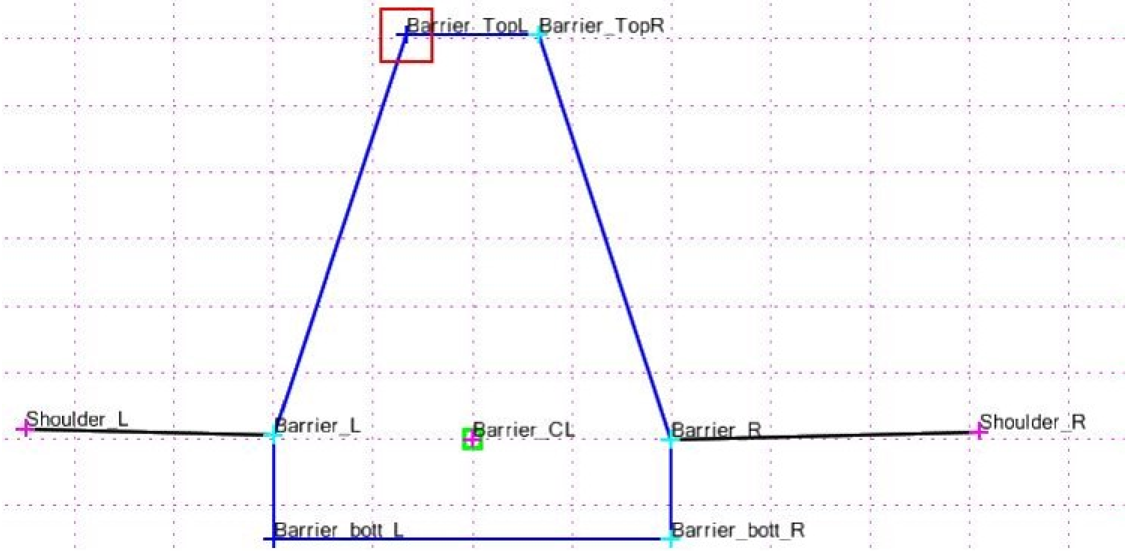
1. *Double-click* on the **EOP** point.
   a. Change the **Vertical** *Constraint* to a **Vector-Offset** constraint.
   b. Set *Parent 1* to **XSample**.
   c. Set *Parent 2* to **Sawcut**.
   d. Set the *Vector-Offset Constraint Value* to **0**.
   e. This is the offset from the vector - the line from the first parent to the second. A negative value is to the left of the line.
   f. *Click* **Apply**.
   g. **Close** the *Point Properties* dialog.
   h. Test the template behavior by right-clicking on the **XSample** point and selecting *Test Point Controls* **> Test Vertical Point Control**.
   i. **Save** the template library.



Exercise 6 - Create a Barrier that Maintains a Minimum Height Above Pavement

Description

In this exercise, you will create a barrier and ensure that it remains a minimum height above both the left and right pavements.



Skills Taught

- Add Vertical Maximum constraint
- Add Vertical Minimum constraint

Horizontal and Vertical Minimum and Maximum Constraints

The Horizontal and Vertical Minimum and Maximum constraints are similar to the Horizontal and Vertical constraints in that a point (the child) is constrained at a vertical or horizontal distance from its parent. The difference is that the Minimum and Maximum constrains have two parents, not one. When using a Minimum constraint the parent with the lesser value (lowest for vertical constraints or furthest left for horizontal constraints) is used. When using the Maximum constraint the parent point with the greater value (highest or furthest right) is used. The two parent points are evaluated at each template drop so the controlling parent point can change along the corridor or linear template.

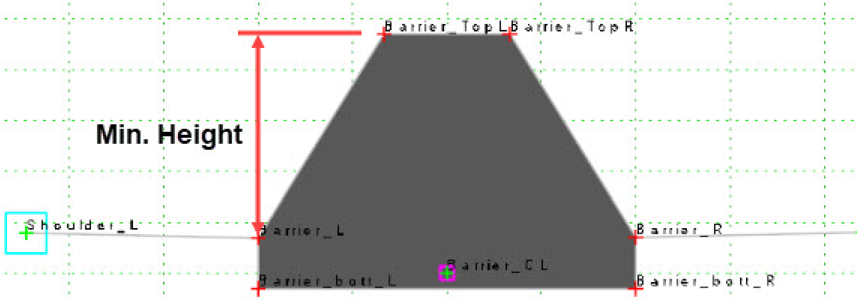
Horizontal Maximum = the Right (of two points)

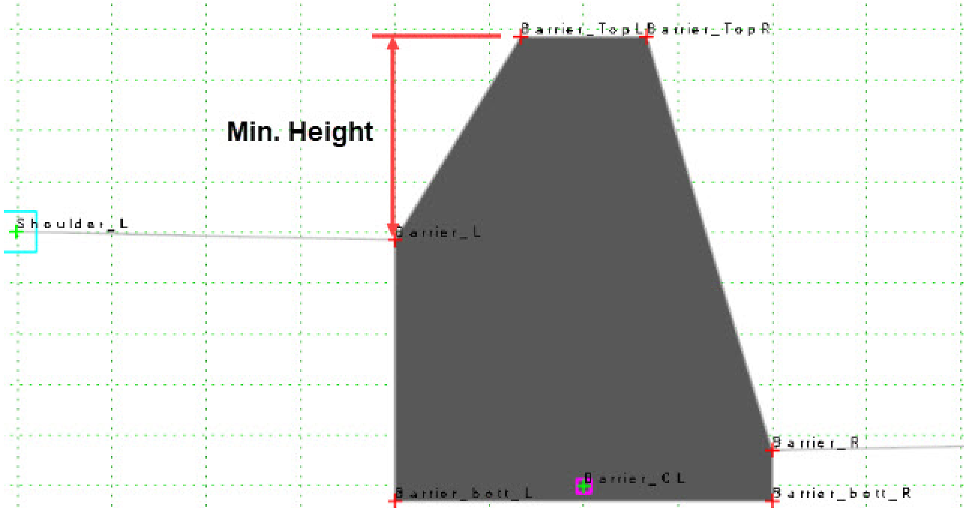
Horizontal Minimum = the Left (of two points)

Vertical Maximum = the Higher (of two points)

Vertical Minimum = the Lower (of two points)

One example of evaluating two points is a median barrier that must maintain a minimum height above both the left and right pavements. The pavements have differing grade lines and thus will be at different elevations as they transition along the corridor and through superelevation. The top of the barrier is constrained using a Vertical Maximum constraint with the Left Pavement Edge and the Right Pavement Edge as the two parent points. Using the Vertical Maximum constraint causes the top of the barrier to be located at a defined distance above the higher (maximum) elevation of the two parent points.





Constrain Barrier to a Minimum Height

1. *Double-click* on the **Barrier_Minimum** template to make it Active.

   The Barrier needs to be at least 4 ft [1.3 m] higher than the highest of the left or right shoulder points.

   a. **Move** one of the *Shoulder* points higher and lower.

   Notice that the barrier does not adjust and the elevation difference between the shoulder and top of the barrier is not restricted to a minimum of 4 ft [1.3 m].

2. In previous exercises you learned to define constraints through the Point Properties dialog. Constraints can also be defined directly from the right click menu. This exercises uses this shortcut technique.

   a. *Right-click* on the **Barrier_TopL** point, click *Add Constraint >* **Vertical Maximum**.

      The lower left corner of the Create Template dialog prompts to "*DB: Select Parent Point, RST: Exit*".

   b. *Click* on the **Barrier_L** point to define *Parent 1*.

      The lower left corner of the Create Template dialog prompts to "*DB: Select Second Parent Point, RST: Exit*".

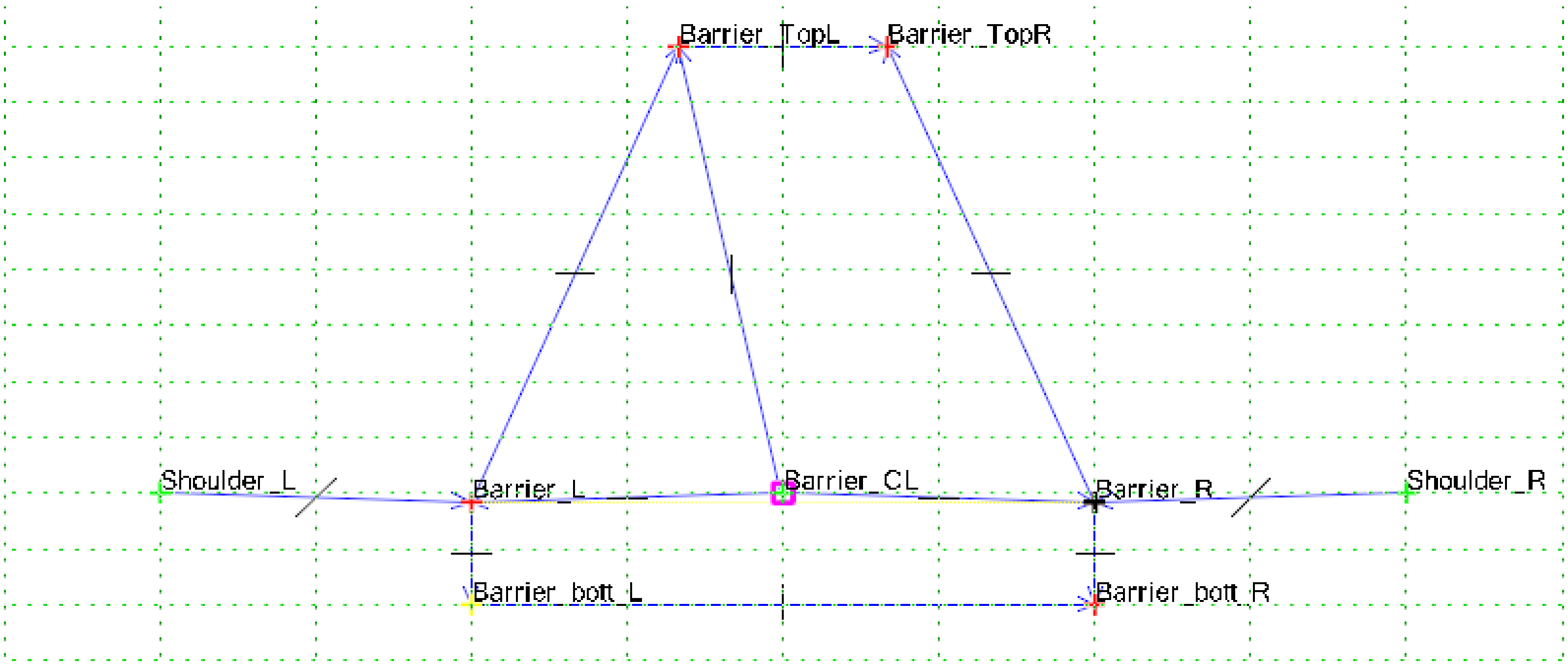
   c. *Click* on the **Barrier_R** point to define *Parent 2*.

   d. In the *Vertical Maximum* dialog, key in **4** *[1.2]* for *Offset*.

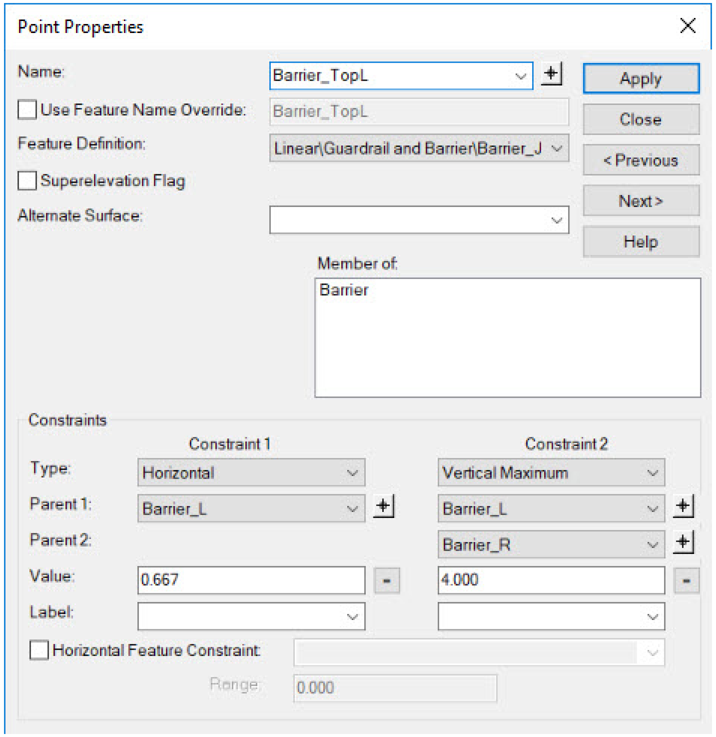
      This defines that the *Barrier_TopL* point remains 4 ft [1.2m] above the **maximum** elevation of either the *Barrier_L* point or the *Barrier_R* point.

   e. *Click* **OK** to accept the offset and close the dialog.

The Constraints of the template are shown below, with the Vertical Maximum highlighted. Note the Vertical Constraint “bar” on the line from the point to the midpoint of the line between the two Parent Points.



3. Open the *Point Properties* for the *Barrier_TopL* point and look at how *Constraint 2* was defined. This exercise uses the *Add Constraint* **> Vertical Maximum** interactive definition but the constraint could also have been defined in the dialog.



4. **Close** the *Point Properties*.
5. **Move** one of the *Shoulder* points higher to verify the behavior of the barrier. The top of the barrier should move vertically to maintain a minimum offset from the highest pavement.
6. **Save** the template library.

Constrain Barrier Footing to a Minimum Depth

1. **Move** one of the Shoulder points below the barrier bottom.

   Notice that the bottom of the barrier is not constructed properly. Currently the *Barrier_bott_L* point is not constrained vertically so it does not adjust when the barrier adjusts. The *Barrier_bott_R* point is constrained to be at the same elevation as the left bottom barrier point.

   a. *Right-click* on the **Barrier_bott_L** point, click *Add Constraint >* **Vertical Minimum**.

   The lower left corner of the Create Template dialog prompts to "*DB: Select Parent Point, RST: Exit*".

   b. *Click* on the **Barrier_L** point to define *Parent 1*.

   The lower left corner of the Create Template dialog prompts to "*DB: Select Second Parent Point, RST: Exit*".

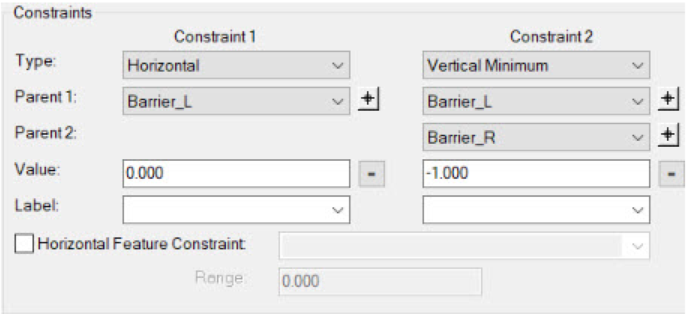
   c. *Click* on the **Barrier_R** point to define *Parent 2*.

   d. In the *Vertical Minimum* dialog, key in **-1** *[-0.3]* for *Offset*.

   e. *Click* **OK** to accept the offset and close the dialog.

   The *Barrier_bott_L* point is now constrained to be **1 ft** *[0.3 m]* below the **minimum** elevation of either the *Barrier_L point* or the *Barrier_R* points.

   f. Review the newly created constraints for the **Barrier_bott_L** point in the *Point Properties* dialog.



   g. Verify the behavior of the barrier now when one of the **Shoulder** points is moved.

   h. **Save** the template library.

Exercise 7 - Create Display Rules to Turn Off Median Barrier and Turn On Median Ditch

Description

In this exercise, you will create a template that will display a median ditch when the median width is 20’ *[6 m]* or greater and display a median barrier when the median width is less than or equal to 20’ *[6 m]*. You will accomplish this by using Component Display Rules.

Templates represent “typical sections”, and while “typical”, they do vary from station to station. Consider a sideslope treatment: it typically targets the existing ground. The daylight line may be at the outside of a cut slope or a fill slope. The new road may be above or below the existing ground and the height or depth typically varies from station to station along the corridor. There are many reasons where it would be important to be able to evaluate template geometry automatically at every station a template is dropped along a corridor.

Display Rules evaluate to a condition of True or False based on some condition. Assigned to a template it will evaluate the geometry of two template points - at *every* station. Is the daylight line above or below the pavement edge? Is the distance between divided inside pavement edges less than or greater than some distance. Is the slope of the pavement positive or negative?

A Conditional Expression can be assigned to a Template Component. A Conditional Expression is made up of one or more Display Rules and evaluates as either True or False. If the Conditional Expression evaluates as False, the Template Component will not display. This simple logic can provide very powerful capabilities on every template dropped along a corridor - automatically.

Skills Taught

- Create Display Rules
- Create Conditional Expression

Display Rules

Component Display Rules provide automated instructions detailing when a component should be displayed (turned on) or undisplayed (turned off) until needed. Component Display Rules can evaluate distances and slopes between two template points to determine if a component should be displayed or undisplayed.

The types of Component Display Rules include:

- *Horizontal:* Horizontal distance between two points
- *Absolute Horizontal:* Absolute horizontal distance between two points
- *Vertical:* Vertical distance between two points
- *Absolute Vertical:* Absolute vertical distance between two points
- *Slope:* Slope between two points
- *Absolute Slope:* Absolute slope between two points
- *Component is Displayed:* Checks if existing components are displayed

Conditional Expression

A Conditional Expression uses one or more Component Display Rules to determine if a component is allowed to display or not. A Component Display Rule evaluates to either TRUE or FALSE when evaluating between two points.

When the Conditional Expression evaluates to FALSE the component will not display. When it evaluates to TRUE the component is allowed to display, though it still may not display it. Template component may not display if disabled by other conditions such as being part of an End Condition that was not used or being a child of a component that is not displayed.

Conditional Expressions are logical statements combining one or more Display Rules. Compound expressions can use parentheses for grouping and logical AND, OR, and NOT operators. At each template drop, individual Display Rules are evaluated as either TRUE or FALSE.

Those results are then evaluated in a Conditional Expression resulting in the component being displayed (TRUE) or not displayed (FALSE). Compound expressions are evaluated in this precedence:

1. Left to Right
2. Parenthesis
3. NOT
4. AND and OR

For example, consider a median between two lanes of a divided roadway where a barrier is required if the median width is less than 20' *[6 m]*. However to complicate this further, if the vertical elevation difference of the two roadways is greater than 1' *[0.3 m]* a different type of barrier is required.

Name	Expression	Evaluation
Display Rule 1	Horizontal Type Calculation: Left Roadway Inside EOP - Right Roadway Inside EOP < 20	■ TRUE if median width < 20 ■ FALSE if median width >= 20
Display Rule 2	Vertical Type Calculation: Left Roadway Inside EOP - Right Roadway Inside EOP < 1	■ TRUE if pavement elevation delta < 1 ■ FALSE if pavement elevation delta >= 1
Conditional Expression on Barrier Type 1 component (standard barrier)	Display Rule 1 AND Display Rule 2	■ TRUE if both Display Rules are TRUE ■ FALSE if either or both Display Rules are FALSE
Conditional Expression on Barrier Type 2 component (barrier for vertically separated roads)	Display Rule 1 AND NOT Display Rule 2	■ TRUE if Display Rule 1 is TRUE and Display Rule 2 is FALSE (NOT FALSE evaluates to TRUE) ■ FALSE if Display Rule 1 is FALSE or Display Rule 2 is TRUE (NOT TRUE evaluates to FALSE)
Conditional Expression on Median Ditch component	NOT Display Rule 1	■ TRUE if Display Rule 1 is FALSE (NOT FALSE evaluates to TRUE) ■ FALSE if Display Rule 1 is TRUE

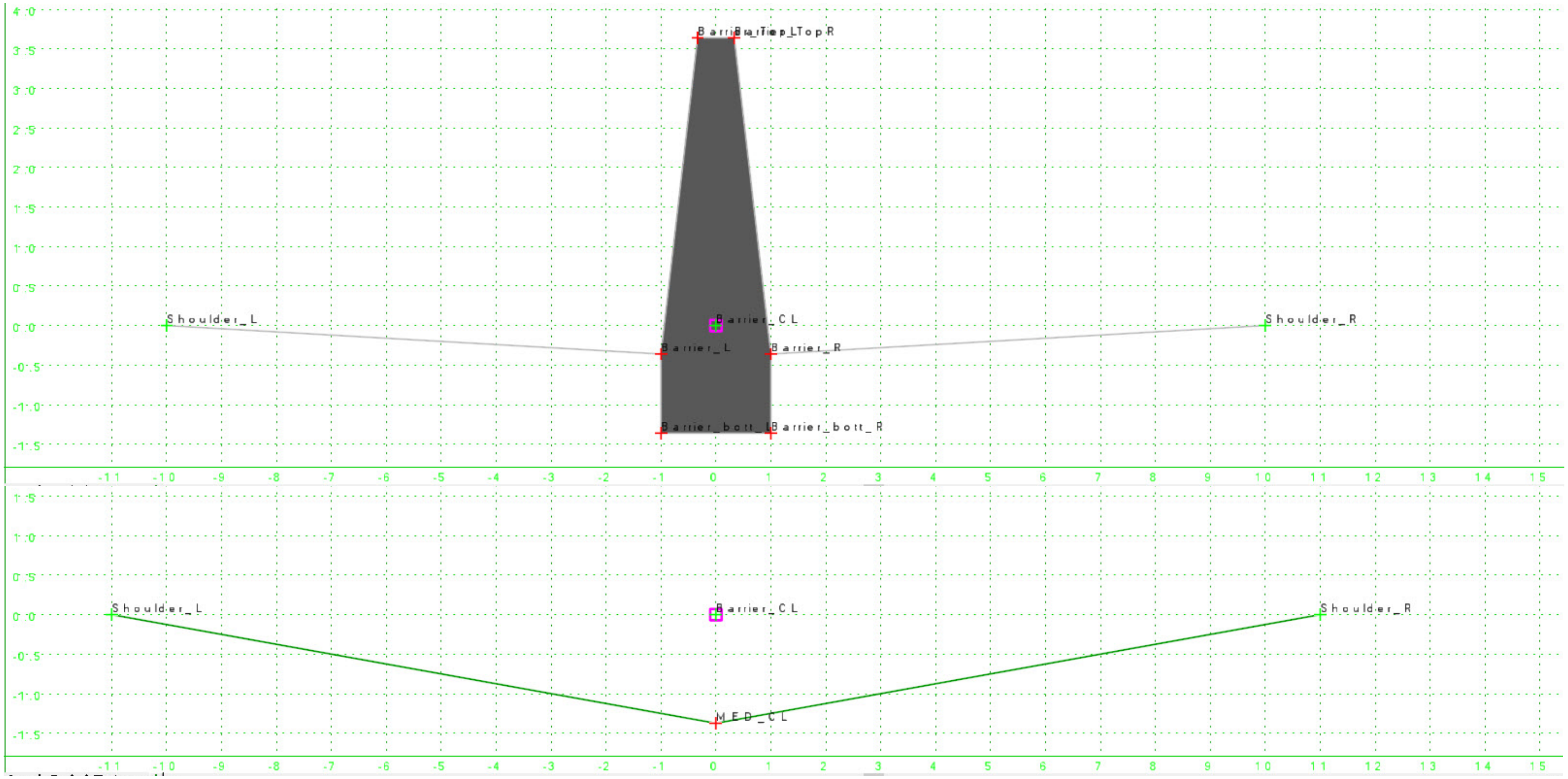
Test the Median Barrier and Ditch Template

1. *Double-click* on the **Median Barrier and Ditch** template to make it active.

   a. *Move* the **Shoulder_L** point horizontally to the left or right any distance. Observe that no matter how wide or narrow the median width (distance between *Shoulder_L* and *Shoulder_R* points) becomes the median barrier and median ditch are always displayed.

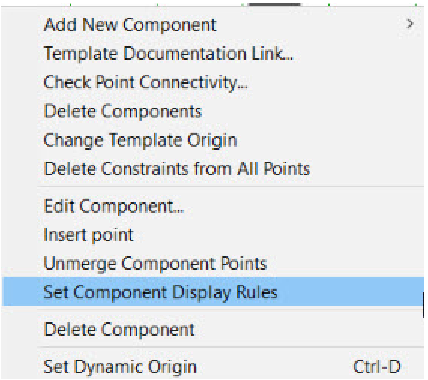
   b. *Click* Undo so the *Shoulder_L* point goes back to its original location.

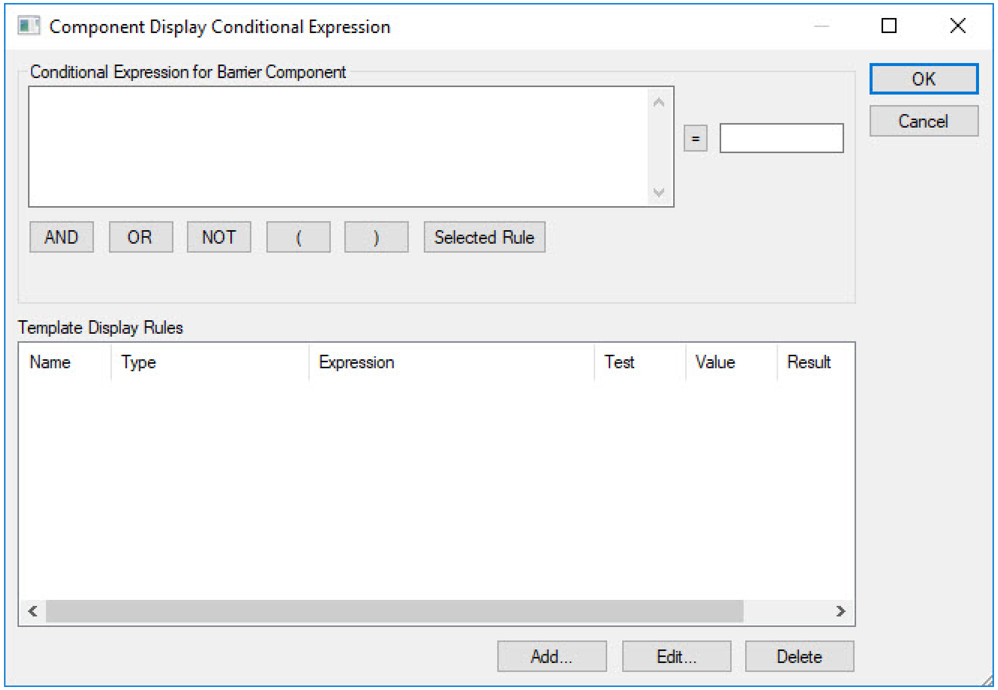
Our design criteria states that when the median width is equal to 20’ *[6 m]* or less then median barrier is required and when the median width is greater than 20’ then a depressed grass median will be used.



Create the Display Rule for the Median Barrier

1. *Right-click* on the **Barrier** component, select **Set Component Display Rules**.





The *Component Display Conditional Expression* dialog opens. The top frame lists the component's Conditional Expression and has controls to build them. The bottom of the dialog lists the Template's Display Rules and controls to Add, Edit, and Delete them.

a. *Click* the **Add** button to add a new Display Rule.

b. Create a *Display Rule* that is true when the median width is equal to or less than 20’ *[6 m]* by defining the settings as below.

Note: Spaces are not allowed in display rule names so using underscores _ to make them easier to read is a good practice.

Name: **Barrier_On**

Description: **Barrier displayed when median is equal to or less than 20’** *[6 m]*

Type: **Absolute Horizontal**

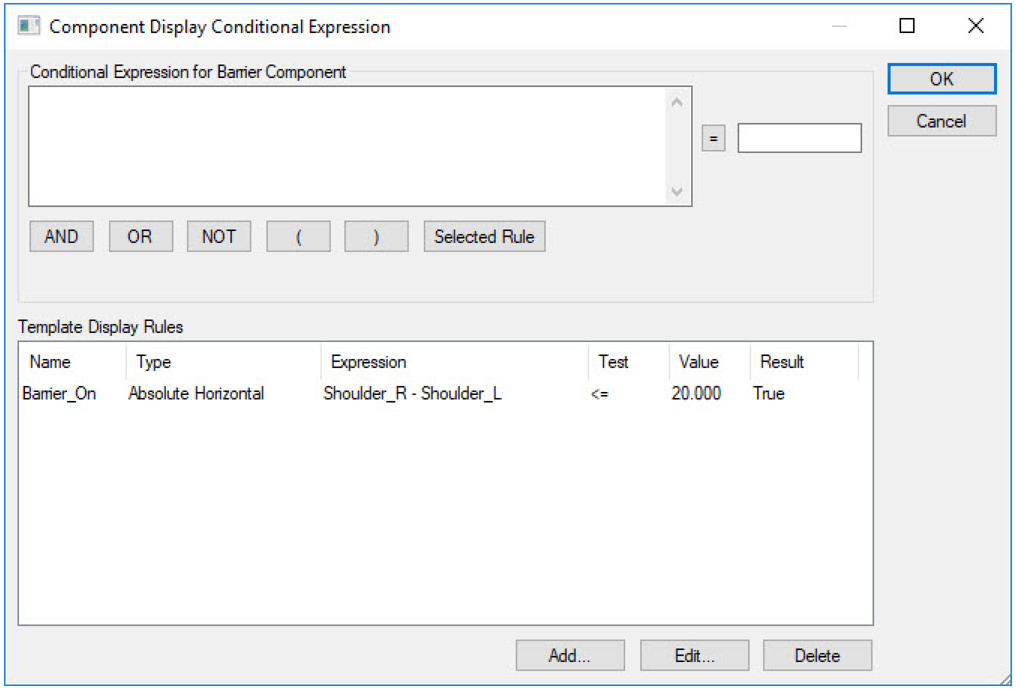
Between: **Shoulder_R**

And: **Shoulder_L**

<=20.00 *[6 m]*

c. *Click* **OK** to close the *Display Rule* dialog.

The Component Display Conditional Expression dialog now shows the **Barrier_On** display rule is listed in the *Template Display Rules* list.



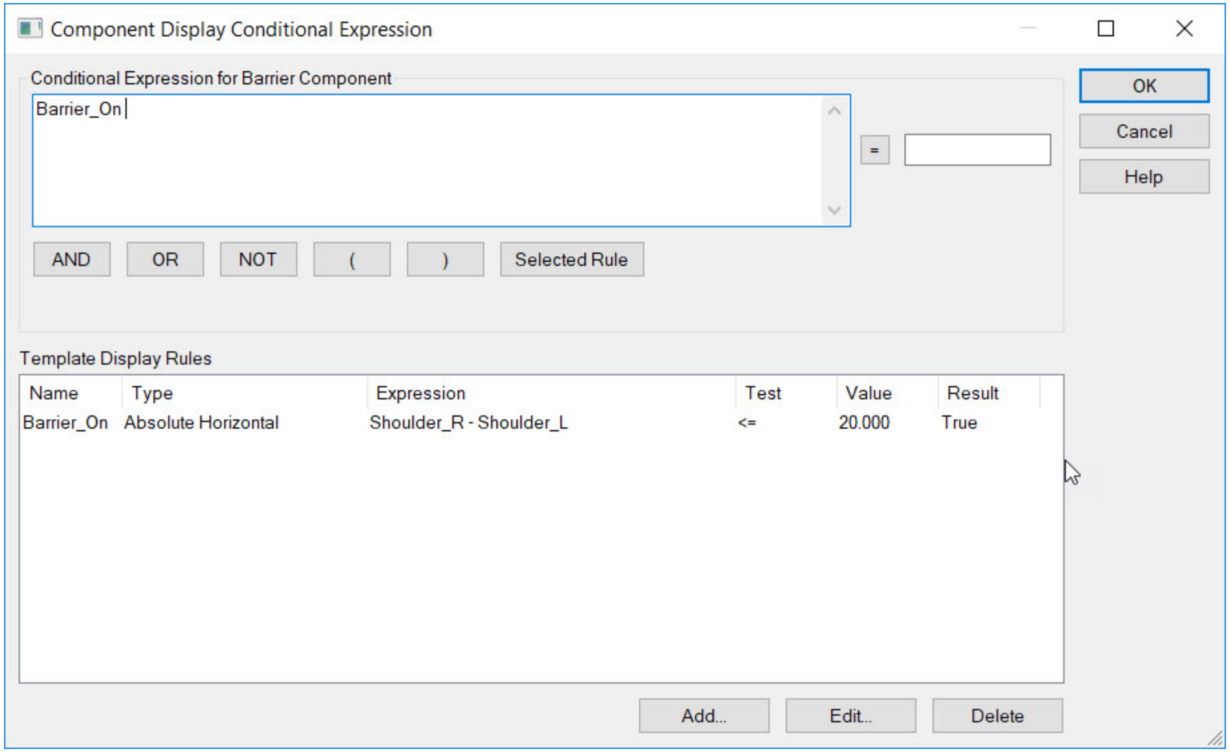
2. Create the *Conditional Expression* for the Barrier.

   a. Highlight the **Barrier_On** Display Rule from the *Template Display Rules* list by clicking on it.

   b. *Click* the **Selected Rule** button to add the rule to the conditional expression.

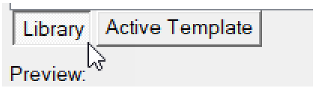
   c. *Click* **OK** to save the Conditional Expression. This conditional expression will evaluate to TRUE by default since the horizontal distance between **Shoulder_R** and **Shoulder_L** is currently 20' *[6 m]*.

   Note: While the conditional expression can be keyed in, using the buttons prevents typing errors and misspellings.



   d. Test the template by moving the **Shoulder_L** or **Shouler_R** points to the left or right horizontally away from the origin. The barrier component should un-display when the median width is greater than 20' *[6 m]* and only display when the median width is equal to or less than 20' *[6 m]*.

   Note that if Active Template is selected, all Components are shown, none disappear in testing. Use Library.

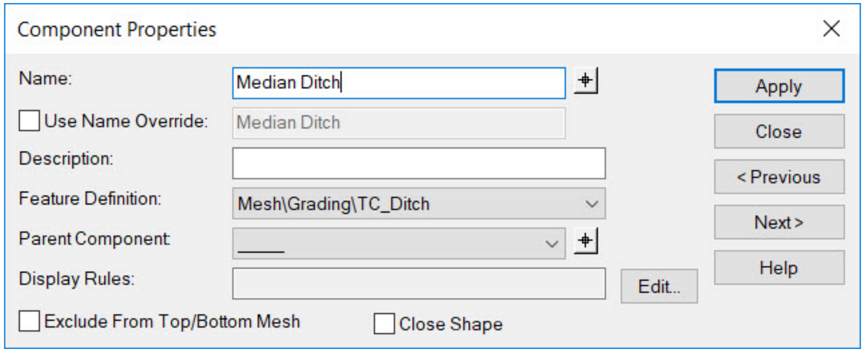


Create Display Rule for the Median Ditch

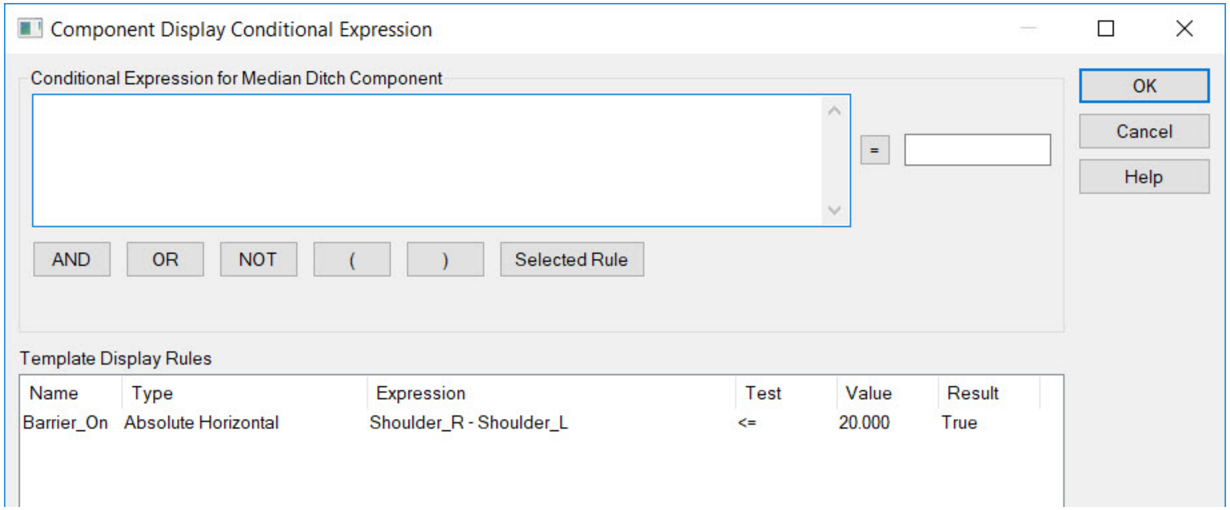
In this section, we will create a display rule that will display the median ditch only when the median width is greater than 20’ *[6 m]*.

1. Create Median Ditch Display Rule.

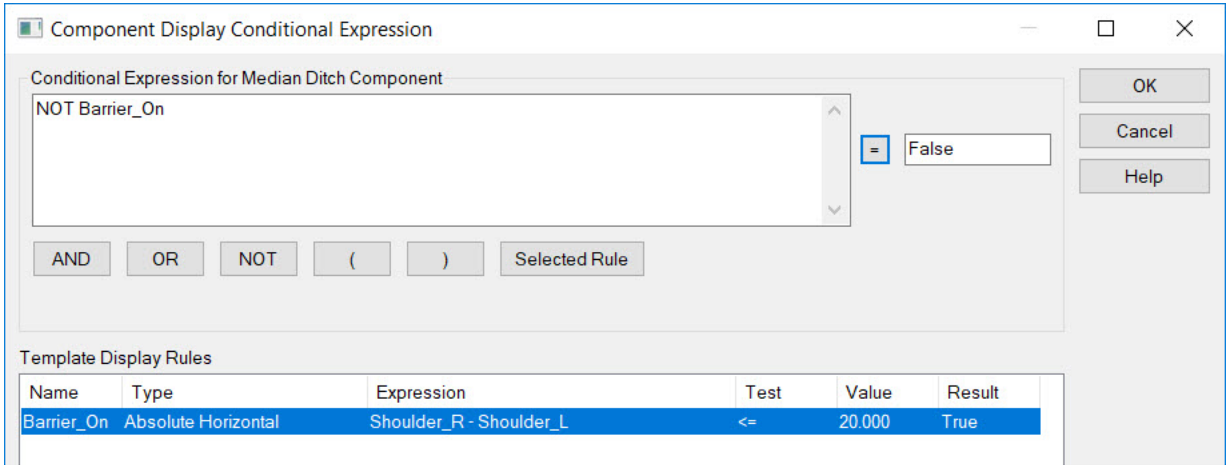
   a. *Double-click* the **Median Ditch** component to view the *Component Properties*.



   b. In the *Component Properties* dialog, select **Edit**. This will open the *Component Display Conditional Expression* window.



c. Select the **Barrier_On** display rule.



d. *Click* **NOT**

e. *Click* **Selected Rule**

f. *Click* **=** to see the results of the conditional expression.

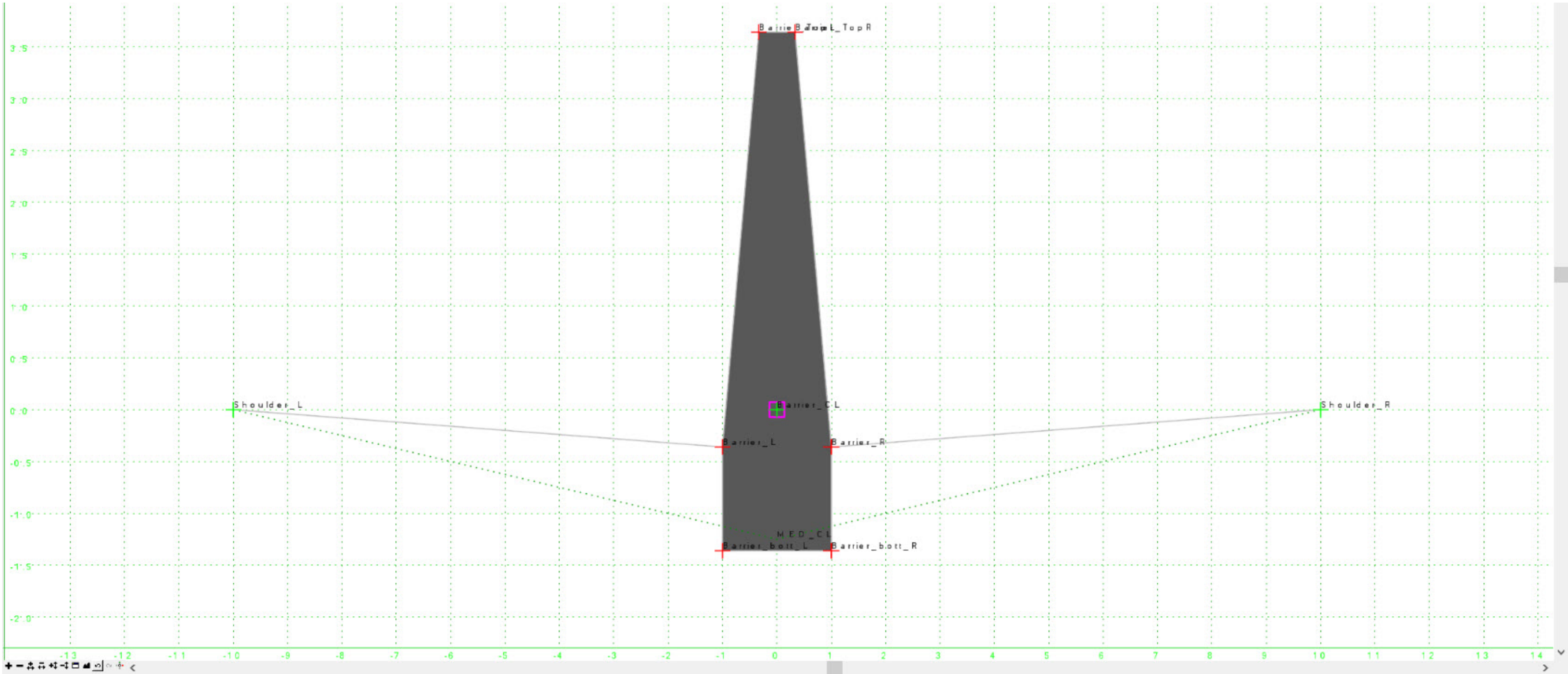
Note: The conditional expression results as **False.** Since the **Barrier_On** display rule is currently true, by adding the NOT command will un-display the median ditch. If the median width becomes greater than 20' *[6 m]* then the barrier will turn off and the median ditch will then display.

g. *Click* **OK** to close the *Component Conditional Expression* window.

h. From the *Component Properties* window, *Click* **Apply** to save all changes.

i. *Click* **Close** to complete.

The completed template should like the image below.



2. Test the template to verify it is working as designed.

   a. *Right-click* **Shoulder_R** point

   b. Select *Test Point Controls* > **Test Horizontal Point Control**

   c. Move the **Shoulder_R** point to the right and notice how the median ditch will display and barrier will un-display.

   d. Move the **Shoulder_R** point back towards the template origin and note how the median ditch will un-display and the barrier will display once the median width is less than or equal to 20’ *[6 m]*.

3. **Save** the template library.

Skills Assessment

The questions below will test your retention of the skills covered in this course.

---

1. Which of the following statements are true about a Simple Component?
    a. It's geometry is based on width, thickness and slope
    b. It's outside top point is constrained to the inside top point by Width and Slope Constraint.
    c. Parametric Labels of Width, Thickness and Slope are automatically assigned to corresponding point constraints.
2. Which of the following statements are true about a Affixes?
    a. Easier to add them later than to take them off en masse.
    b. It's better to leave the Apply Affixes checkbox cleared by default.
    c. There is an Apply Affixes to existing Template Points tool.
3. You have a traveled way that is being superelevated up to 6% and shoulder adjacent to the traveled way. Your design criteria states you not allowed to have a grade break greater than 4%. What do you do?.
    a. Superelevate the shoulder
    b. Set the Shoulder Rollover to a Highside Difference of -4%
4. The Vector Offset constraint can be used to make a point follow a slope between two points.
    a. TRUE
    b. FALSE
5. Component Display Rules can be used to display and undisplay template components
    a. TRUE
    b. FALSE

Skills Assessment - Answers

The answers to the skills assessment questions are highlighted below.

1. Which of the following statements are true about a Simple Component?
    a. **It's geometry is based on width, thickness and slope**
    b. **It's outside top point is constrained to the inside top point by Width and Slope Constraint.**
    c. Parametric Labels of Width, Thickness and Slope are automatically assigned to corresponding point constraints.
2. Which of the following statements are true about a Affixes?
    a. **Easier to add them later than to take them off en masse.**
    b. **It's better to leave the Apply Affixes checkbox cleared by default.**
    c. There is an Apply Affixes to existing Template Points tool.
3. You have a traveled way that is being superelevated up to 6% and shoulder adjacent to the traveled way. Your design criteria states you not allowed to have a grade break greater than 4%. What do you do?.
    a. Superelevate the shoulder
    b. **Set the Shoulder Rollover to a Highside Difference of -4%**
4. The Vector Offset constraint can be used to make a point follow a slope between two points.
    a. **TRUE**
    b. FALSE
5. Component Display Rules can be used to display and undisplay template components
    a. **TRUE**
    b. FALSE

Summary

In this course you learned many tools and techniques for creating templates.

You learned how to:

- Create a Pavement Slab Component
- Place Template Points Using Dynamic Settings
- Insert Template Points
- Review and Edit Template Point Properties
- Create Parametric Constraint Labels
- Create a Curb Component
- Define Shoulder Rollover Locks
- Create Vector-Offset Constraints
- Create Pavement Widening Components That Match Existing Pavement Slopes
- Create a Median Barrier
- Create Vertical Minimum and Vertical Maximum Constraints
- Define Component Display Rules

.