OpenRoads Designer User Manual

U.S. Department of Transportation
Federal Highway Administration

Chapter 8

TEMPLATE LIBRARY

2021 R2 Update 10 – Version 10.10.21.04 – WorkSpace 10.10V
January 2023

Chapter 8 Template Library

This chapter covers the creation and modifications of Templates. Templates are cross-sectional elements used in Corridor and Site Modeling.

TABLE OF CONTENTS

8D – Template Components 8-55

8E – Creating and Manipulating Points and Components 8-79

8F – Template Creation Workflows 8-92

8G – Surface Templates 8-118

8A – INTRODUCTION TO TEMPLATES AND THE CORRIDOR

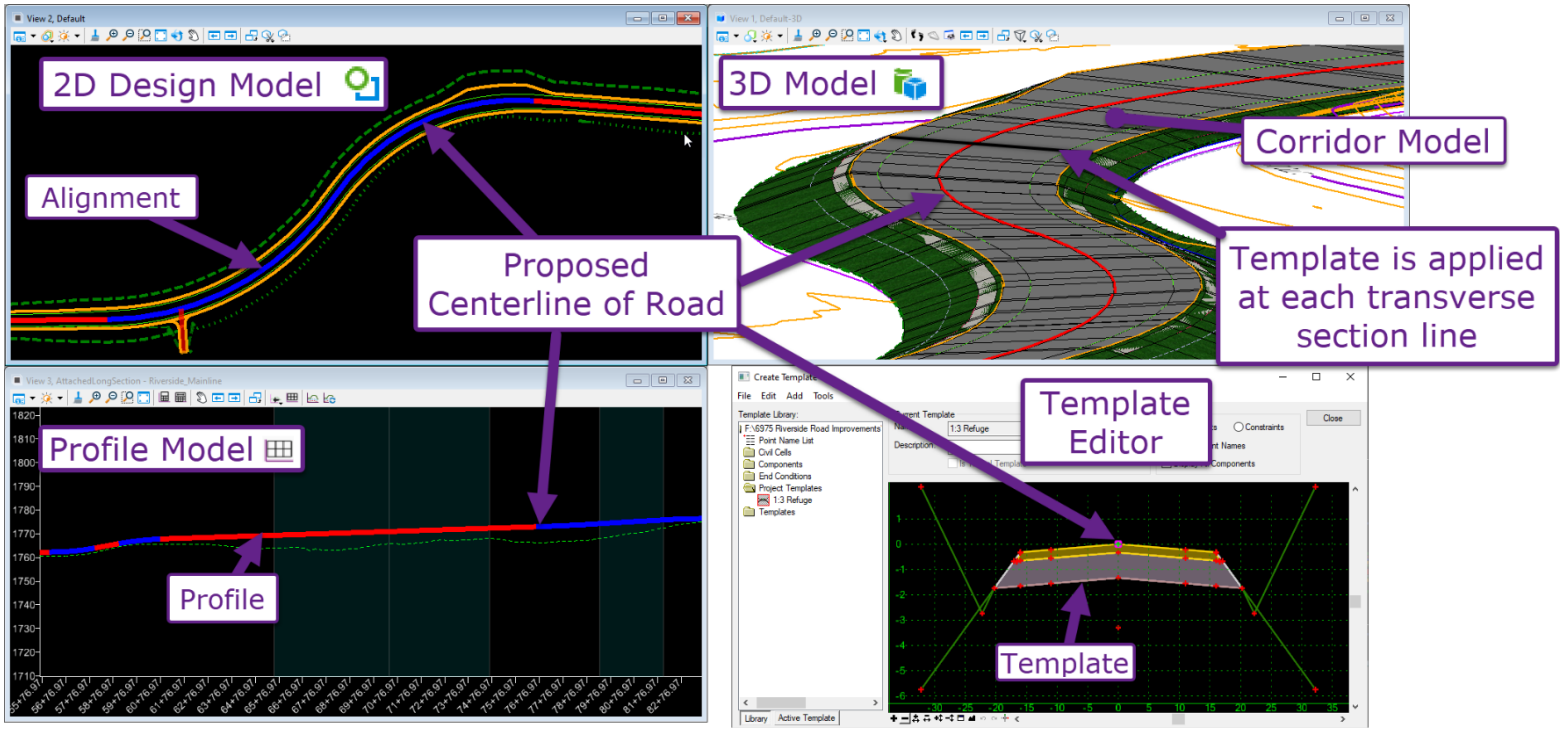
Corridor modeling is the foundation of modern roadway design. In OpenRoads, 3-dimensional model of a roadway can be created and related to an existing ground terrain model. Corridor models allow the User to visualize how a design interacts with the existing ground surface and perform useful quantity calculations, such as earthwork cut/fill volumes and material quantities. There are three distinct parts that makeup a Corridor: the Alignment, Profile, and Template.

Alignment: Alignments are also referred to as Horizontal ORD Elements in this Manual. Alignments are created and edited in the *2D Design Model*. Typically, the proposed centerline of road is drawn out as an Alignment and used as the baseline for a road Corridor model. For Alignment creation procedures, see *7D – Create Horizontal ORD Elements*.

Profile: Profiles are also referred to as Vertical ORD Elements. The Profile is associated with the Alignment and represents the vertical baseline for the road Corridor model. Profiles are created and edited in the *Profile Model* of the Alignment. When a Profile is *activated*, the profile elevation information is projected on the Horizontal Alignment to create a 3D Linear Element.
See *7F – Create Vertical ORD Elements*.

Template: The Template can be thought of as the "Cross Section" or typical section for a Corridor Model. Templates are stored in separate *Template Library files* outside of the OpenRoads software. Template Library files have an .itl file extension. To create or edit a Template, a Template Library file (.itl) must be loaded into the Template Editor. The FLH WorkSpace has an FLH-specific Template Library, which should be used for Corridor modeling. See *8A.1 Accessing the Template Editor and Template Libraries*.

Corridor: The Corridor is created by combining the Alignment, Profile, and Template. To create a Corridor, the Template is extruded along the path of the Alignment/Profile. The Template is applied or *dropped* along the Alignment/Profile at a set station interval. The space between adjacent transverse *Template Drops* is filled by interpolation.



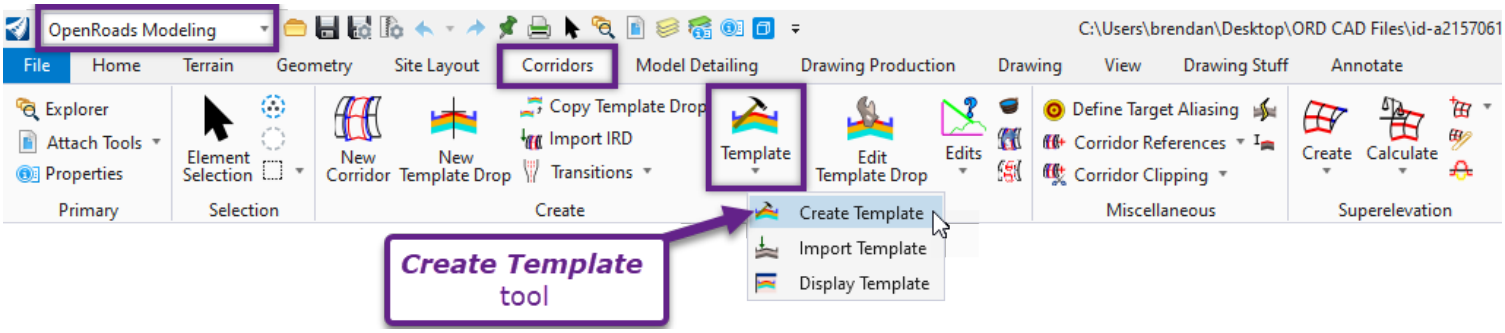
8A.1 Accessing the Template Editor and Template Libraries

Templates are stored and organized in Template Libraries. Every project that utilizes Corridor Modeling should have a unique *Project Template Library*. The Project Template Library is created by copying the *FLH Template Library* into the project file directory folder. The procedure for creating the Project Template Library is shown in *2E - Create the Project Template Library*.

The *FLH Template Library* contains pre-made Templates and Components that can serve as a starting point for creating Templates for the project. See *8B.2 The FLH Template Library*.

Assessing the Template Editor: Templates are created/edited in the *Template Editor*. The Template Editor is accessed with the *Create Template* tool:

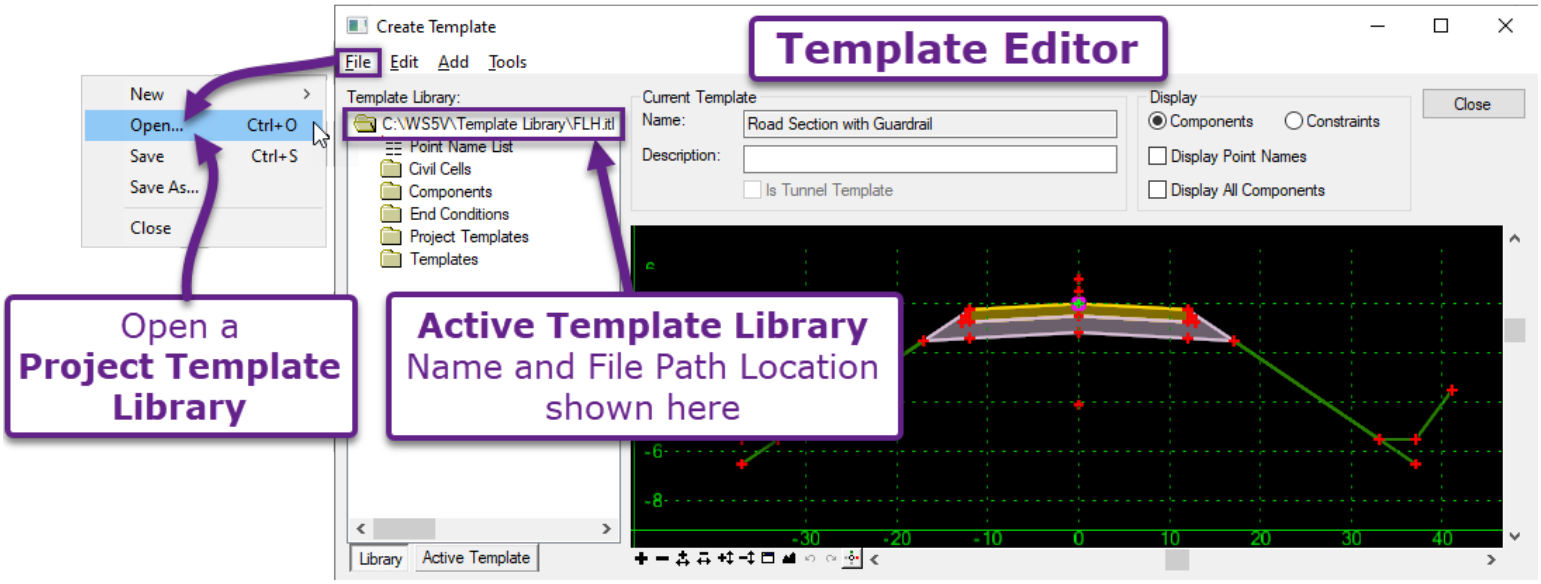
[**OpenRoads Modeling** workflow → **Corridor** tab → **Create** panel → **Template** drop-down]



When the Template Editor is accessed for the first time in a session, the *FLH Template Library* may be automatically loaded. The file name for the *FLH Template Library* is "FLH.itl" – which can be identified at the top of the *Template Library List* tree.

Do NOT make direct edits to the FLH Template Library. Instead, the Project Template Library should be loaded to create and edit Templates.

To Open a Project Template Library in the Template Editor:
File → Open... → *Navigate to project folder* → Highlight Project Template Library (.itl) → Open



8A.2 Basic Parts of a Template

There are two types of geometric entities that comprise a Template – *Points* and *Components.*

8A.2.a Template Points and Components

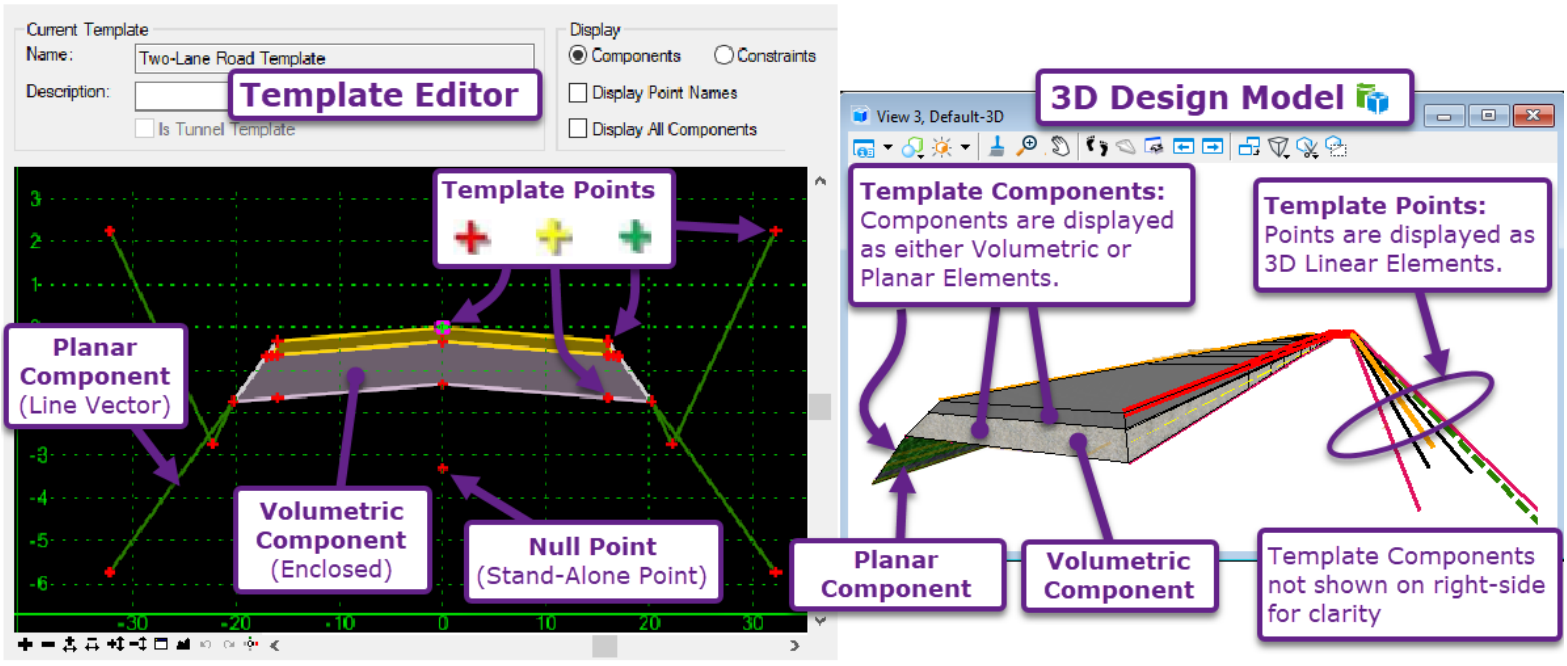
Points: Template Points are placed and repositioned to create the desired shape for a Template. Each Point has a set of *Point Properties* which are used to place the point in the desired position using *Constraints.* In the Template Editor, Points appear as red, yellow, or green crosses: .

From the perspective of the Template Editor, Points can be thought of as Linear Elements that run in and out of the page (or in this case: the computer screen). When a Corridor is created, a Linear Element is created for each Template Point. Additionally, each Template Point is assigned a Feature Definition that affects the display of a corresponding Corridor Linear Element. For more information on the Template Point Feature Definition and display of the resulting Corridor Linear Element, see *9C – Graphical Display of Corridor Geometry*.

Components: Template Components are shapes that represent different real-world *materials*, such as asphalt, concrete, and aggregate. In the Template Editor, a Component can appear as either as an enclosed shape OR as a line vector. As shown in the graphic below, enclosed components become *Volumetric* when the Corridor is created. Line vectors components become *Planar* when the Corridor is created. An example of a *Planar Component* is the green End Conditions, which represent the cut/fill embankment slopes.

Points and Components Interaction: The geometry of Components is dictated by the position of the Points. Each Component has an assigned set of Points. In the *Point Properties* menu, the selected point is shown as a *Member* of the Components it is assigned to.

> ***NOTE:*** *Null Points* are the exception. Null Points are NOT assigned to any particular Component and typically appears as a floating, stand-alone Point. For more information on Null Points, see *8C.1 Template Point Types and Identification*.

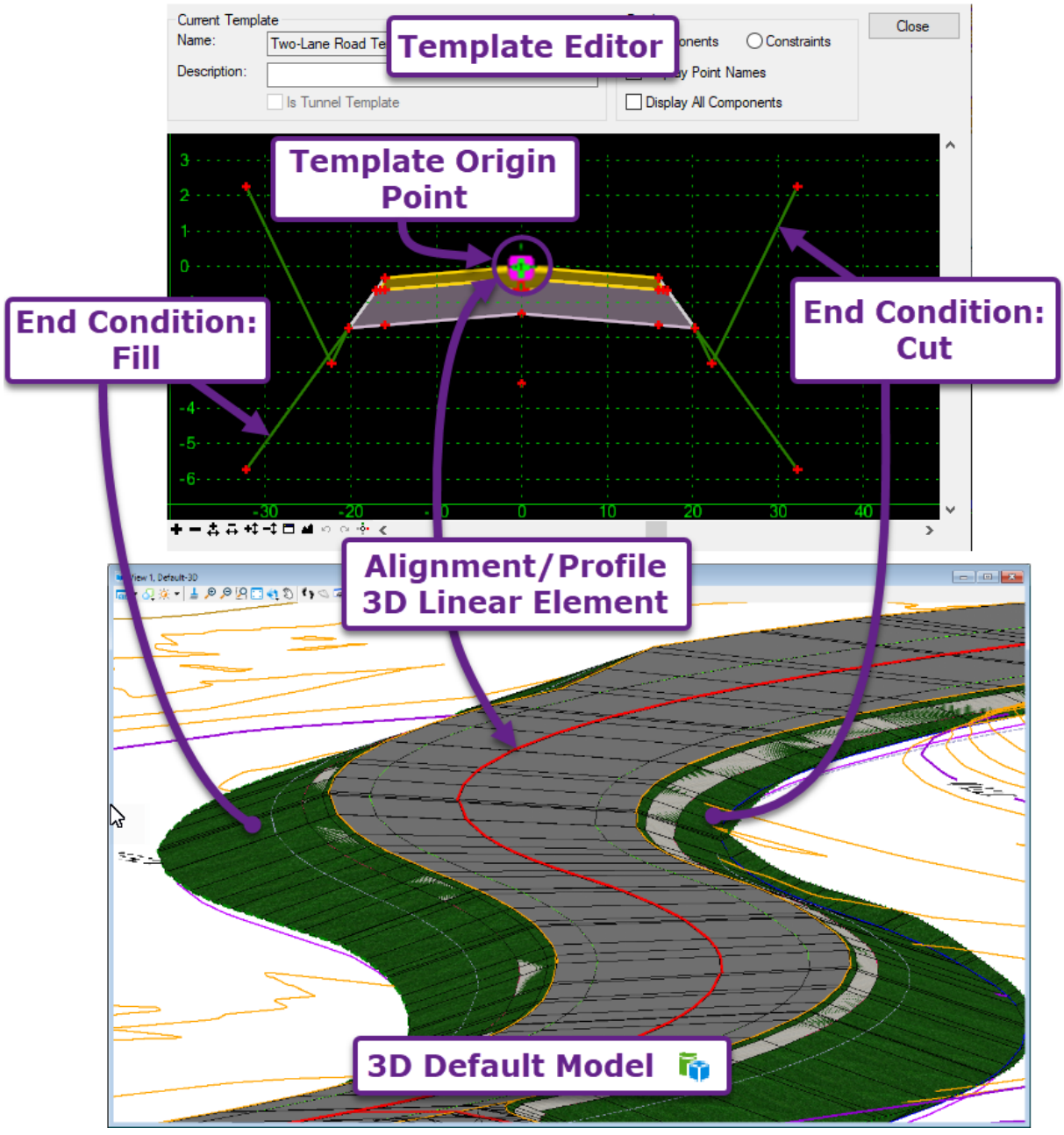


8A.2.b Template Origin Point and End Conditions

Template Origin Point: The Template Origin Point directly follows the selected Alignment/Profile when the Corridor is created. All other Points in the Template are positioned relative to the Template Origin Point through geometric *Constraints*. For example, a Template Point that represents the edge of road may be placed 12' horizontally at a -2% slope relative to the Origin Point. The horizontal and slope values fully constrains the geometric position of the edge of road Template Point relative to the Origin Point.

The Template Origin Point is marked with a magenta box. Additionally, the Template Origin Point should be shown as a green cross – meaning the Point is *Unconstrained*. For more information on the Constraint types (i.e., red, yellow, or green crosses), see *8C.6 Constraints*.

End Conditions: End Condition components are positioned on the outside of a Template representing cut and fill slopes. End Condition components are shown in green. The last Template Point in an End Condition extends or retracts to intercept the Existing Ground Terrain Model. For more information on End Components, see *8C.7 End Condition Template Points* and *8D.7 End Condition Component Properties*.

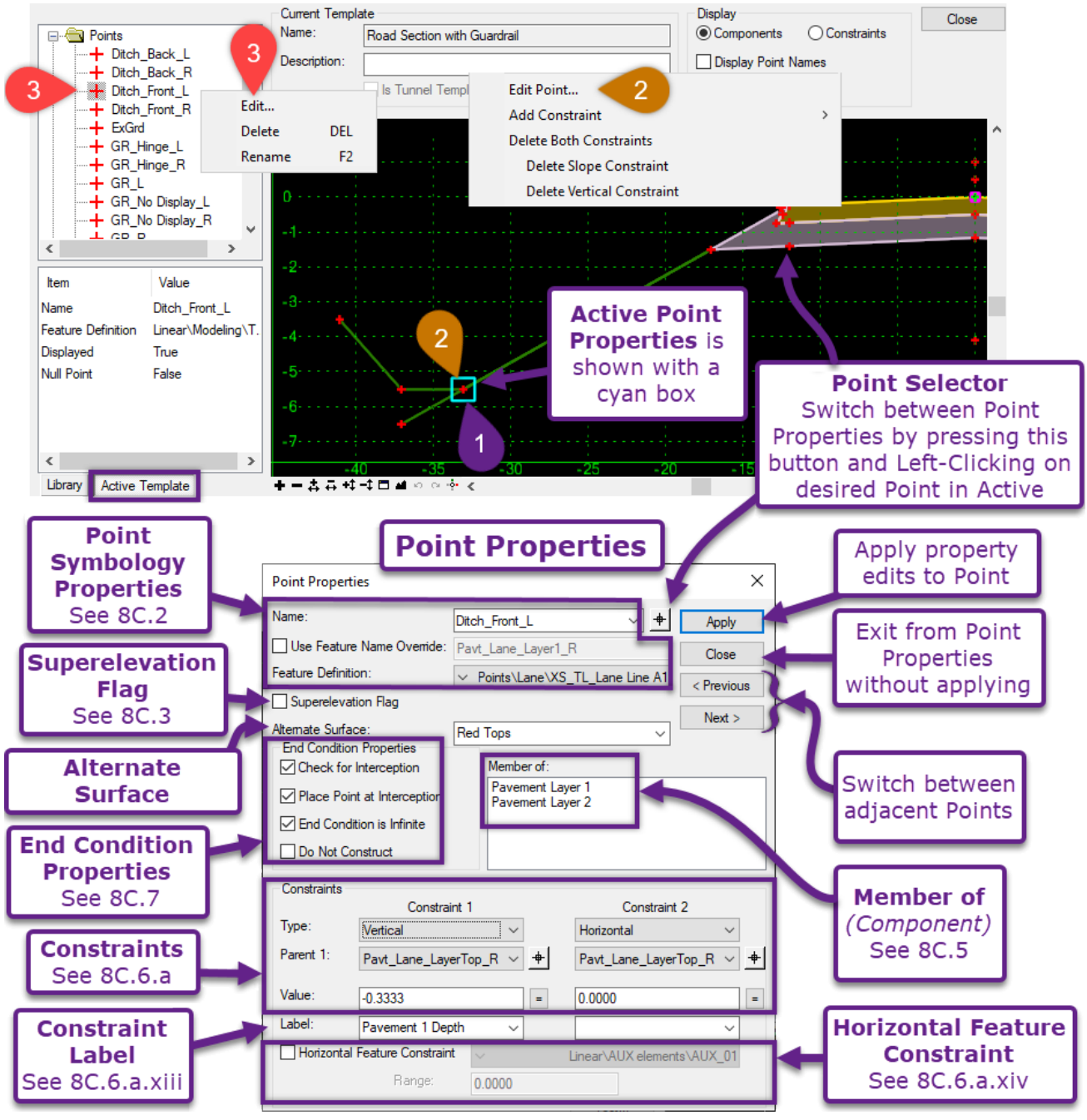


8A.3 Point and Component Properties

Through the *Point Properties* and the *Component Properties* menus, the position and behavior of Template Points and Components can be programmed. Double-click on a Point or Component to access its properties.

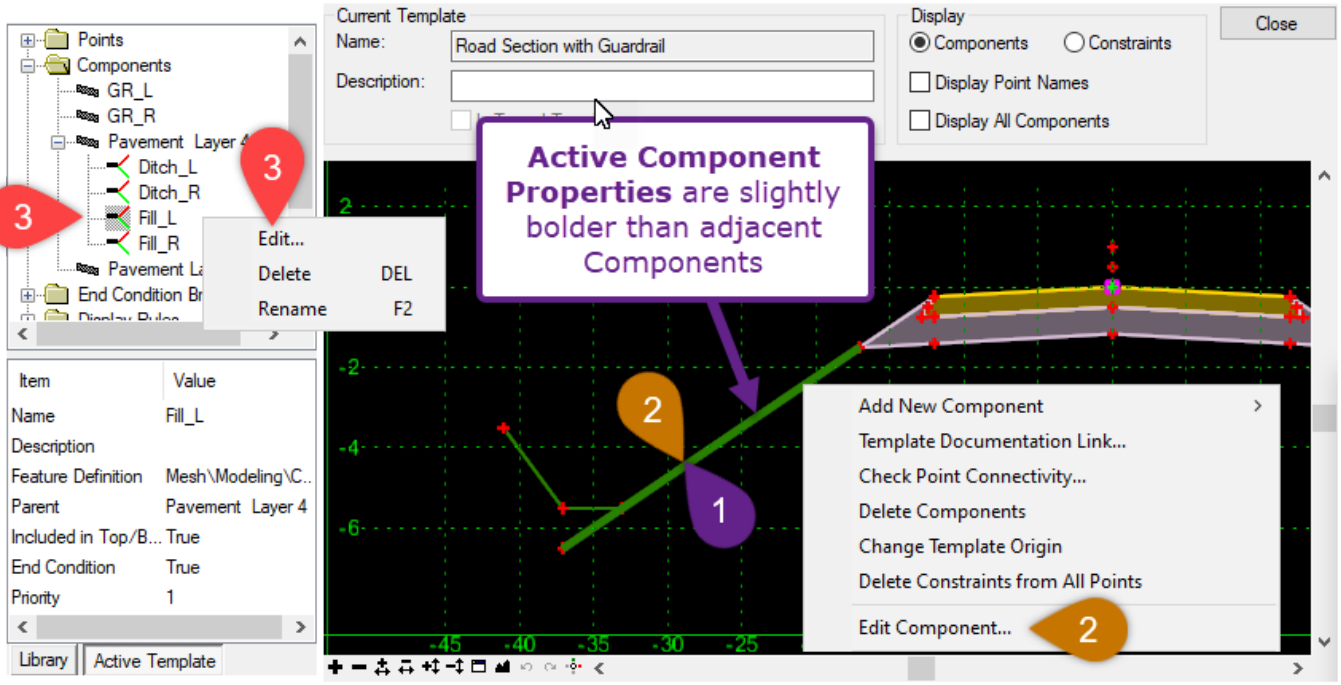
8A.3.a Access Point Properties and Overview

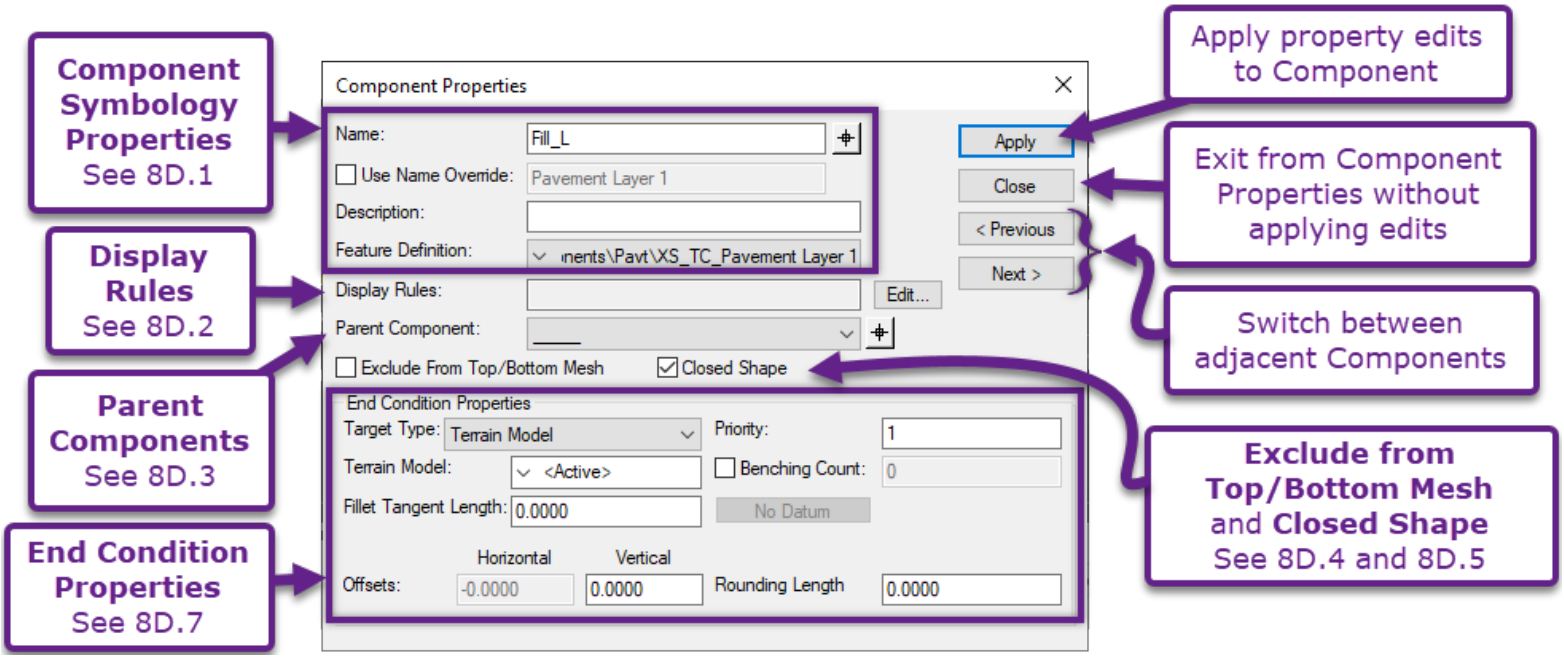
1	**Method 1:** Double-Click on the desired Point.
2	**Method 2:** Right-Click on the desired Point and select *Edit Point…*
3	**Method 3:** In the Active Template list, Right-Click on the desired Point and select *Edit…*



8A.3.b Access Component Properties and Overview

1	**Method 1:** Double-Click on the desired Component.
2	**Method 2:** Right-Click on the desired Component and select *Edit Component…*
3	**Method 3:** In the Active Template list, Right-Click on the desired Component and select *Edit…*





8A.4 Project Template Creation - Overall Workflow and Strategies

This flowchart details the **THREE STRATEGIES** recommend for creating a Template.

STRATEGY 1: Modify a pre-made Template from the FLH Template Library.

STRATEGY 2: Start with a blank Template. Assemble the Template using pre-made Template Components from the FLH Template Library.

STRATEGY 3: Start with a blank Template. Create a Template from scratch by using all new Template Points and Components.

Create a NEW Project Template Library

See *2E – Create the Project Template Library*

Open the Template Editor and load the NEWLY-CREATED Project Template Library

See *8A.1 Accessing the Template Editor and Template Libraries*

Determine the Project Typical Section

See *8A.4.a Project Template Considerations and Best Practice*

Note: See next page for PROS/CONS of all three STRATEGIES

Create the Project Template

STRATEGY 1	STRATEGY 2	STRATEGY 3
Modify a pre-made Template from the FLH Standard Template Library	Create from scratch by compiling and modifying pre-made Template Components found in the FLH Standard Template Library	Create from scratch by creating all new Template Points and Components
Identify the pre-made Template that is most similar to desired Project	**Create a new blank Template**	**Create a new blank Template**
Copy the pre-made Template into the Project Template Folder See *8B.3 Create a New Template or Edit a Pre-Made Template*	**Place Pre-Made Template Components into new Template** See *8B.4 Place Pre-Made Templates and Components into Active Template*	**Create New Template Components** See *8E.7 Create New Template Components*

Edit the Template as Necessary

See *8C – Template Points*
See *8D – Template Components*
See *8E – Creating and Manipulating Points and Components*

Template Creation Strategies		
Strategy:	**Pros:**	**Cons:**
Strategy 1: Modify a pre-made Template	• Assuming a pre-made Template like the project typical section can be found, this is the simplest method for Template creation. • The geometry is pre-configured, but Template Points can be easily repositioned by modifying Constraint values. • *Feature Definitions* and *Names* for Points and Components are pre-configured. This provides conformity with FLH drafting standards when calculating material quantities, working with FLH Pen Tables, and getting the corridor graphics linework to show correctly. • Very efficient if pre-made Template is similar to desired Project Template. • The FLH Standard Road Templates contain Display Rules that can accommodate approaches, intersections, and driveways. In the vicinity of approaches, the shoulder and End Condition (cut/fill) Components can be turned OFF.	• Pre-made Templates will have to be modified to a certain extent to meet Project conditions. • Project Template may behave unexpectedly if Point Constraints, Symbology, and Display Rules are not fully understood. • Pre-made Templates may have hidden Components (Pavement Layer 2 & 3, Leveling/Milling Components). • Can be more time-consuming if pre-made Template is dissimilar to desired Project Template. • When unwanted Points and Components are deleted, adjacent Template Points may become *Unconstrained* or *Partial Constrained* causing the Template to unravel.
Strategy 2: Create a Template from assembling pre-made Components	• Constraints are pre-configured. • Points and Component Feature Definitions and Names are pre-configured. • Template should behave as expectedly because it is manually assembled by the User. • Less likely to have hidden or unwanted Points and Components because the User selects all Components to add.	• It is likely that pre-made Components will have to be modified to a certain extent to meet Project conditions. • It may still be necessary to create Template Components from scratch if a particular Component is not found in the FLH Standard Library.
Strategy 3: Create a Template from scratch	• Template should behave exactly as the User expects because all pieces and parts are created and configured by the User.	• User has to manually set constraints and Template Point/Component *Feature Definitions and Names* – which can be very time-consuming. • Other Users to work on the Corridor/Template may be unfamiliar with the Template Construction if it is dissimilar to FLH standard road Templates.

8A.4.a Project Template Considerations and Best Practice

The Project Template should resemble and match the geometry of the Typical Section shown in the Plan Set. However, most roadway projects will have deviations to the Typical Section.

> **IMPORTANT:** It is not necessary to create a new Project Template for every deviation to the Typical Section.

Using advanced Template functionality in conjunction with Corridor Object tools (i.e., Point Controls and Parametric Constraints), a single Template can accommodate both significant and minor deviations to the Typical Section.

BEST PRACTICE: Use as few Project Templates as possible along the length of a Corridor. If a minor deviation to the Template is necessary, then attempt to accommodate the Deviation with Corridor Object tools or Display Rules. For more information on Corridor Object tools, see *9G – Corridor Objects – Manipulation of the Corridor*. For more information on Display Rules, see *8D.2 Display Rules*.

8A.4.a.i Considerations When Planning for the Project Template

1. **Determine the Typical Section geometry for the Project Template**
    a. Determine the typical pavement section and material configuration.
        i. Depths of materials – such as asphalt, concrete, and aggregate.
    b. Determine lane configurations and widths.
        i. Determine if a fog line, curve widening, and superelevation is needed.
    c. Determine Cut/Fill slopes and ditch configurations.

2. **Determine what sort of MINOR deviations to Typical Section may be needed**
    a. MINOR Deviations may include:
        i. Turn outs
        ii. Roadway Width deviations, such as curve widening
        iii. Superelevation
        iv. Pavement depth variation
        v. Steepen or flatten Cut/Fill slope, change in ditch depth
        vi. Steepen or flatten an aggregate shoulder

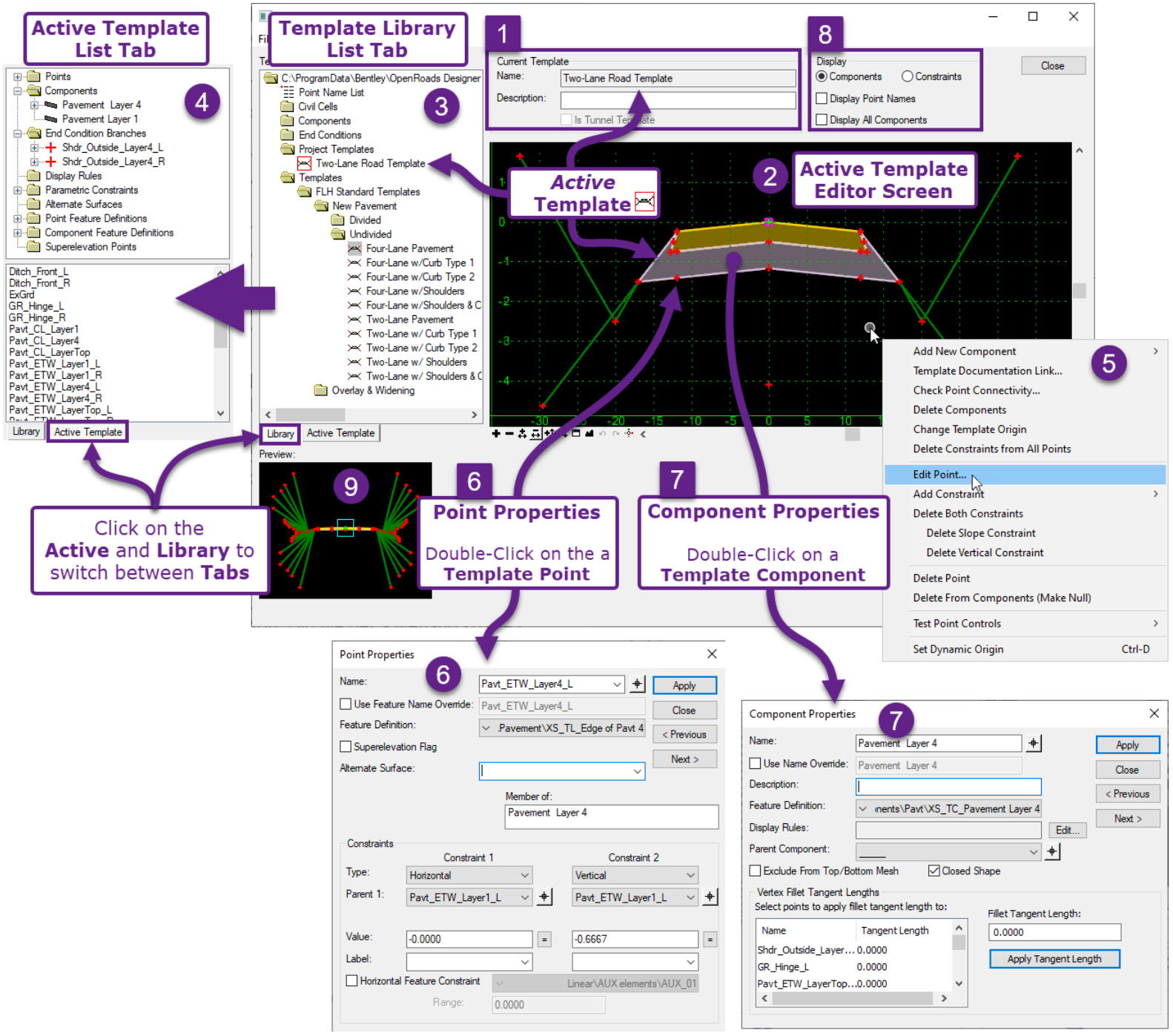
3. **Determine what sort of MAJOR deviations to Typical Section are needed**
    a. MAJOR deviations may include:
        i. Guardrail sections
        ii. Retaining Wall sections
        iii. Curb and Gutter sections
        iv. Bridge Sections

MINOR deviations to a Project Template can be addressed using *Corridor Object* tools after the Corridor has been created. See *9G - Corridor Objects – Manipulation of the Corridor*. *Corridor Object* tools allow the User to override widths, slopes, and depths for a Template over a specified station range. For example, to create a turn out – a paved shoulder simply has to be widened over a certain station range.

MAJOR deviations require additional Template Component configurations in the road Template. Example of major deviations include a guardrail sections or a short span of a retaining wall **MAJOR deviations** are accommodated using Display Rules, Null Point triggers, and Horizontal Feature Constraints to conditionally display (or not display) Template Components as necessary. An example of a single Template accommodating both the typical road section and guardrail sections is shown in *8F.3 Advanced Road Template with Guardrail and Display Rules*.

8B – THE TEMPLATE EDITOR AND TEMPLATE LIBRARIES

8B.1 Template Editor Overview



1	**Current Template**	Displays the name and a description of the *Active Template.*
2	**Active Template Editor Screen**	Graphically displays the Points and Components that comprise the *Active Template*. Edits to the *Active Template* are typically made by double-clicking on Points and Components to alter their Properties and Constraints.
3	**Template Library List Tab**	Displays all different Templates that are available in the currently-loaded Project Template Library. The Template that is currently *Active* will have a rex box next to it . Double-Click on a Template in the List to make it Active. At the bottom of this list, click the *Active* tab to switch to the **Active Template Tab**.
4	**Active Template List Tab**	All parts of the *Active Template* are listed in the **Active Template List Tab**. The different parts that comprise a Template include: Points, Components, End Conditions, Display Rules, Alternate Surfaces, and Feature Definitions. Through the **Active Template List**, double-click on a Point or Component to open its properties. Click the *Library* tab to switch back to the **Template Library List Tab**.
5	**Right-Click Menu**	Right-Clicking in the Active Template Editor and Template Library List will provide access to a plethora of manipulation tools. **The tools available through the Right-Click Menu depend on the Right-Click location in the Template Editor.** For example, Right-Clicking on a Template Point will bring up different tools than Right-Clicking on a Component.
6	**Point Properties**	Edits to a Template Point position are made from within the Point Properties. Double-click on a Template Point to access properties. See *8C – Template Points*.
7	**Component Properties**	Edits to a Template Component are made in the Component Properties. Double-click on a Template Component to access properties. See *8D – Template Components*.
8	**Display Options**	Controls appearance of Active Template Editor Screen. The Display Options are used to switch between Component and Constraint views. See *8C.6 Constraints*. Also, used to turn off Point Names display.
9	**Template Preview**	Templates that are highlighted in the **Template Library List** are shown here. Left-Click on a Template in the **Template Library List** to show it in the **Template Preview.** Double-Click on a Template to make it *Active*

8B.2 The FLH Template Library

The FLH Template Library contains pre-made roadway Templates that can be modified to fit the project. Additionally, the FLH Template Library contains pre-made Components, which can be assembled by piecemeal to form a Template.

NOTE: A new FLH Template Library has been released in FLH WorkSpace 10.10.21.00V. The new Template Library is significantly updated and reorganized when compared to the old FLH Template Library found in FLH WorkSpaces prior to the 10.10.21.00V update. **Always use the FLH Template Library that corresponds with the FLH WorkSpace version used on the project.**

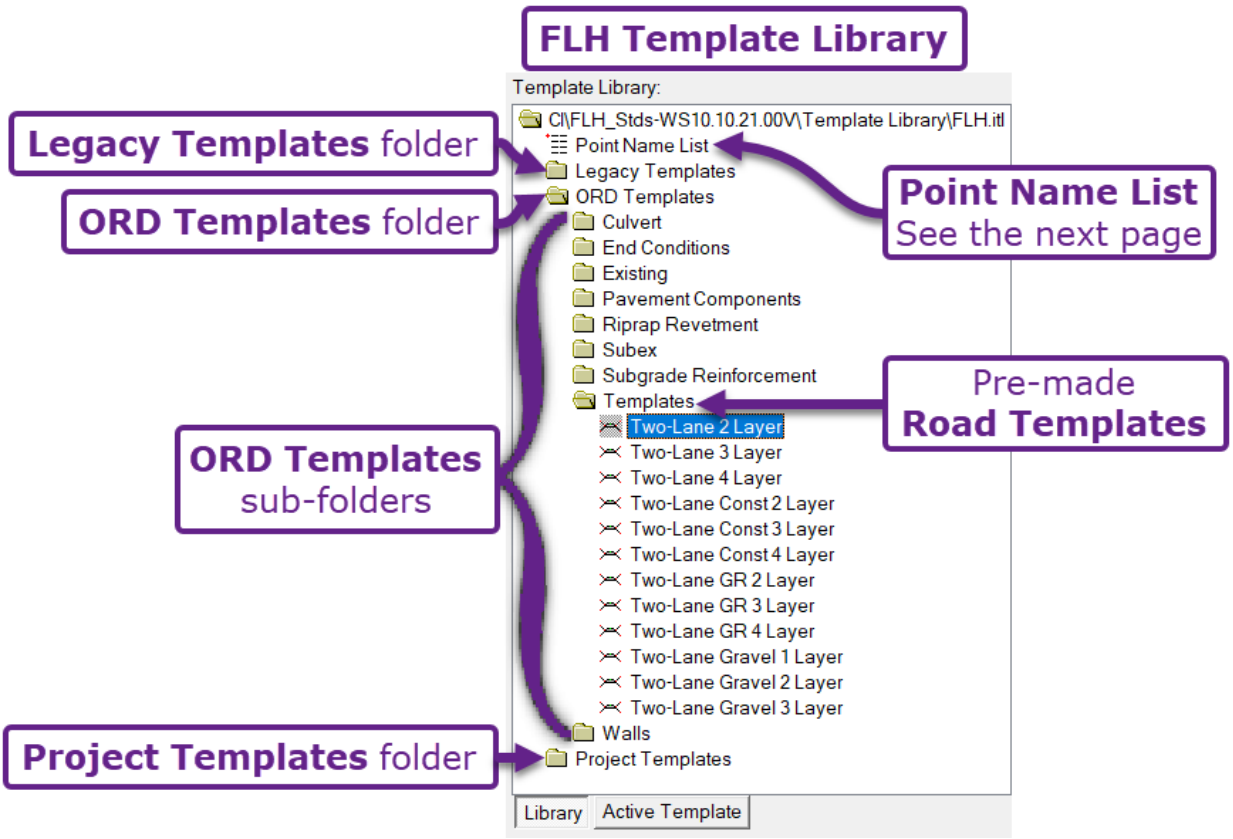
WARNING: The FLH Template Library is a shared resource. Do NOT make direct edits to the FLH Template Library. Instead, make a copy of the FLH Template Library (.itl) file for the specific project. Before making project Templates, copy the FLH Template Library (.itl) into a project file directory as shown in *2E – Create the Project Template Library*.

The FLH Template Library is divided into **three** main folders:

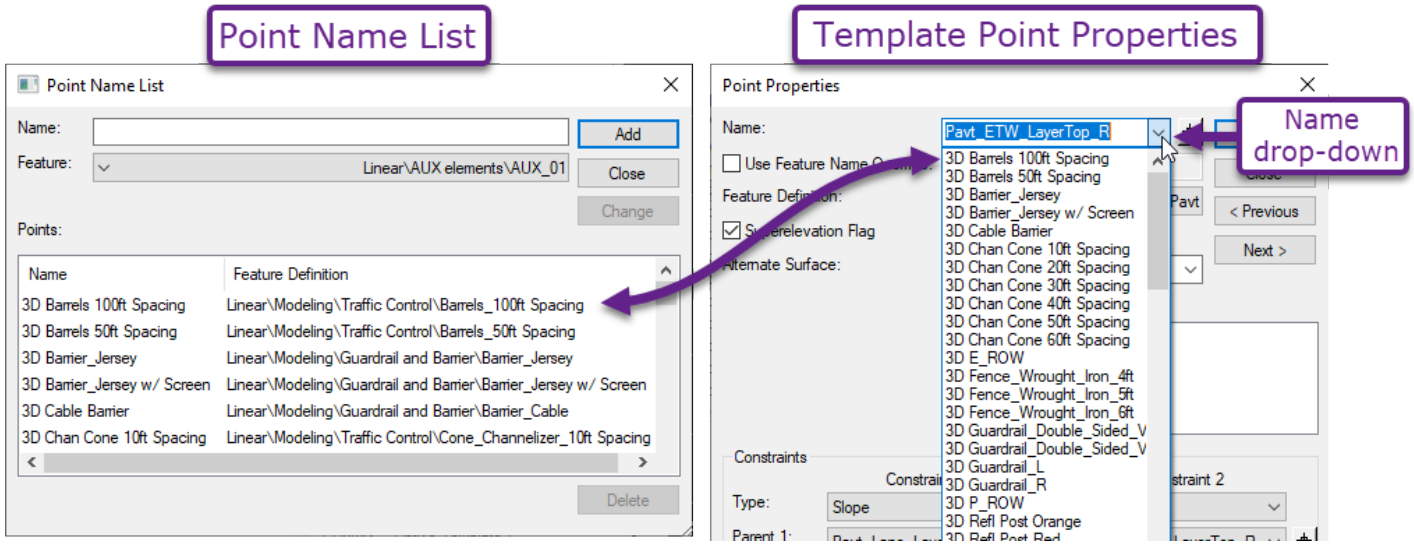
Legacy Templates: This folder contains all Templates found in the old FLH Template Library used prior to FLH WorkSpace 10.10.21.00V. Templates found in the Legacy Folder can be used for projects that use the FLH WorkSpace 10.10.21.00V update and future updates.

ORD Templates: This folder contains pre-made Templates and Components. Roadway Templates are found in the **Templates** sub-folder. All other sub-folders contain Components and pre-made assemblies that can be added to the project Template or compiled to form a new Template by piecemeal.

Project Templates: This folder is intentionally empty. This folder should not be used until the Project Template Library has been created. This folder is intended to store Templates specifical created for a project. Templates from the other folders should be copied into the Project Templates folder before editing.



Point Name List: This list **c**ontains pre-made Point Names that were created by FLH. Each Point Name has a corresponding Feature Definition. When manually creating Template Points and Components, a pre-made Point Name can be selected in the *Template Point Properties* menu to set the Name and Feature Definition.

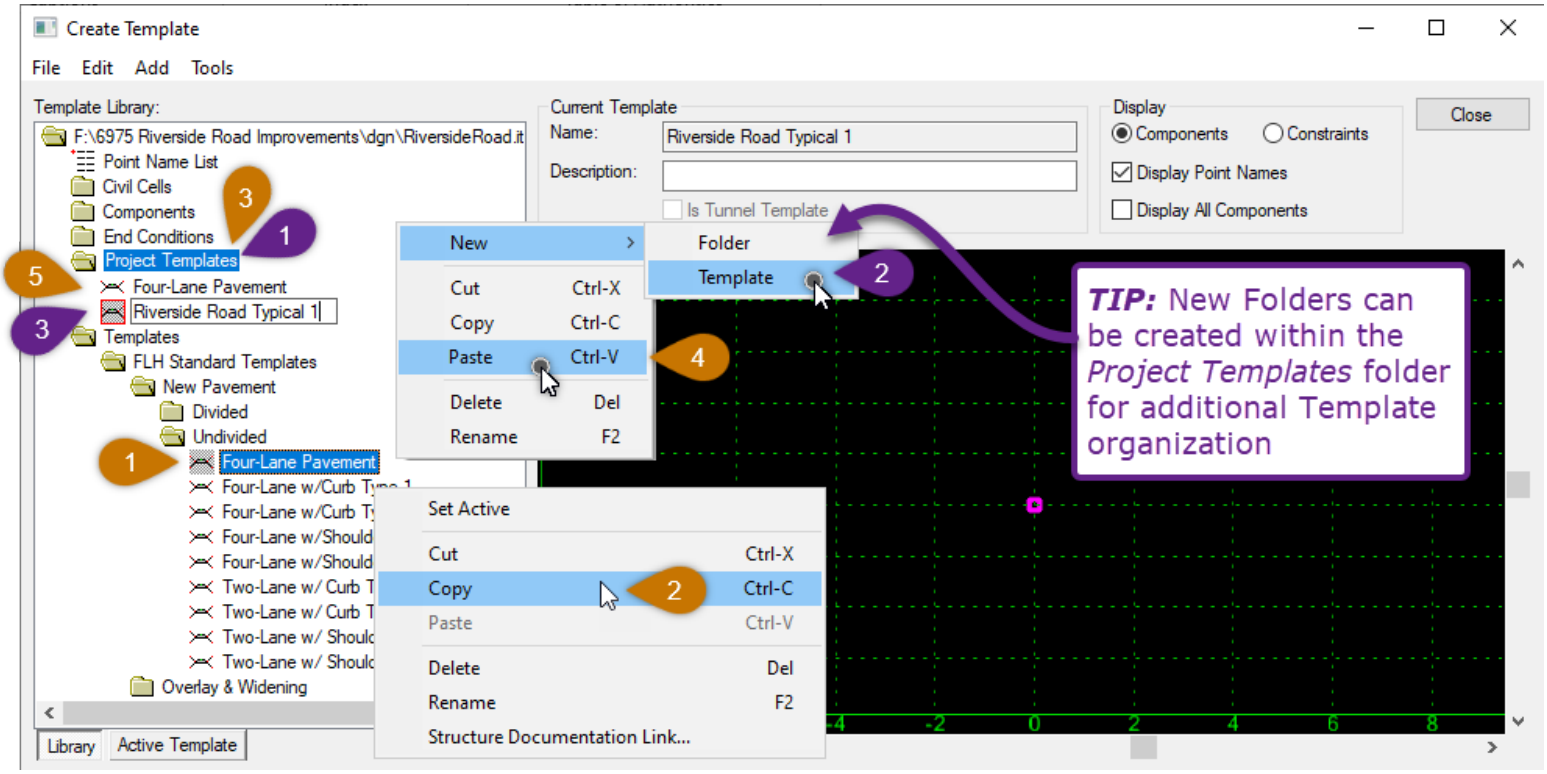


8B.3 Create a New Template or Edit a Pre-Made Template

After the appropriate Project Template Library is loaded – Template editing and creation can begin.

In general, there are two methods to approach Template creation

1. **Create a blank new template** in the Project Templates folder. Components are made from scratch or pre-made components are dropped in.
2. **Copy a pre-made Template** and paste it in the Project Templates Folder.



Create a blank New Template

1	In the Template Library List, Right-Click on the *Project Templates* folder.
2	In the Right-Click Menu, create a New Template by selecting *New → Template.*
3	Assign the Template an appropriate Name.

Copy a Pre-Made Template

1	In the Template Library List, Right-Click on the *Pre-Made Template* to be copied.
2	In the Right-Click Menu, select *Copy.*
3	Right-Click on the **Project Template** folder.
4	In the Right-Click Menu, select *Paste.*
5	Assign the Template an appropriate Name.

8B.4 Place Pre-Made Templates and Components into Active Template

Any pre-made Template or Template Component in the Template Library List can be dragged and placed into the Active (current) Template. A common strategy for Template creation is to start with a blank Template and drag Components from the Template Library into the new Template.

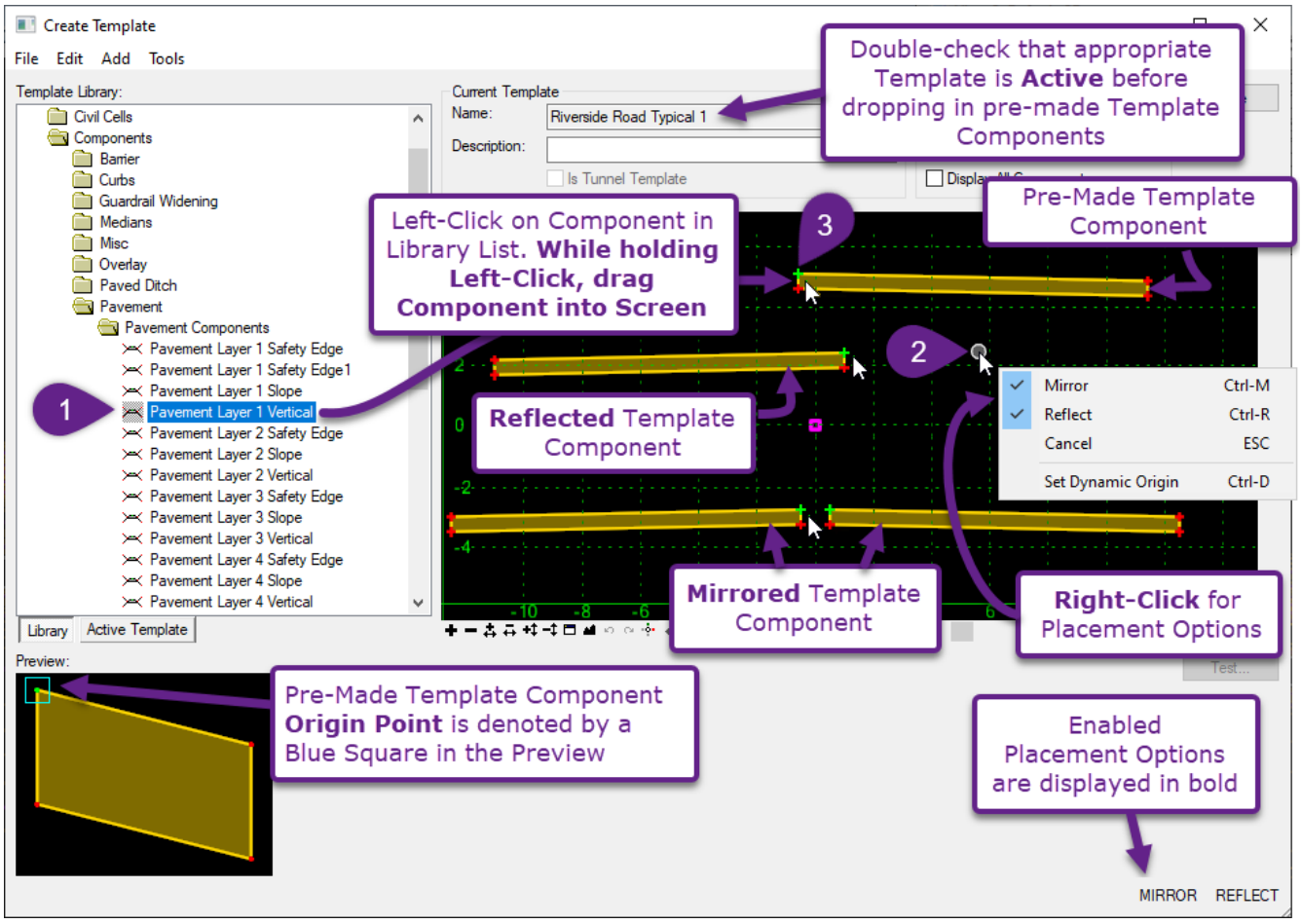
TIP: Before placing pre-made Template Components, set the **Suffix** and **Step Options** as shown in *8B.4.a Placement Options for Pre-Made Template Components*.

By *Right-Clicking* during Template Component placement, additional Placement Options are available:

Reflect: The Template Component is flipped around.

Mirror: Both the Template Component and a mirror copy are placed.

Set Dynamic Origin: When enabled, the Origin Point of the pre-made Template being placed becomes the Origin Point of the Active Template.



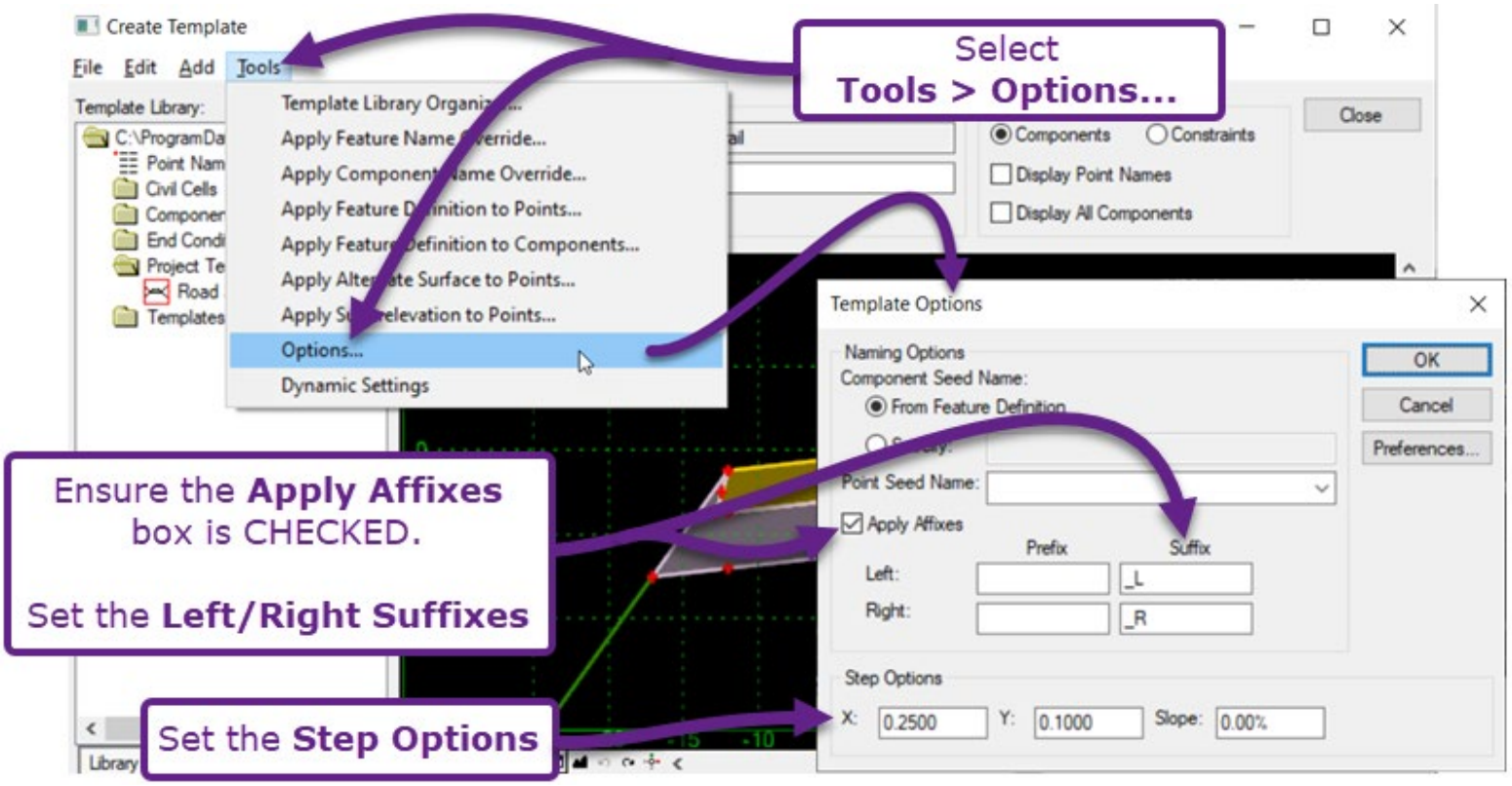
To place a Pre-Made Template Component into the Active Templates:

1	In the Template Library List, Left-Click (and hold) on the Template Component to be placed.
2	While holding down the Left-Click button, drag the Template Component into the Active Template Editor Screen. If desired, Right-Click (while still holding down the Left-Click button) to access Placement Options. Select desired Placement Options.
3	When satisfied with Template Component position and orientation, release the Left-Click button to place the Template Component.

8B.4.a Placement Options for Pre-Made Template Components

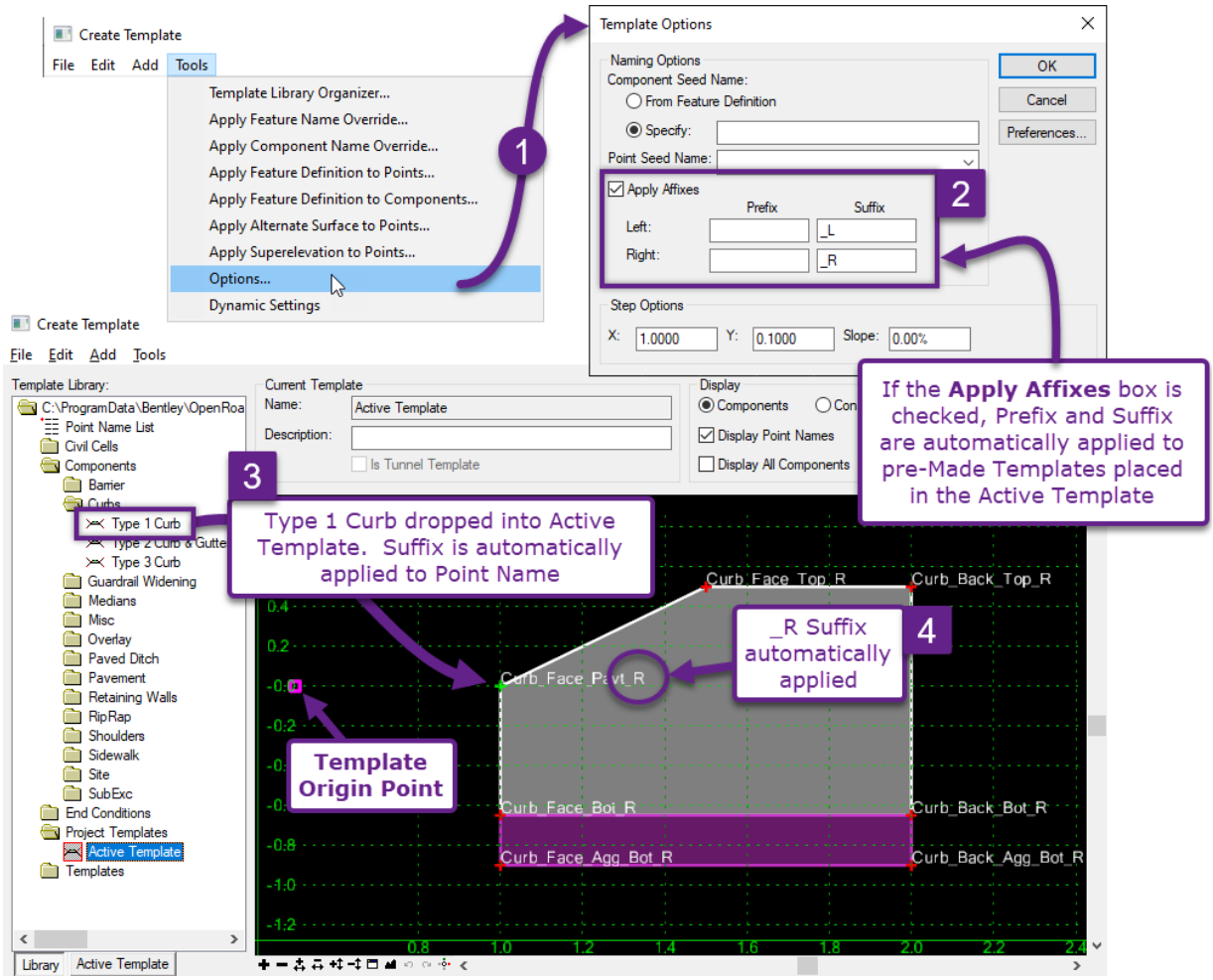
Before placing pre-made Template Components into the Active Template, the Template Placement Options should be set. The Template Placement Options are accessed through the *Tools* drop-down in *Options…* menu.

It is recommended that **Apply Affixes** box is CHECKED and the Suffixes are set to "_L" and "_R". Also, ensure the Step Options are set to appropriate values.

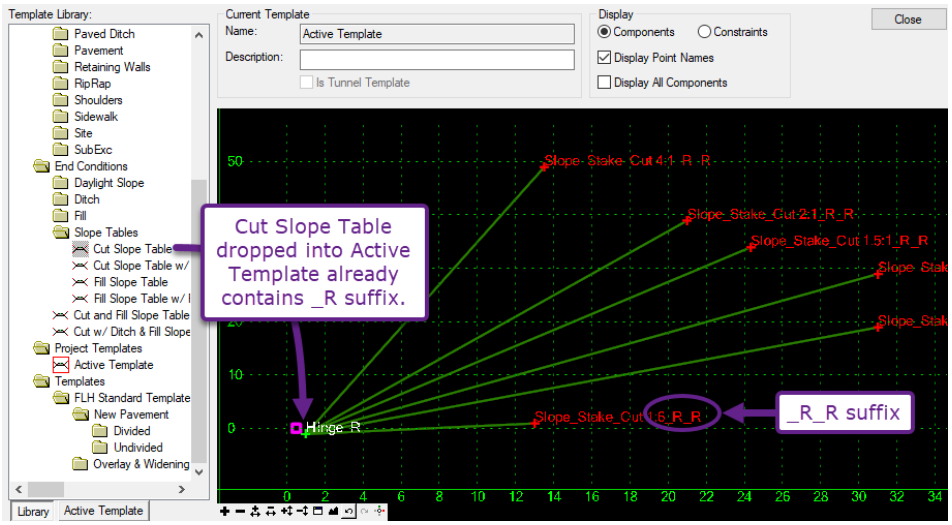


8B.4.a.i Point Name Organization (Affixes)

When placing pre-made Template Components, suffixes and prefixes (_R or _L) can be automatically applied to Template Point names. The affix, _R or _L, will be automatically applied depending on the side the Template is placed relative to the Origin Point of the Active Template. Enabling the **Apply Affixes** options helps to reduce effort and time in naming Template Points.

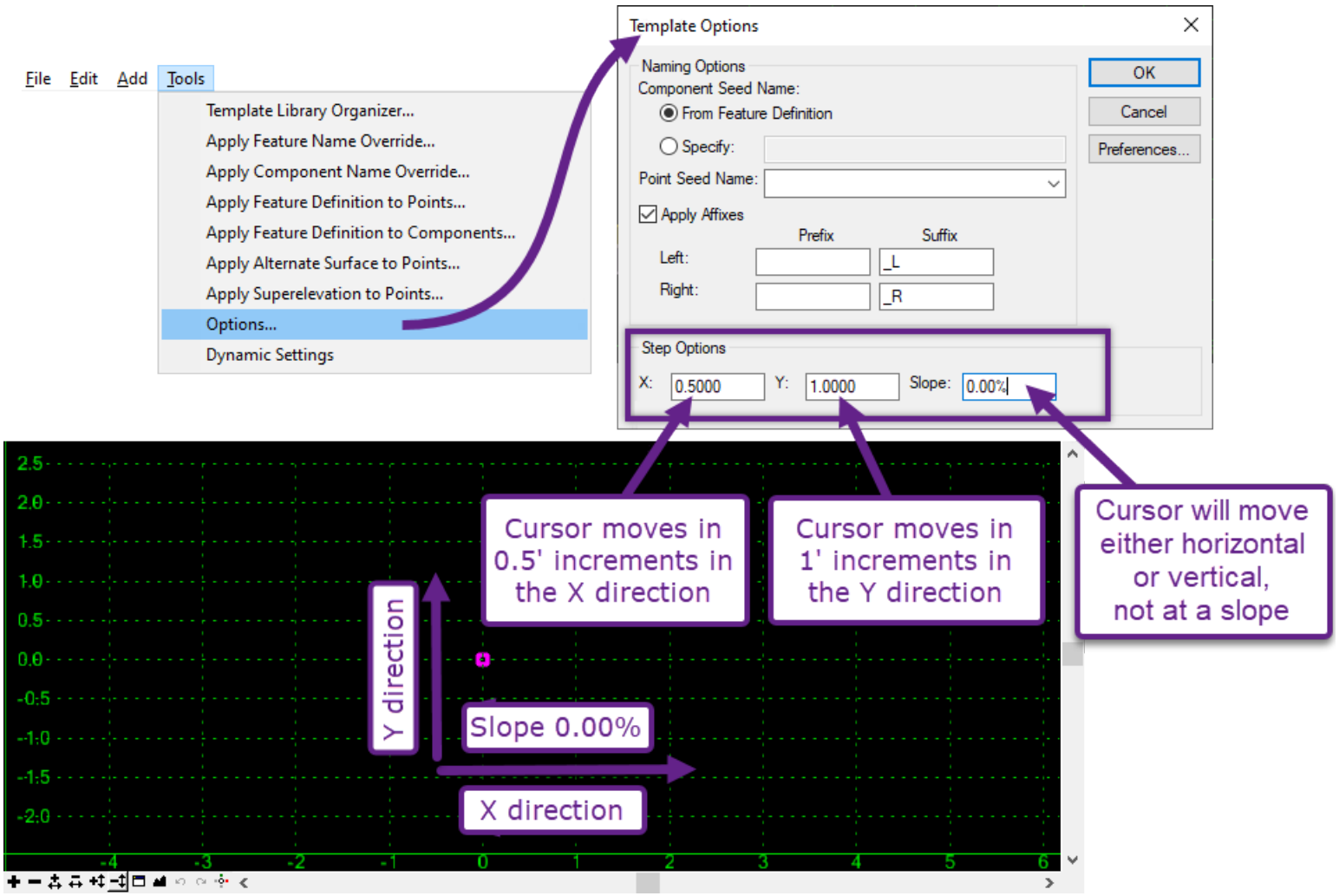


WARNING: When the *Apply Affixes* box is checked, ensure that the pre-made Template Component being placed does NOT already contain suffixes. If so, the Template Point name will contain a repeated suffix (i.e., "…_R_R").



8B.4.a.ii Step Options

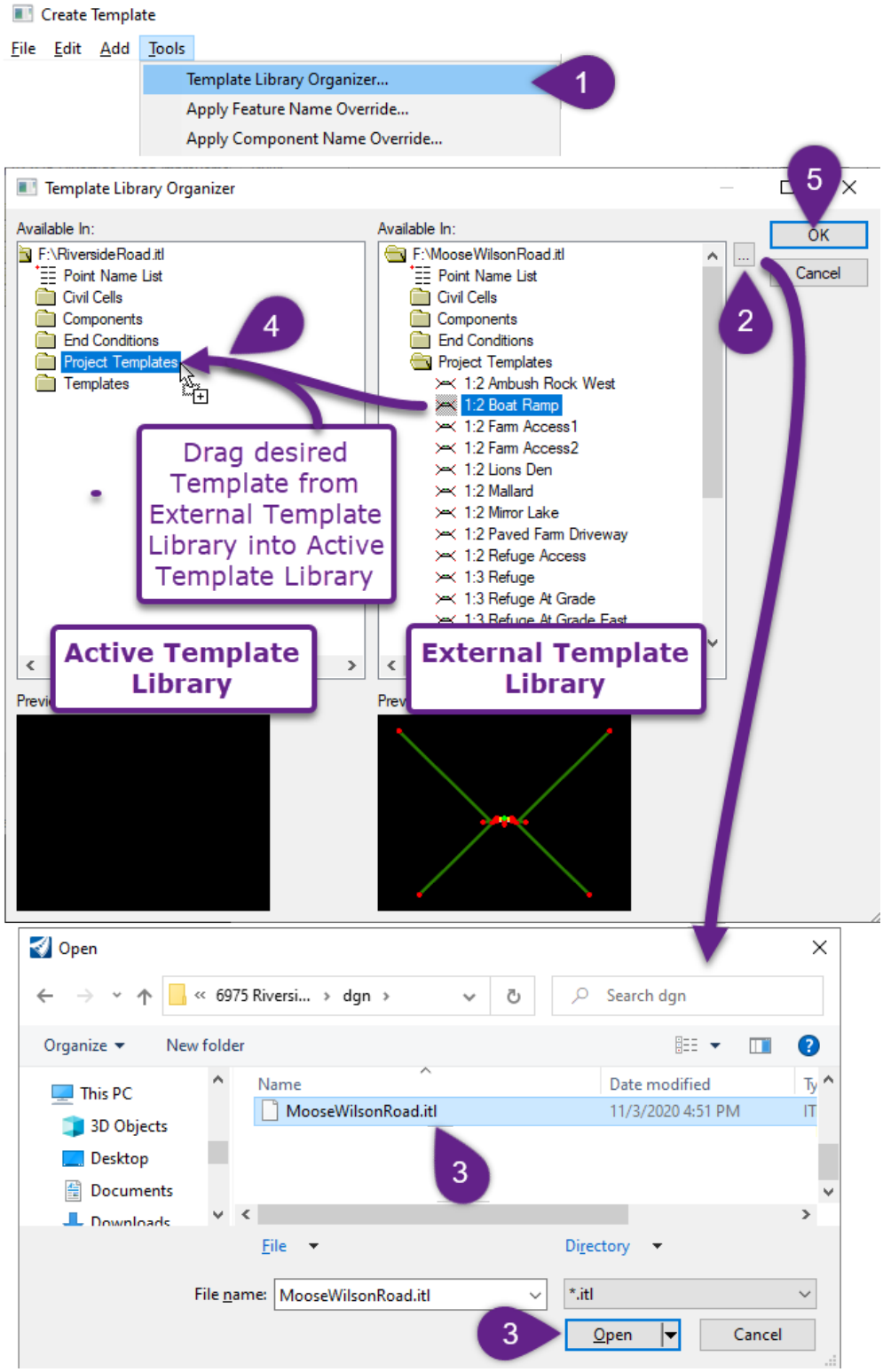
When Step Options are set at values greater than zero, the mouse-cursor will move around Active Template Editor Screen grid in *steps* – as opposed to a fluid motion. This is useful when placing pre-made Template Components and *moving* Template Components in the Active Template. By specifying non-zero Step Options, the mouse-cursor will snap to the grid and other Template Points.



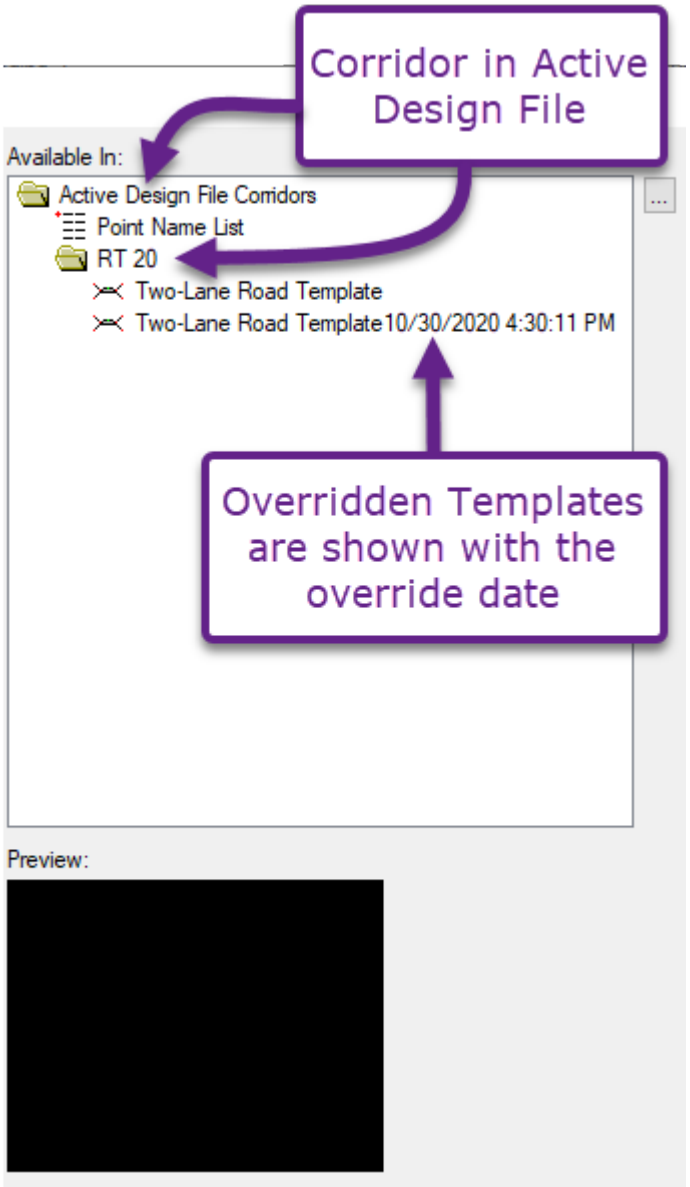
8B.5 Transfer Templates between Project Template Libraries

The *Template Library Organizer* tool is used to transfer Templates between two Project Template Libraries. This could be used if a custom Template used for an old project is desired for use on a new project.

This tool can also be used to extract overridden Templates from a Corridor into a Project Template Library. This is convenient when the *Edit Template Drop* tool is performed directly to a Corridor to override a Template. Templates overridden with *Edit Template Drop* tool are shown in the *Active Design File Corridor* list.



1	In the Template Editor, navigate to the *Template Library Organizer* tool from *Tools* drop-down.
2	Select the … button to load an **External Template Library**.
3	In the *Open* window, navigate to the location of the External Template Library, highlight it, and select *Open*. Do NOT select the active Template Library
4	In the External Template Library list, locate the Template to be transferred to the Active Template Library. Left-Click on the Template (hold down Left-Click) and drag the Template into the appropriate Folder in the Active Template Library List
5	Select *OK* to complete the transfer.



8C – TEMPLATE POINTS

8C.1 Template Point Types and Identification

There are three Template Point types: **Conventional Points**, **Null Points**, and **End Condition Points**.

Conventional Points: Conventional Points are referred to as Points or Template Points in this manual. Conventional Points are ALWAYS assigned/associated to one or more Components. Unlike **Null Points**, Conventional Points CANNOT be "floating" (appear disconnected from a Component).

In a cross-sectional view, Conventional Points remain in a fixed position as determined by the assigned Constraints. The exception to this rule is when the Point is subjected to Point Control, Parametric Constraints, and/or Horizontal Feature Constraints. In other words: Conventional Points will not move from their assigned positions except when under the influence of *Corridor Object* tools, such as *Point Control*.

Null Points: Null Points operate the same as Conventional Points – but are NOT assigned to any Component. They can be thought of as stand-alone or "floating" Points. Null Points are commonly used to represent features that are difficult to characterize with planar or volumetric components, such as guardrail. Null Points are commonly used to trigger Display Rules ,which are used Advanced Template Creation.

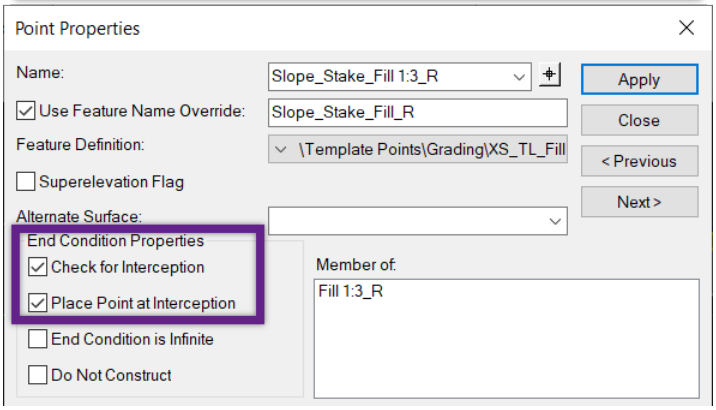
End Condition Points: End Condition Points are ONLY found on End Condition Components. End Conditions Points have the ability to adjust position to intercept the intended *End Condition Target*. The most common End Condition configuration is Cut/Fill Points seeking out the Existing Ground Terrain Model. The types of Targets available are as follows:

> **Terrain Model:** The End Condition Point will intercept and be placed on the active Terrain Model. This is the default Target for End Condition Points in the FLH Template Library.
>
> **Elevation:** The End Condition Point will extend or contract to terminate at a set elevation.
>
> **Linear Element:** The End Condition Point will adjust to intercept an ORD Element placed in the *2D Design Model* . An example would be manually creating a ditch alignment and profile. The End Condition Point will seek out the ORD Element representing the ditch.

NOTE: NOT all Points in an End Condition Component will seek out a target (i.e., the Existing Ground Terrain Model). End Condition Points will only terminate at a Target if the **Check for Interception** and **Place Point at Interception** boxes are CHECKED in the Point Properties box. If these boxes are UNCHECKED, then the End Condition Point will behave similarly to Conventional Points.

For a detailed explanation of each End Condition Property, see *8C.7 End Condition Template Points*.

End Condition Point Properties

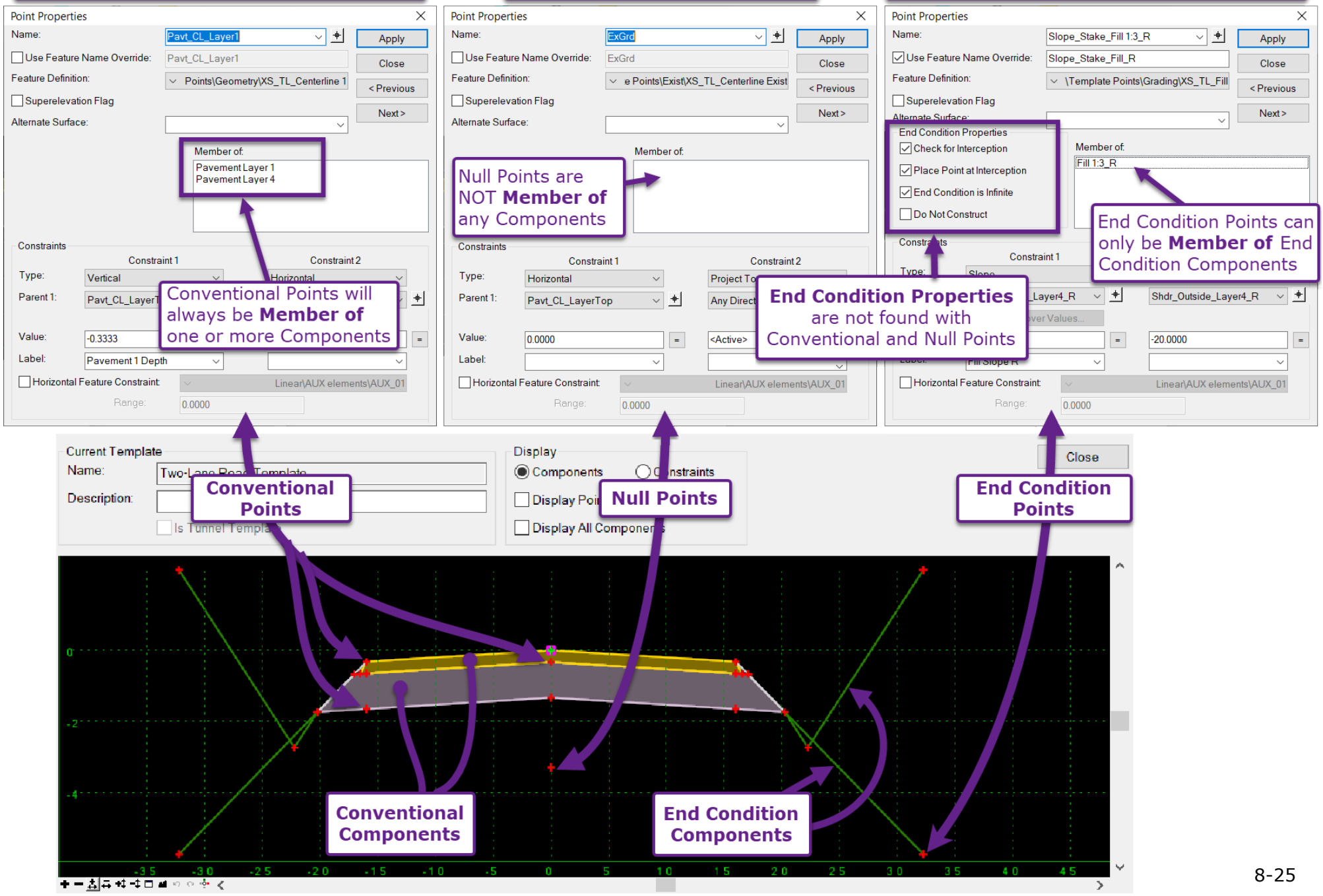


Template Point Type Identification

Conventional Point Properties

Null Point Properties

End Condition Point Properties



8C.2 Point Feature Definition and Name Properties

When the Corridor is created, a Point's Feature Definition and Name affects the display of the corresponding linear element in the *2D* and *3D* *Design Models*. For a detailed explanation of the display of Template Points/Linear Element after Corridor creation, see *9C – Graphical Display of the Corridor Geometry*.

1 **Feature Definition:** The Feature Definition assigned in the Template Point Properties determines the Feature Definition of the corresponding Linear Element in both the *2D* and *3D* *Design Models*. The Feature Definition sets the Level, Color, Line Weight, and Line Style.

The Feature Definition for a Template Point determines if a 2D Complex Element is created. All Template Points will generate a 3D Linear Element in the *3D Design Model*. However, only a few Feature Definitions will generate a 2D Complex Element in the *2D Design Model*. For more information, see *9C.4.a* Effect of Feature Definition and Feature Name for a Complex Element.

2 **Name:** The Name is used to identify a Template Point. Every Point in a Template MUST have a unique *Name*.

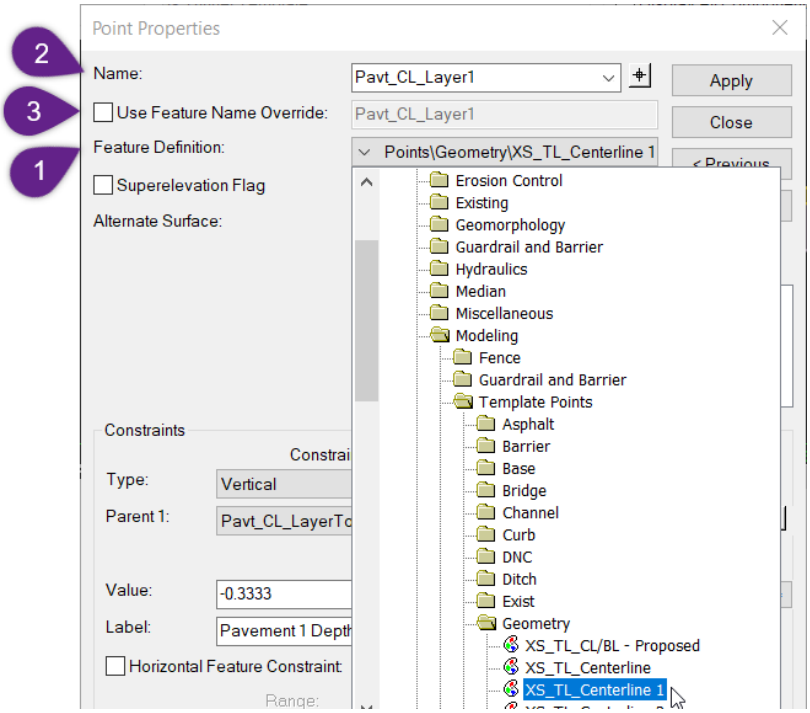
Naming Requirements: Each Template Point must contain an appropriate suffix (_L, _R, or _CL):

_L = left of Template Origin **_R** = right of Template Origin **_CL =** in line with Template Origin

The Name of a Template Point should be logical and but descriptive. An example of an appropriate name is *Pavt_Outside_Layer1_R*. This name describes the material (Pavt), horizontal position (Outside), vertical position (Layer 1), and the side it falls on (_R).

3 **Use Feature Name Override:** When this box is checked, the *Name* is overridden to whatever is typed into the *Use Feature Name Override* box. The *Feature Name Override* is used to give multiple Points the same Name, which has a significant effect on how Template Points are joined together when the Corridor is created. For more information, see *9C.4.a.ii Effect of Template Point Name Overrides on Corridor Complex Elements*.

Name Overrides must be applied to all End Condition Template Points to be displayed in the 2D Design Model correctly. For more information, see *8C.7.b Feature Name Override Convention for End Condition Points*.



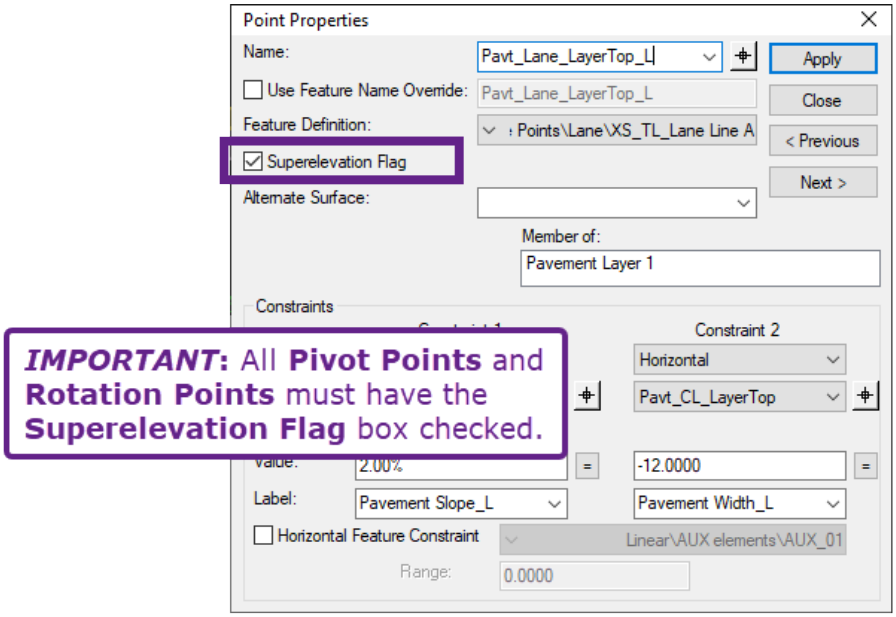
8C.3 Superelevation Flag

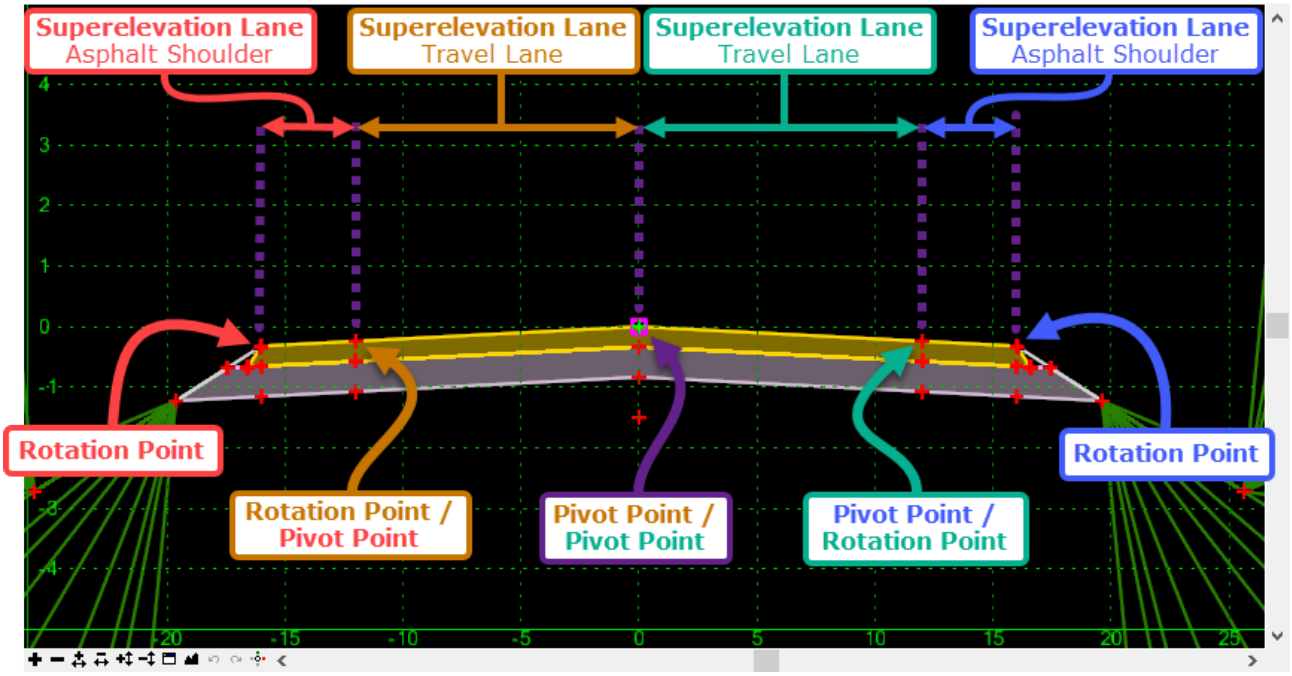
When the Superelevation Flag box is CHECKED, the Template Point is eligible for rotation in superelevation applications. See *Chapter 10 – Superelevation*. The Point will not be superelevated until the processes in *Chapter 10* are performed.

NOTE: Typically, the Superelevation Flag box is checked only for the Pivot and Superelevation Points needed to define a Superelevation Lane. See *10A.1 Superelevation Lanes and Superelevation Flag*. **Do NOT place the Superelevation Flag on Points other than the Pivot and Superelevation Points.**

WARNING: The Constraints of adjacent Template Point should be understood and considered when applying Superelevation Flags. Always predict how adjacent Template Points will react when a Point is superelevated. Inciteful

TIP: The *Test Point Control → Test Vertical Point Control* tool can be used to simulate Superelevation application within the Active Template Editor Screen. Right-Click directly on the Superelevation Template Point and select: *Test Point Control → Test Vertical Point Control*.





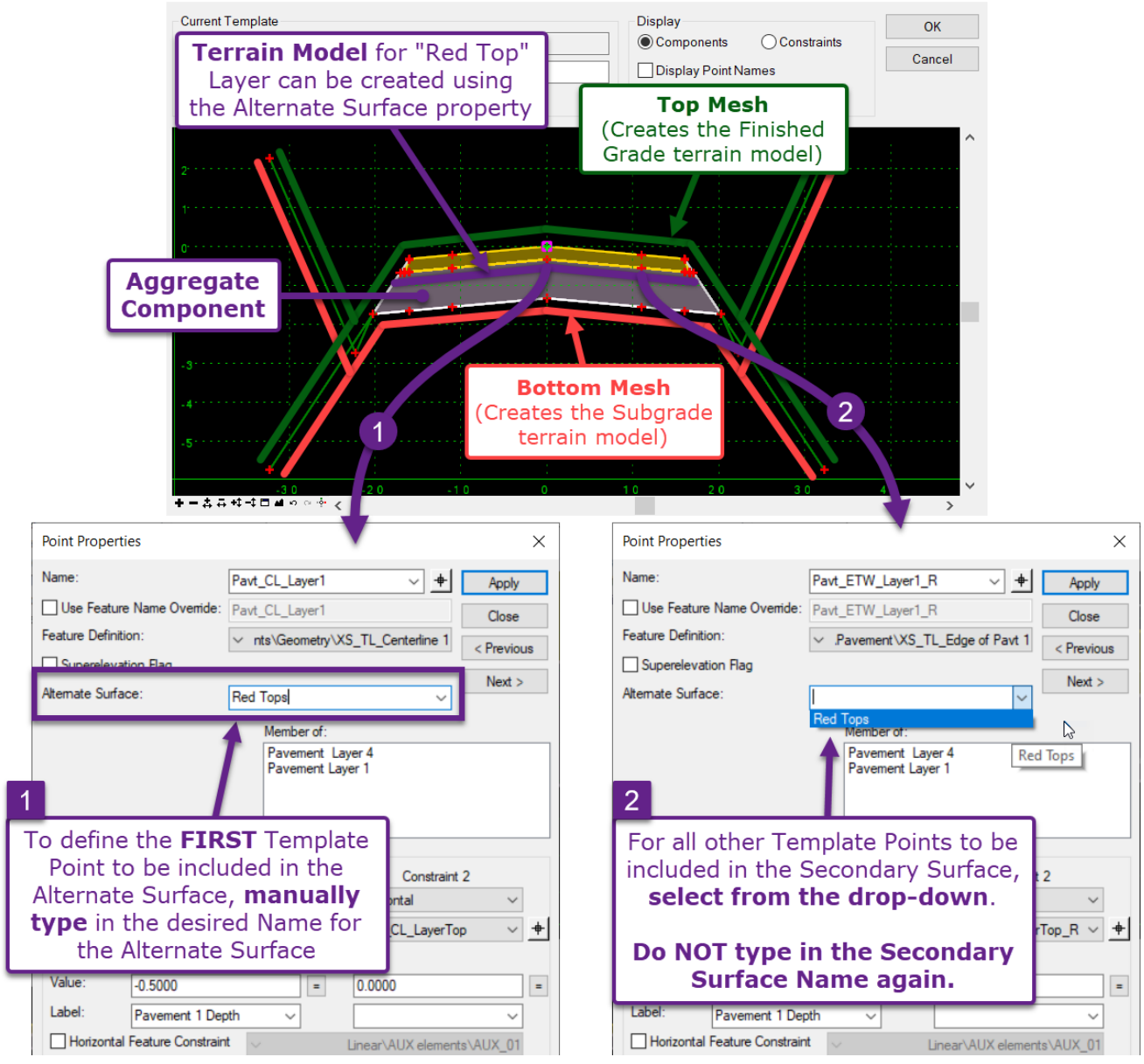
8C.4 Alternate Surface

The software automatically creates a Top and Bottom Mesh from a Template. As shown in the graphic below, the Top Mesh is comprised of the string of Points that traces the top surface of the Template. The Bottom Mesh traces the bottom surface of the Template. Creating proposed Terrain Models from the Top and Bottom Mesh is shown in *Chapter 22 – Proposed Terrain Model Creation*.

The Alternate Surface property allows the User to specify a custom string of Points to create an *Alternate Surface* Terrain Model. An example of a custom string of Points is shown in the graphic below. The Points that comprise the top of the aggregate Component (or "Red Tops" Layer) are strung together to create a custom Terrain Model, which could be used in construction staking.

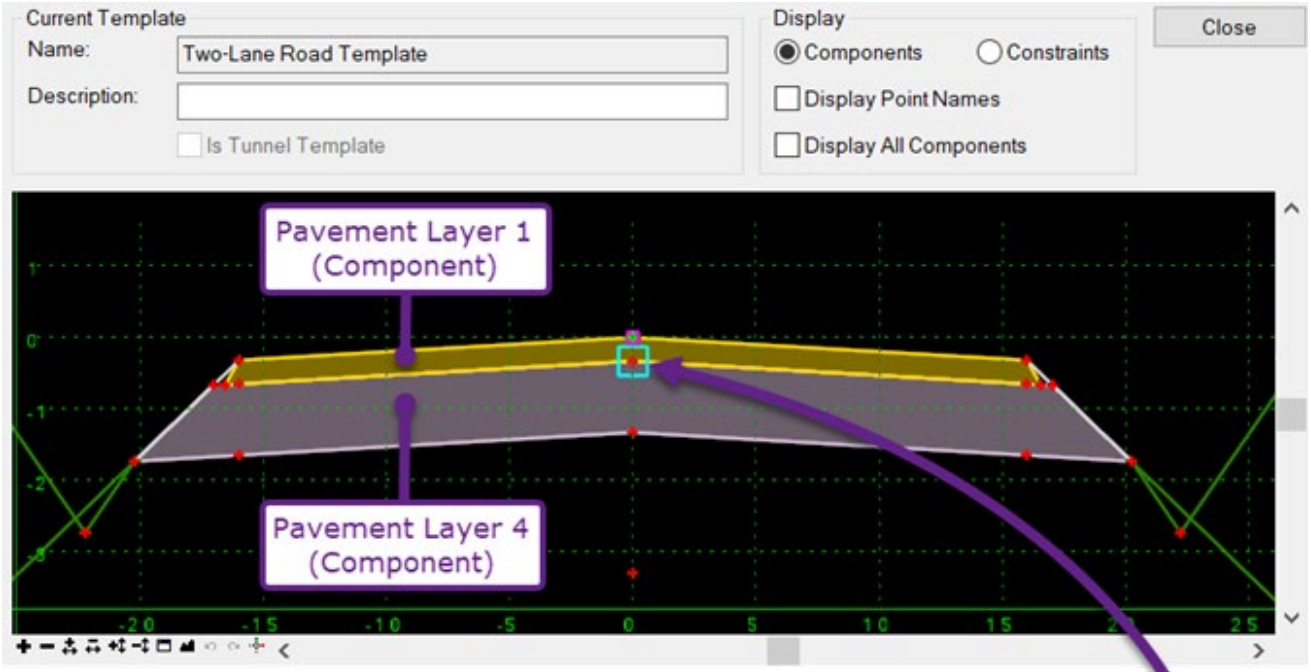
The process for establishing Template Points to be included Alternate Surface is shown in the graphic below. For an example workflow on creating an Alternate Surface, see *22C – Alternate Surfaces (Intermediate Layers)*.

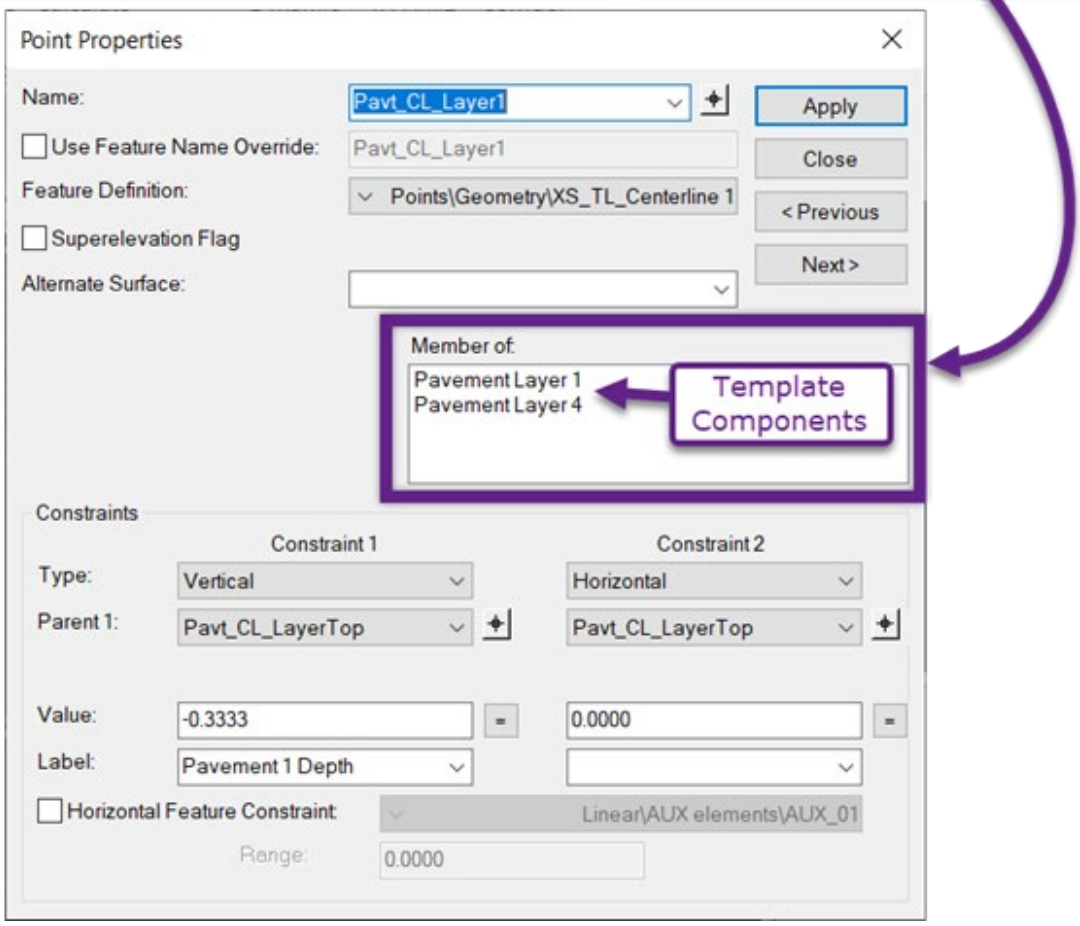
NOTE: The Alternate Surface Terrain Model will not be created until the *Create Corridor Alternate Surfaces*  tool is used.



8C.5 Member of (Point Property)

The **Member Of** box displays which Template Component(s) that a Template Point belongs to. A Template Point can be "Member of" multiple Components if the Components share a common edge. The "Member of" box is for information only and cannot be altered directly in the *Point Properties* menu. To insert or delete a Template Point from a Component, see *8E.1 Insert a Point into a Template Component* and *8E.3 Delete Template Point*.





8C.6 Constraints

Constraints set the position of a Template Point. A Template Point needs TWO defined Constraints to set the position. When a Template Point is **Fully Constrained**, the Point will be shown as a Red Cross in the Template Editor.

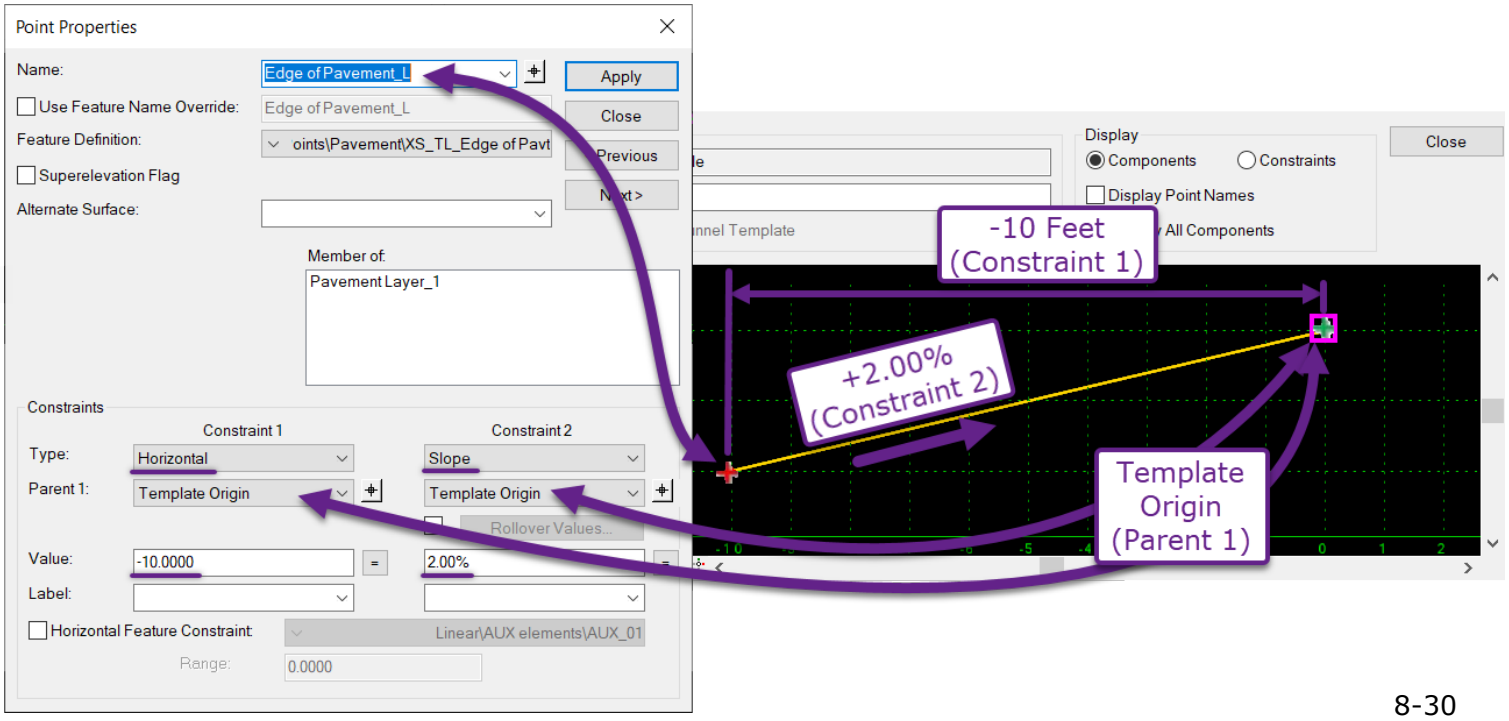
Template Grid: The Template Grid operates as cartesian coordinate system. The Template Origin sets the (0,0) point of the Template grid. Coordinates located to the left and below the Template Origin are considered negative.

Template Point Constraints		
Constraint Identifier:		**Description:**
Fully Constrained		• Two Constraints defined. Position in the Template Grid is fixed. • All Templates Points should be Fully Constrained.
Partially Constrained		• One Constraint defined. Position in the Template grid is partially fixed. • The Point is not geometrically defined in the Template grid. • Resolve all Partially Constrained Points to be Fully Constrained - before use in Corridor Modeling.
Unconstrained		• No Constraints defined. • Resolve all Unconstrained Points to be Fully Constrained, before use in Corridor modeling. • **EXCEPTION:** The Template Origin Point should be Unconstrained.

Constraint Types and Parents
A Template Point is positioned relative to one or more *Parent* Points. The position of a Point is determined by the selected *Parent* point and *Constraint Type*.

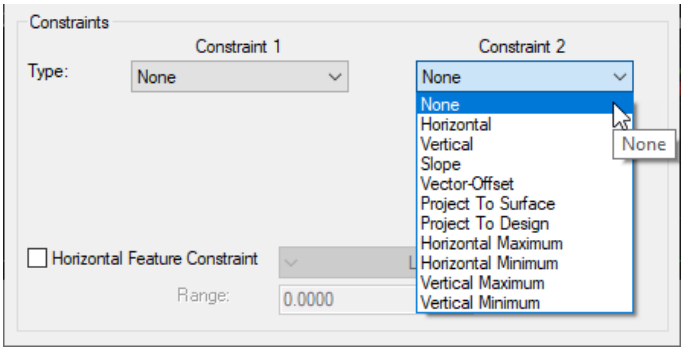
In the simple example shown below, the **Edge of Pavement_L** point is defined by the **Horizontal** and **Slope** constraint types, which are relative to the **Template Origin** point (parent).

Constraint 1: the point is set **Horizontal** (*Type*) to the **Template Origin** (*Parent 1*) at **-10 feet** (*Value*).
Constraint 2: the point is set at **Slope** (*Type*) to the **Template Origin** (*Parent 1*) at **+2.00%** (*Value*).

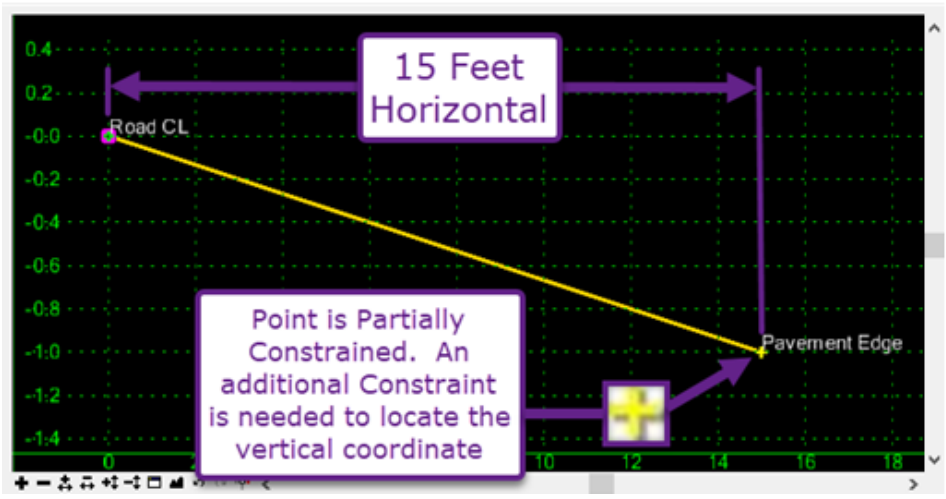


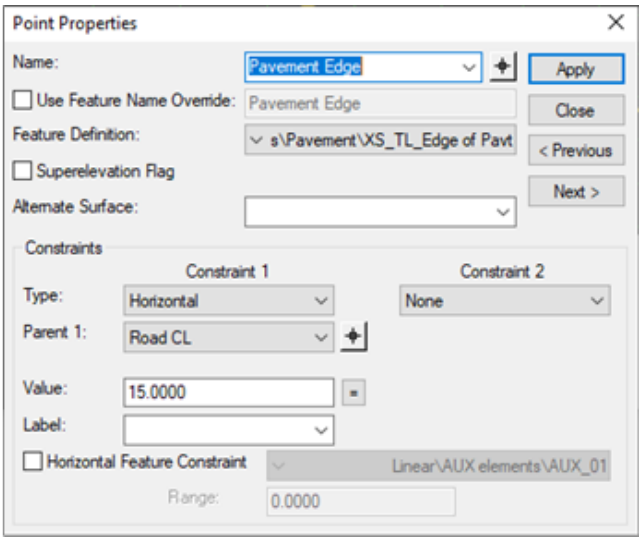
8C.6.a Constraint Types

8C.6.a.i None - No Constraint is defined. Template Points containing this Constraint Type will be Partially Constrained or Unconstrained .

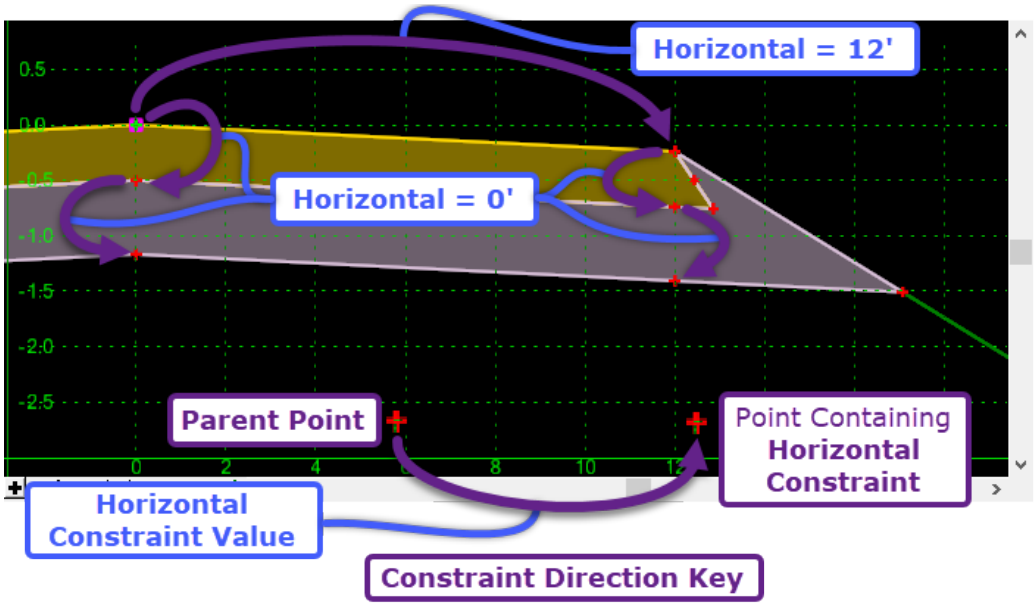


8C.6.a.ii Horizontal - The Template Point is fixed at a specified *Horizontal* distance from the Parent Point. Requires a Parent Point and horizontal distance Value.

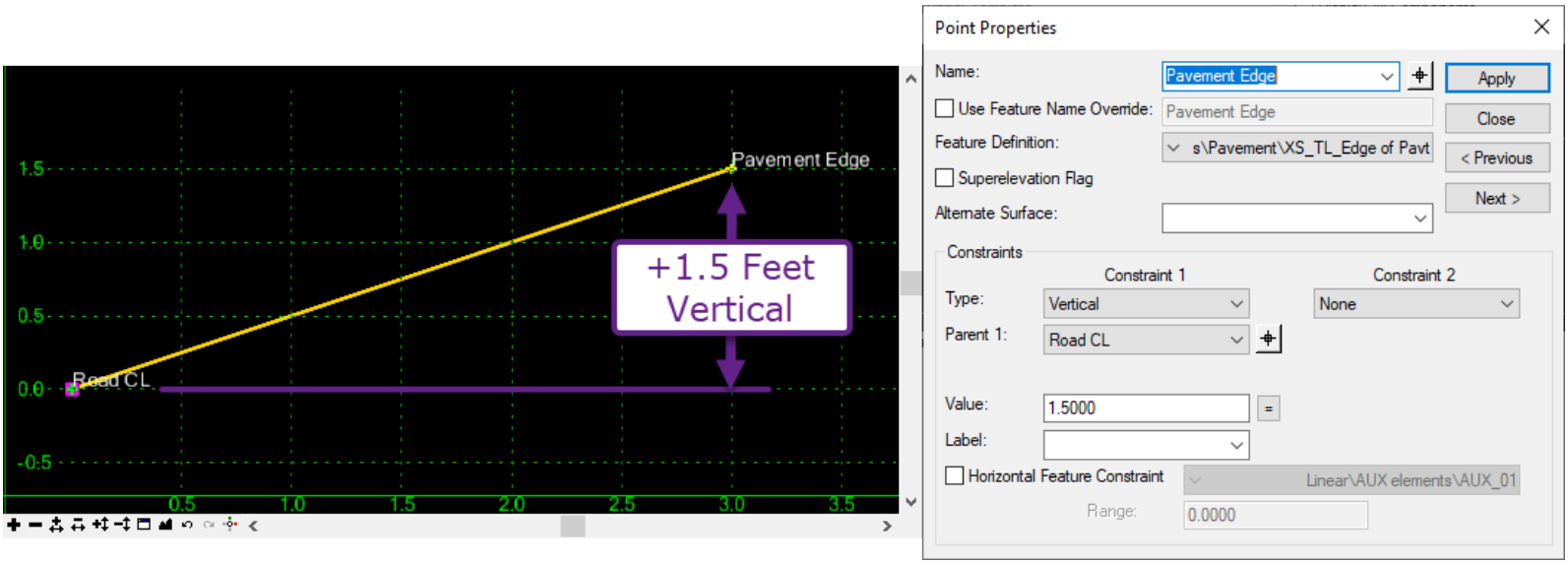




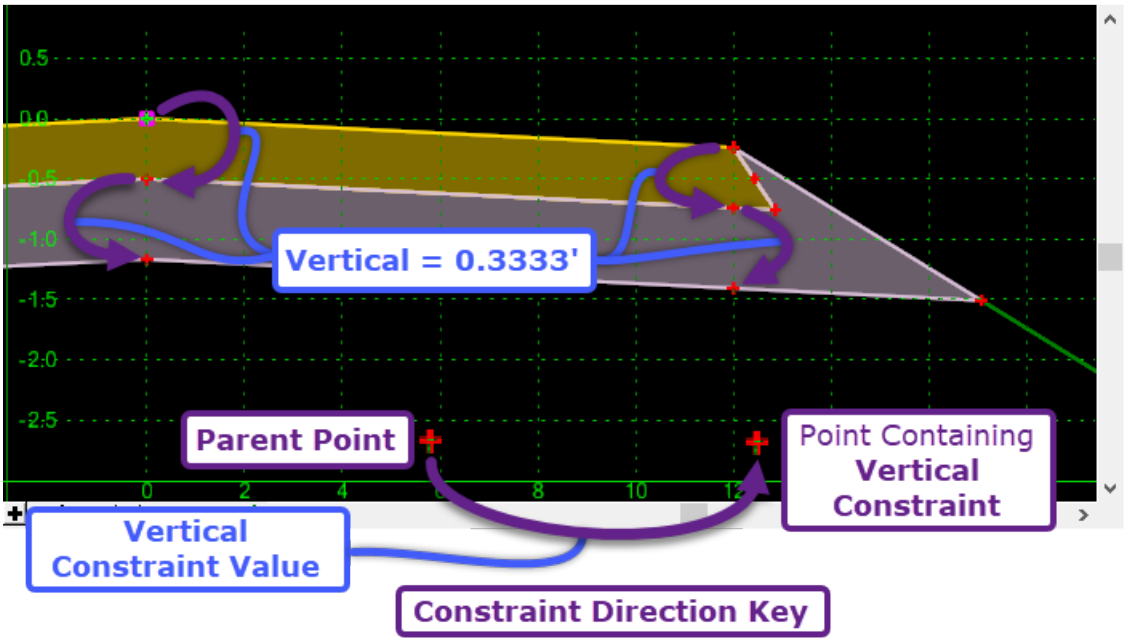
The *Horizontal* Constraint type is used when the Point position is fixed in a horizontal position relative to a Parent Point. Example use of the *Horizontal* Constraint use in a Road Template is shown below:



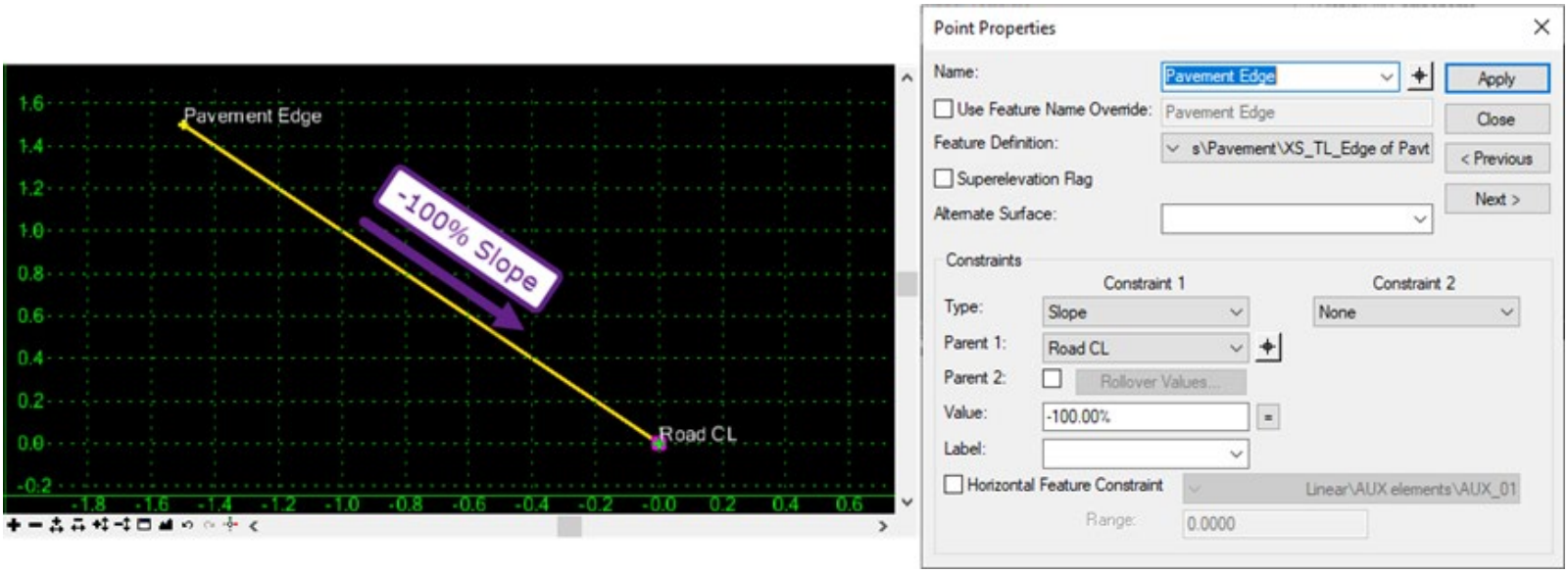
8C.6.a.iii Vertical - The Template Point is fixed at a specified *Vertical* distance from the Parent Point. Requires one Parent Point and a vertical distance Value.



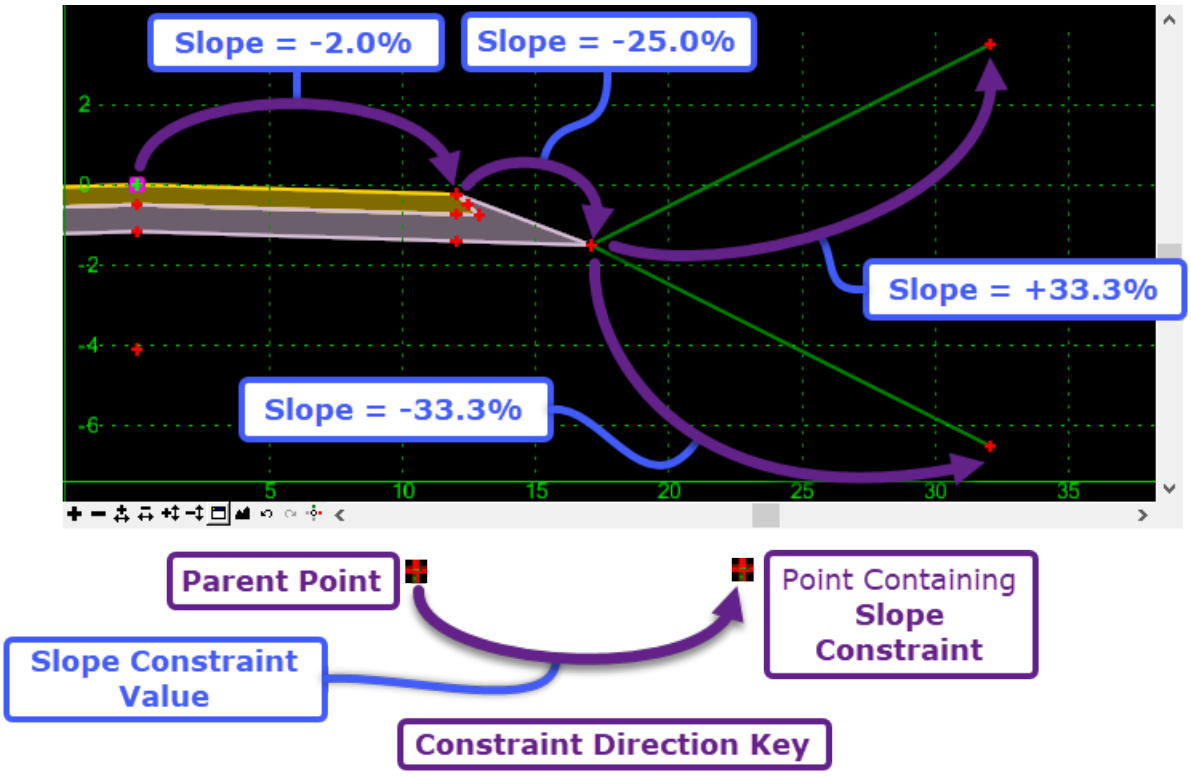
The *Vertical* Constraint type is used when the Point position is fixed in a vertical position relative to a Parent Point. Example use of the *Vertical* Constraint use in a Road Template is shown below:



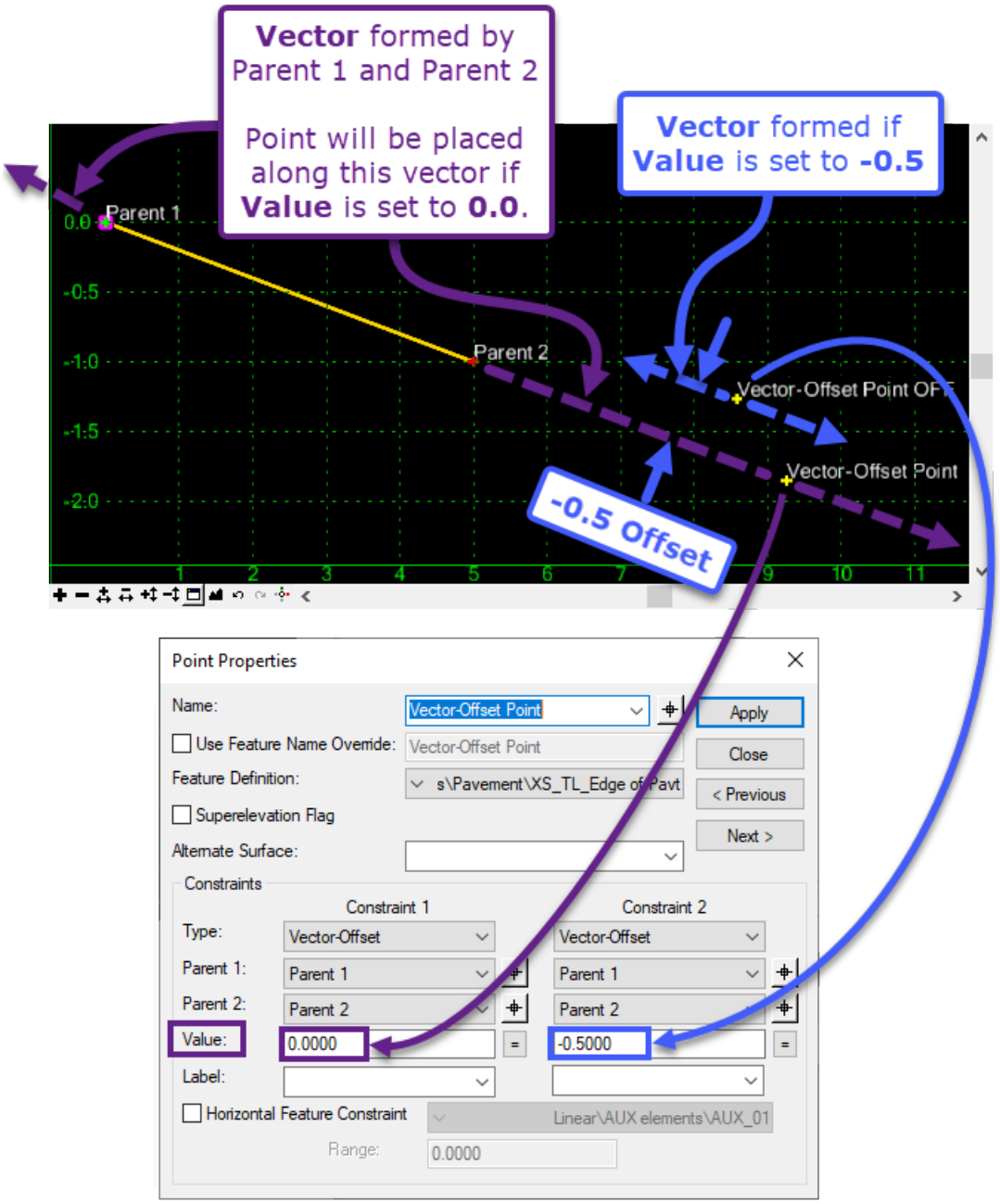
8C.6.a.iv Slope - The Template Point is fixed along a specified *Slope* vector from the Parent Point. Requires one Parent Point and a Slope Value in percentage form.



Example use of the *Slope* Constraint use in a Road Template is shown below:

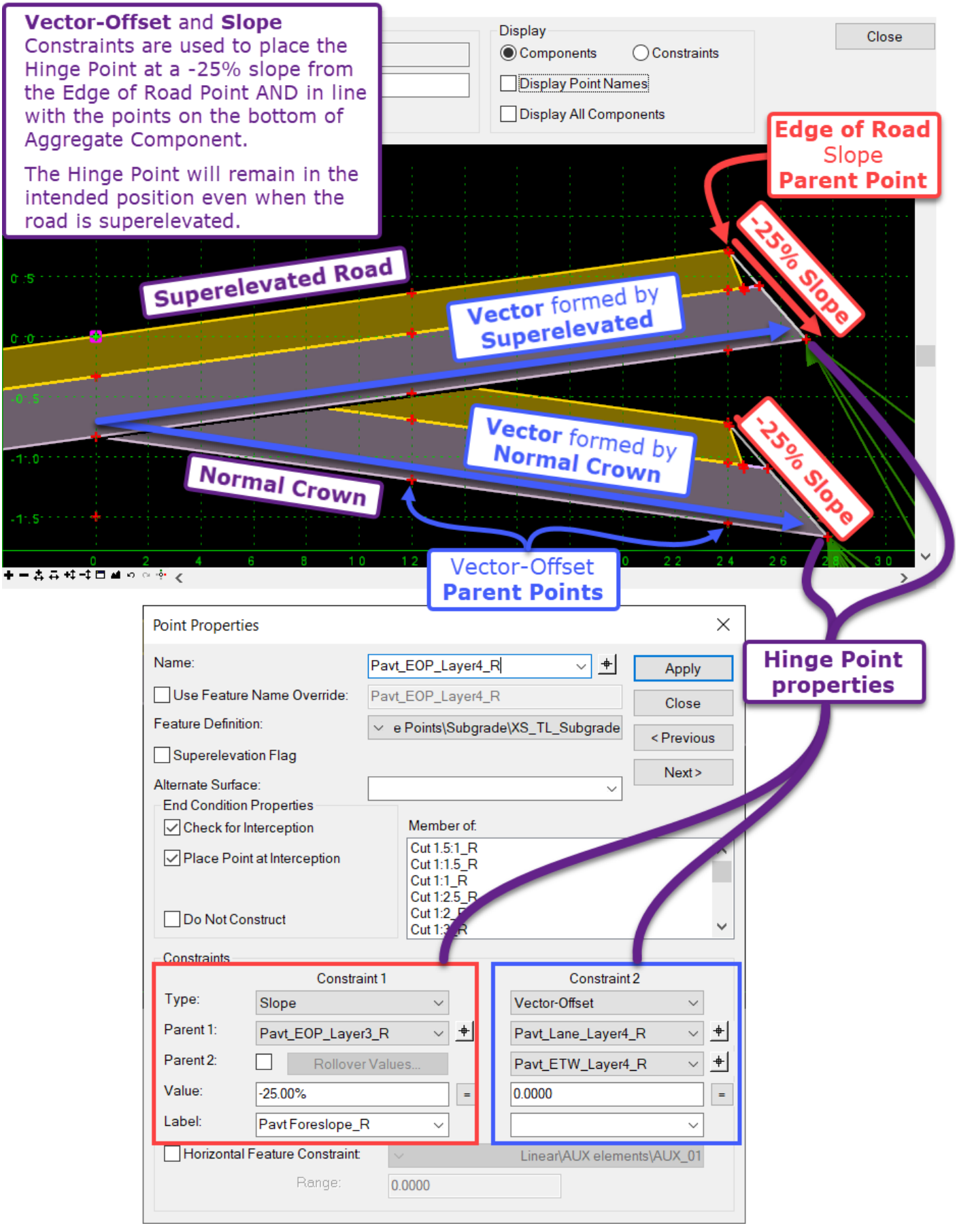


8C.6.a.v Vector Offset - The Vector-Offset constraint is very common in FLH Templates and is one of the few constraint types that use TWO Parent Points. In its simplest and most common use, the Vector-Offset is used to place a Point along a vector that is defined by two Parent Points (Offset Value = 0). If the Offset is a non-zero value, then the Point will be placed on a parallel, offset vector. Negative Offset values place the Point to the *left* of the parent vector. Positive Offset values place the Point to the right.



See the next page for an example use of the Vector-Offset Constraint Type.

The Vector-Offset constraint is commonly used on the Hinge Point of a road template. The Vector-Offset constraint allows the Hinge Point to remain in the intended position, even when a road is superelevated.



8C.6.a.vi Project To Surface - The Template Point is automatically placed on the surface of a Terrain Model.

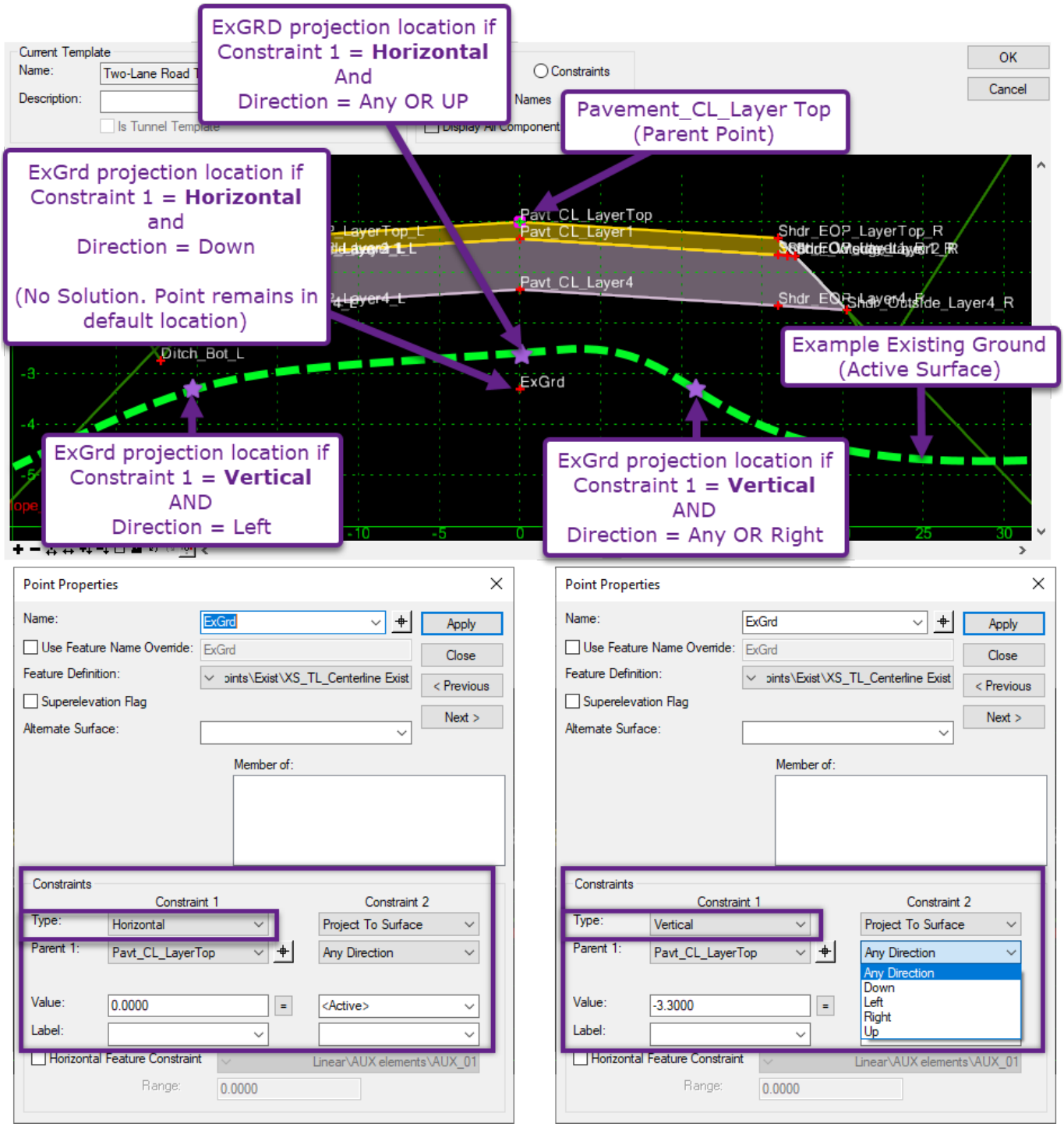
WARNING: Do NOT use this Constraint Type with End Condition Points. Project to Surface functionality is inherently built into End Condition Points when the *Place Point at Interception* box is checked. See *8C.7.a End Condition Point Properties*.

In FLH Road Templates, the *ExGrd* Null Point utilizes this Constraint Type. The *ExGrd* is used to show the existing ground elevation in Cross Section Production. See *16H.4 Existing Ground Elevation Label ("Original Grnd") is Absent*.

This Constraint Type could also be used in creation of advanced templates to model retaining walls or sub-excavation.

This Constraint Type does NOT require Parent Points but requires a precursor Constraint to determine the direction which the Template Point should be projected in order to intercept the Terrain Model surface.

See the next page for an example use of the Project To Surface Constraint Type.



In the graphic above, the ExGrd point has a **Horizontal** Constraint 1 with a value of 0.0000 in relation to the Pavement_CL_LayerTop (parent point). This means the ExGrd point is ONLY free to move in the **vertical** direction when seeking the surface. ***NOTE:*** Left and Right options for the Project to Surface constraint have no effect when the Horizontal position of the point is constrained.

When the Constraint 1 is changed to **Vertical** with a value of -3.300 in relation the parent point, then the vertical datum for the ExGrd is locked and the point can only move in the **horizontal** direction when seeking the surface.

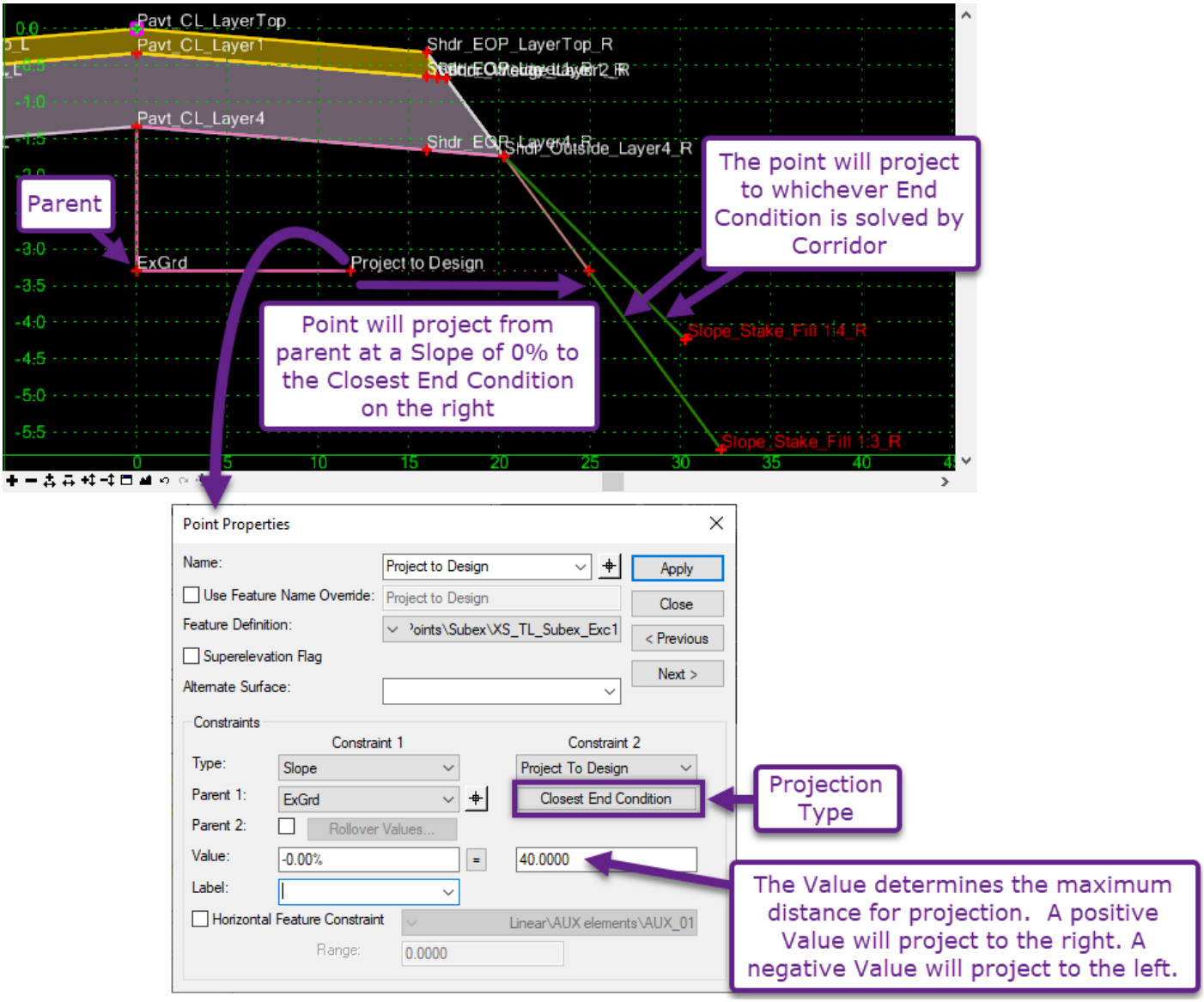
NOTE: If the Point CANNOT find a solution, given the specified terrain model and direction, the Point will remain in its default position in the Template.

8C.6.a.vii Project To Design - This Constraint Type is similar to *Project To Surface*, except the Template Point is automatically projected to the surface of a Template Component. This Constraint type could be used in advanced template creation to model roadway sub-excavation.

Similar to the *Project To Surface* constraint type, a precursor Constraint is required to determine the projection direction.

NOTE: If the Point can NOT find a solution, given the specified terrain model and direction, the Point will remain in its default position in the Template.

Project to Design – Closest End Condition

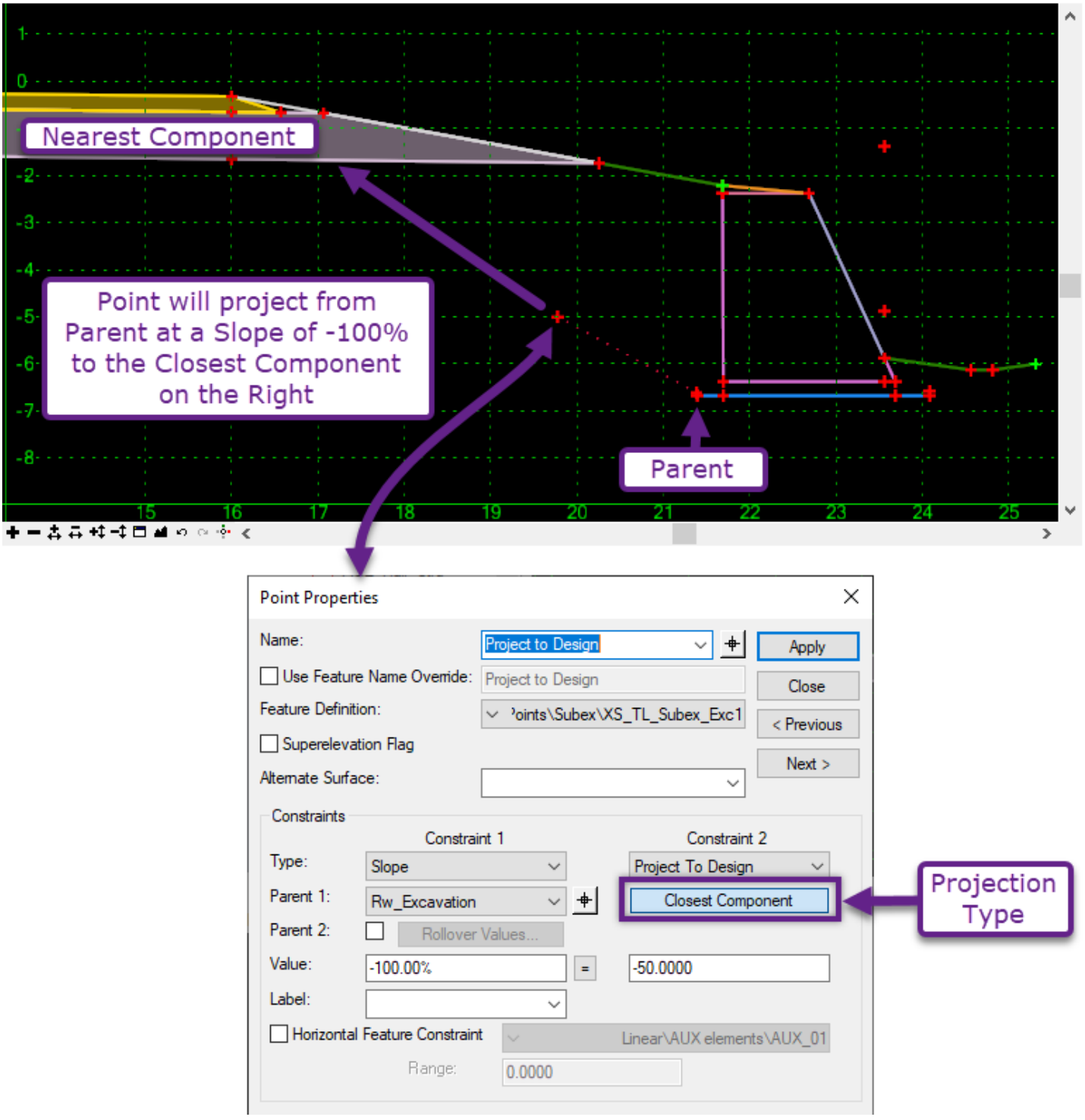


In the example shown above, the *Project to Design* point has a precursor Constraint of 0.00% from the Parent Point. This means the Project to Design Point can be placed anywhere on the 0.00% slope vector from beginning at the Parent Point. Constraint 2 specifies that the Project Type is to the *Closest End Condition*.

Also, the Value for Constraint 2 is set at +40.0000. This will allow the point to search for an End Condition Component in the positive direction (to the right) for a maximum distance of 40 feet. The Value provided under Constraint 2 also specifies the maximum distance the point will search for the End Condition.

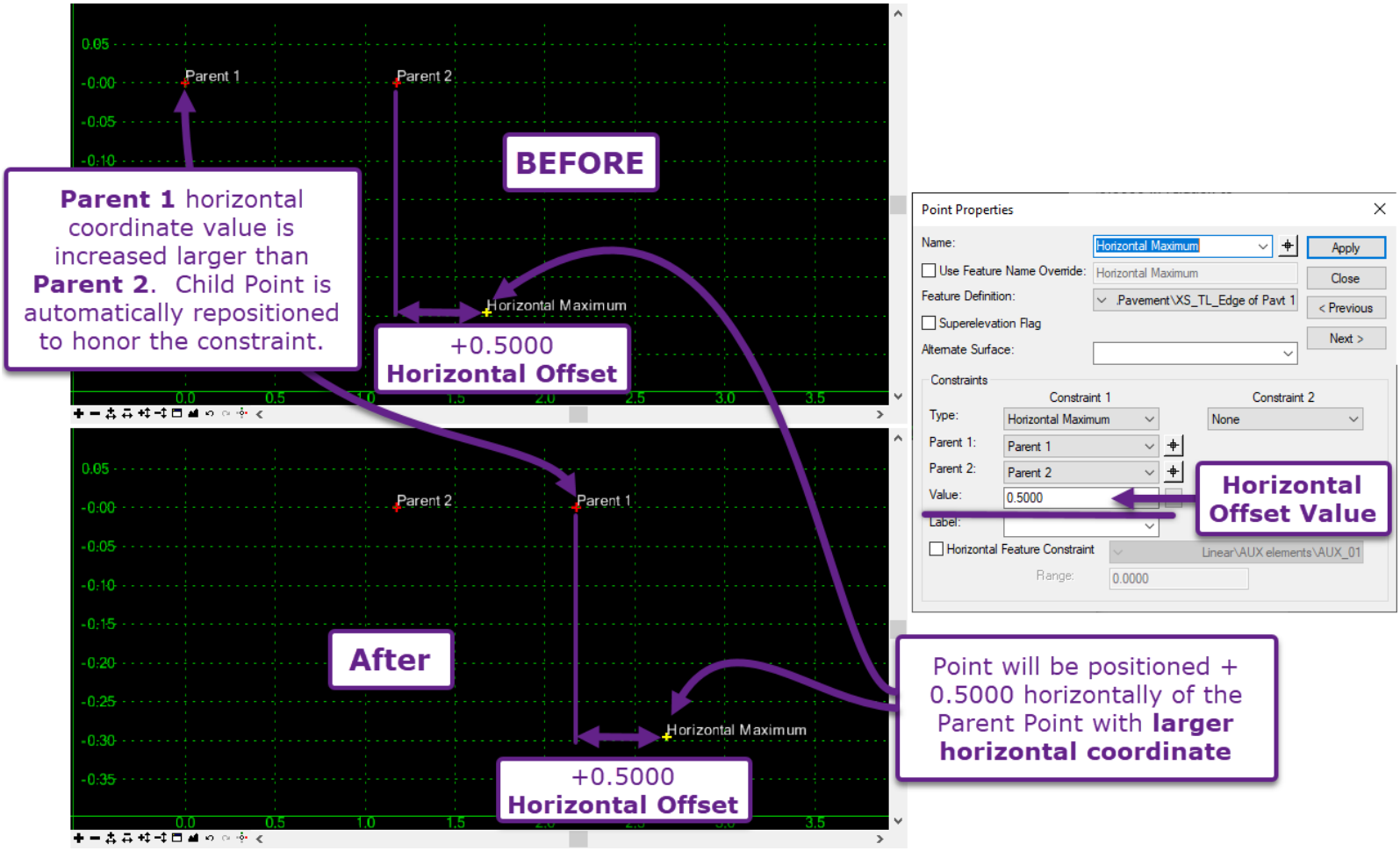
Project to Design – Closest Component

The example below shows how *Project To Design* constraint with the *Closest Component* Projection Type can be used to model excavation behind a retaining wall.



8C.6.a.viii Horizontal Maximum - This is an advanced Constraint Type, requiring TWO Parent Points. The Template Point will be horizontally offset relative to one of the two Parent Points. The Parent Point that is used depends on which Parent has the larger horizontal coordinate value (further to the right in the grid).

This Constraint is useful in building conditional templates – if one of the two Parent Points is subjected to Point Control, Parametric Control, Horizontal Feature Constraint, or any other means of external template point manipulation.



8C.6.a.ix Horizontal Minimum - Operates the same as Horizontal Maximum, except the Parent Point with the smaller Horizontal Coordinate value (further to the left in the grid) will define the offset origin of the Template Point.

8C.6.a.x Vertical Maximum - Operates under the same principals as Horizontal Maximum. The Template Point will be offset vertically from one of two Parent Points – whichever Parent having the larger Vertical coordinate Value (further up in the grid).

8C.6.a.xi Vertical Minimum - Operates under the same principles as Horizontal Minimum. The Template Point will be offset vertically from one of two Parent Points – whichever Parent has the smaller Vertical coordinate Value (further down in the grid).

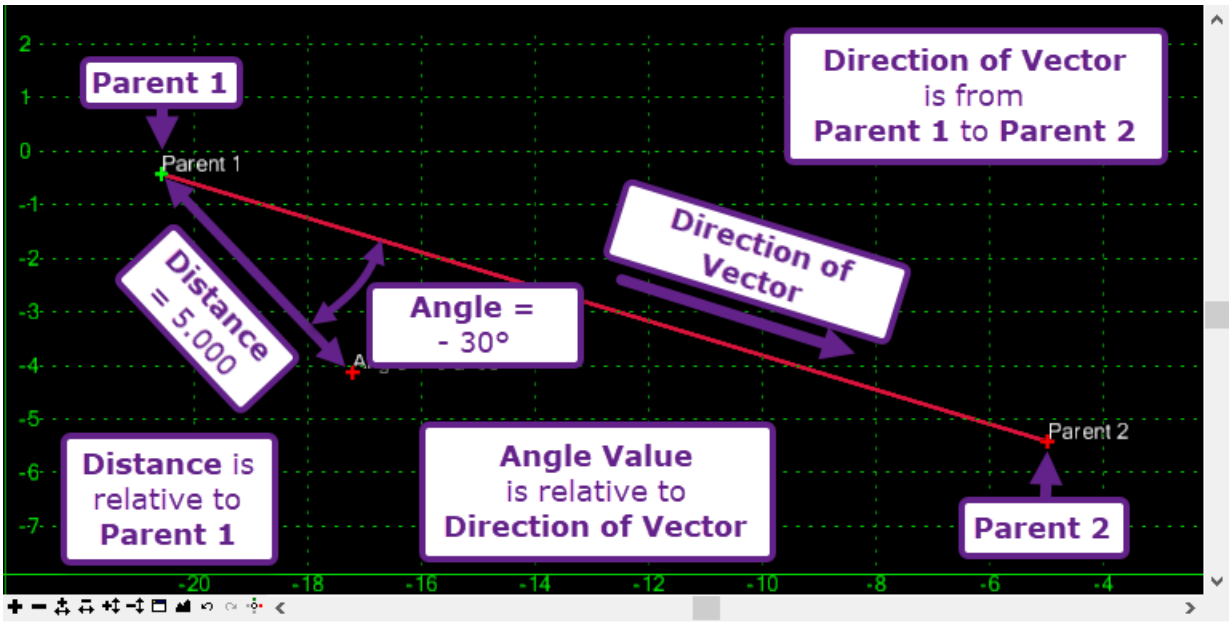
8C.6.a.xii Angle Distance - The *Angle Distance* Constraint Type is unique because it does NOT need to be paired with a Constraint 2. By itself, the *Angle Distance* Constraint Type provides enough geometrical information to Fully Constrain a Point.

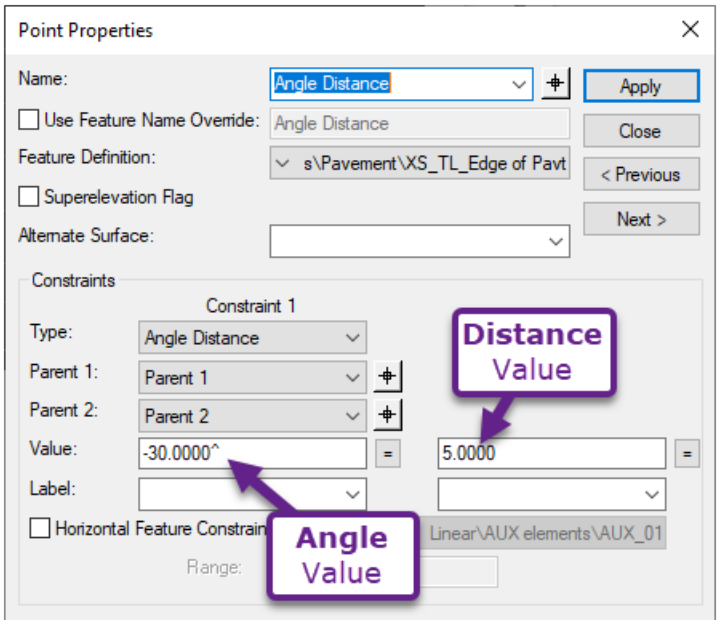
The *Angle Distance* Constraint is commonly used to create a properly configured Pavement Safety Edge. With the *Angle Distance* Constraint, the Pavement Safety Edge slope is maintained, even when the road is superelevated See *8F.2 Simple Road Template for Superelevation and Safety Edge*.

This *Angle* Constraint Type requires TWO Parent Points, an Angle Value, and Distance Value.

A directional vector is determined by Parent 1 and Parent 2. The vector direction is from Parent 1 to Parent 2. The Angle Value is relative to the Vector. A POSITIVE Angle Value will place the Template Point COUNTER-CLOCKWISE to the vector. A NEGATIVE Angle Value will place the Point CLOCKWISE to the vector.

The Distance Value is measured from Parent 1 location and is projected from the Angle.



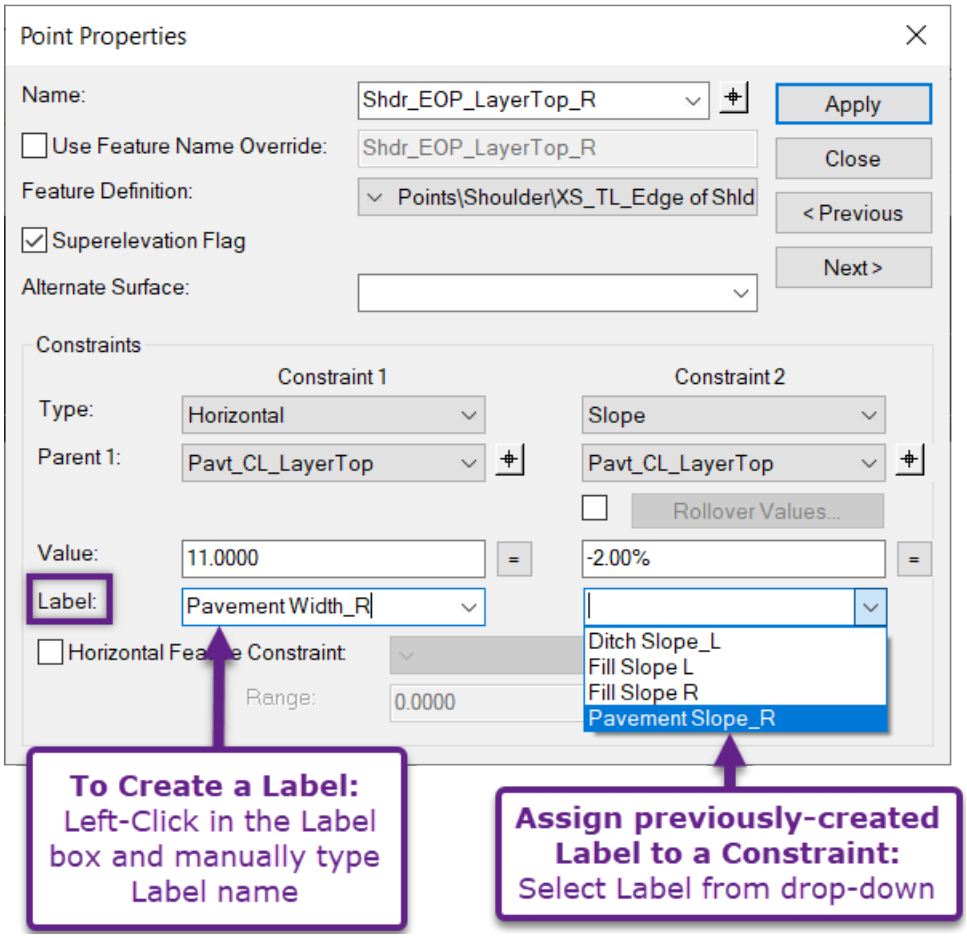


8C.6.a.xiii Label -The *Label* box works in construction with the **Parametric Constraints** tool. See *9G.4 Parametric Constraints*. When a Label is defined for a Constraint, the corresponding value of the Constraint can be modified over a certain station range with the *Parametric Constraints* tool.

To Create a Label – simply Left-Click in the Label box and type in the desired Label name.

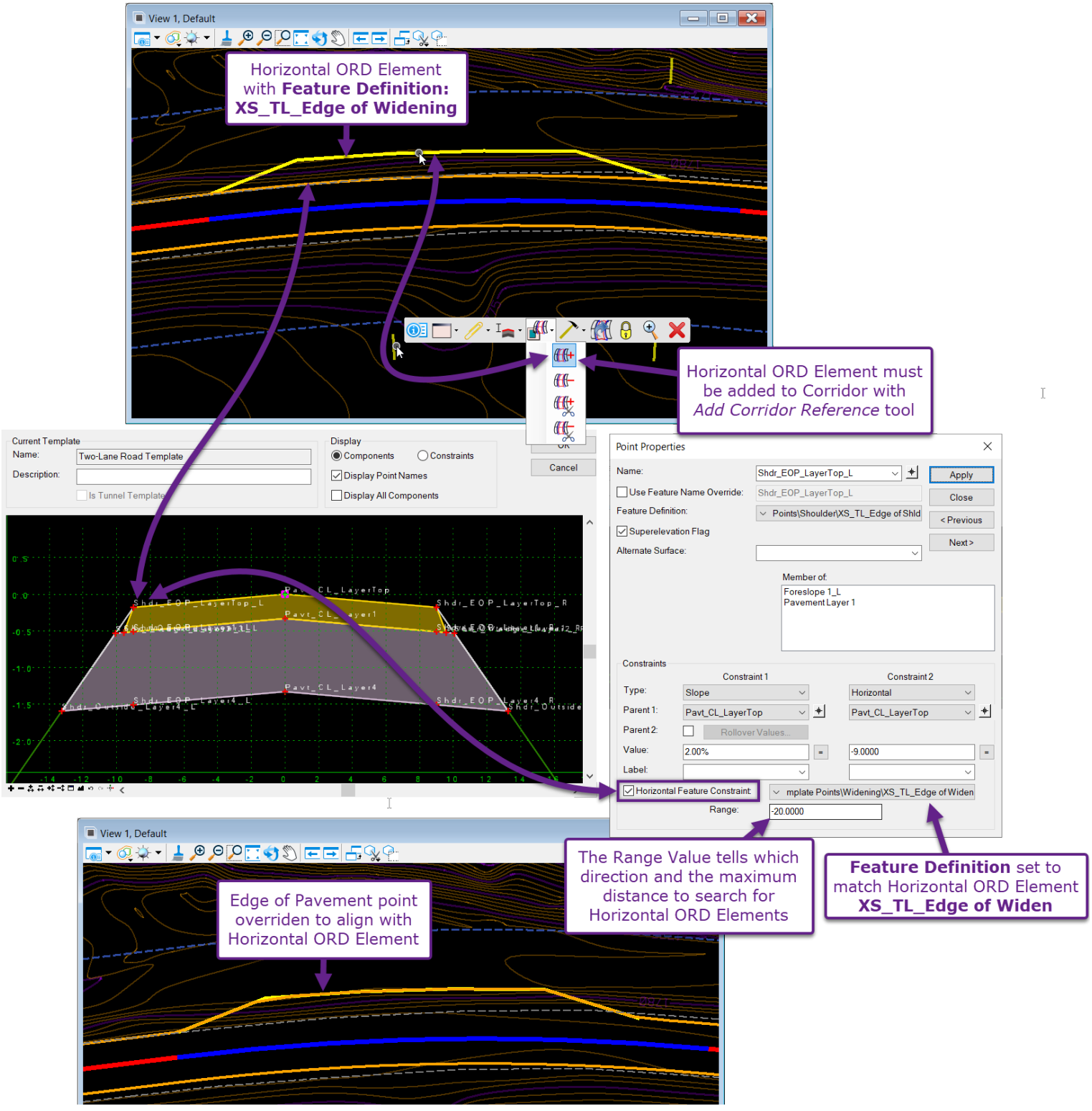
Once a Label has been created, it will be shown in the Label drop-down list. Previously-created Labels can be assigned to any Constraint within the Template.

WARNING: Labels should NOT be mixed between different Constraint Types. For example, do NOT assign a Horizontal Label to a Slope Constraint.



8C.6.a.xiv Horizontal Feature Constraint - This Point Property operates similar to the *Point Control* tool. If this box is CHECKED, the horizontal position of a Template Point is overridden to align with an ORD Element in the *2D Design Model* . The Template Point will seek out ORD Elements that match the set **Feature Definition**. If an ORD Element (with the corresponding Feature Definition) is found, then the Template Point is moved to match the horizontal position of the ORD Element. If an ORD Element is NOT found, then the Template Point remains in its default position.

IMPORTANT: The ORD Elements must be added as a Corridor Reference with the *Add Corridor Reference* tool for the *Horizontal Feature Constraint* to function. See 9G.9 Corridor References.



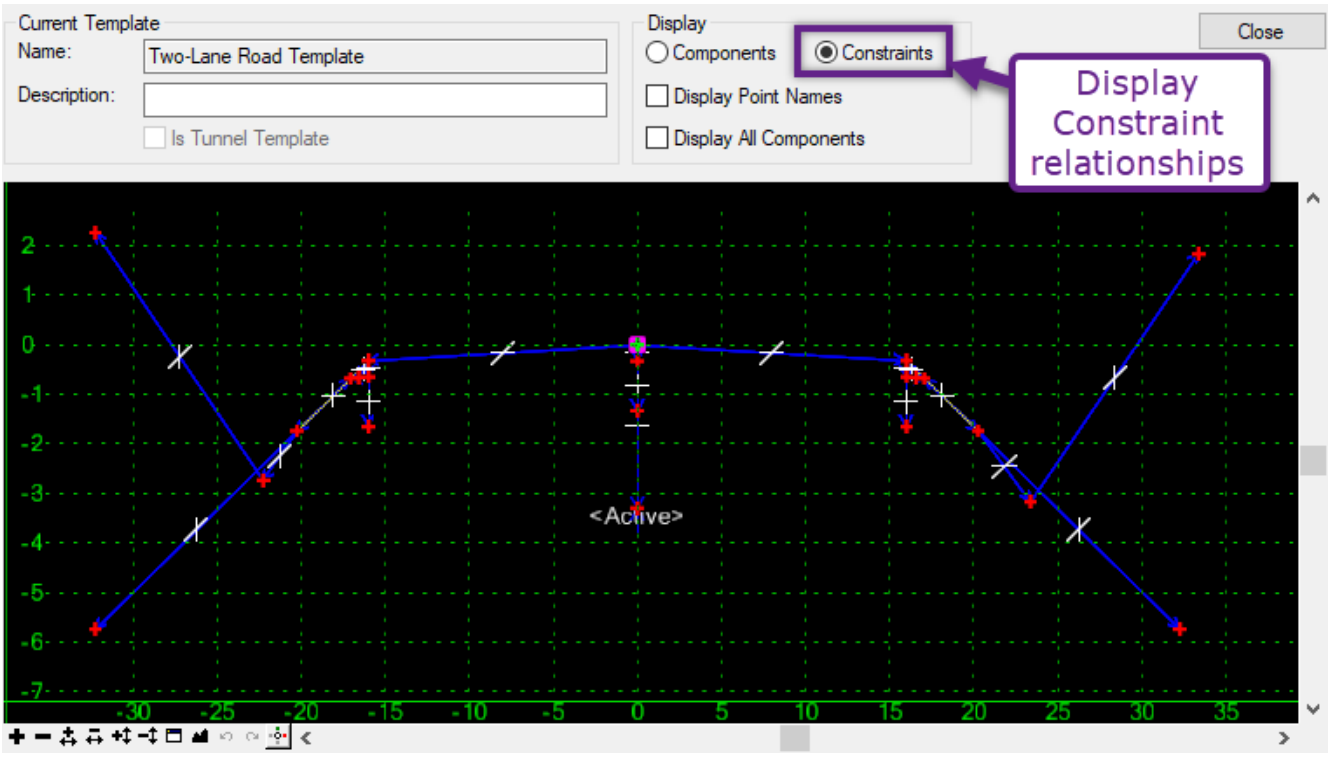
Range: The Range value sets the direction and maximum horizontal distance the Template Point will seek the Horizontal ORD Element. Positive values will seek to the right of the Template Point. Negative values will search to the left. If the Range is set to 0, the Template Point will search infinitely in both directions.

Horizontal Feature Constraints are primarily used with Display Rules to trigger a Null Point. When a Null Point is triggered, Components in the Template will appear or disappear. For more information on Horizontal Feature Constraints use with Display Rules, see *8D.2 Display Rules*.

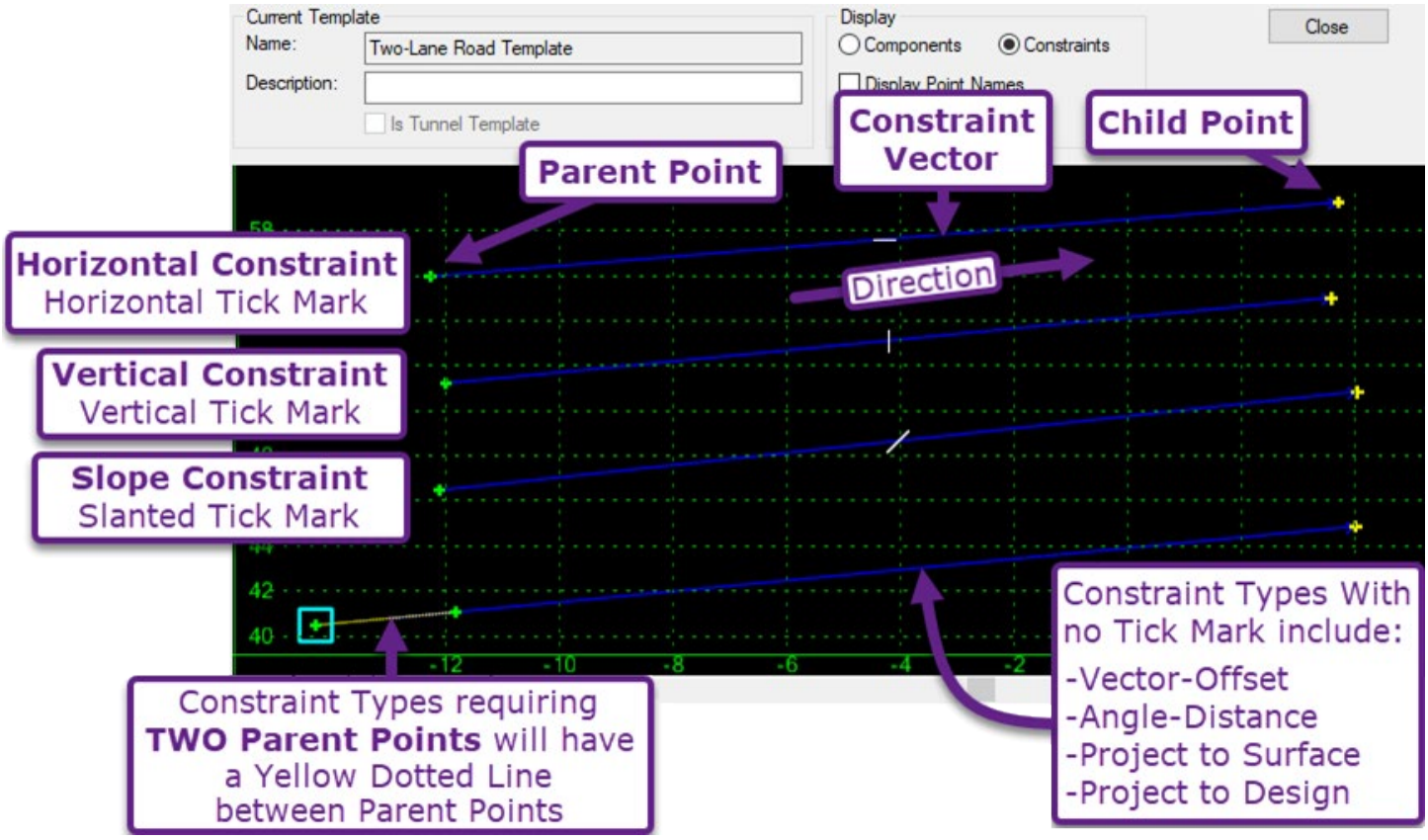
Horizontal Feature Constraints are demonstrated in *8F.3 Advanced Road Template with Guardrail and Display Rules* and *8F.4 Mainline Road Template with Display Rules for Managing Approach Roads and Driveways*.

8C.6.b Constraint Display in Active Template Editor Screen

As a Templates increases in complexity, it its beneficial to *visualize* Constraint relationships contained in the Template. The Active Template Editor Screen can be changed to view the Constraints relationships in the Template.



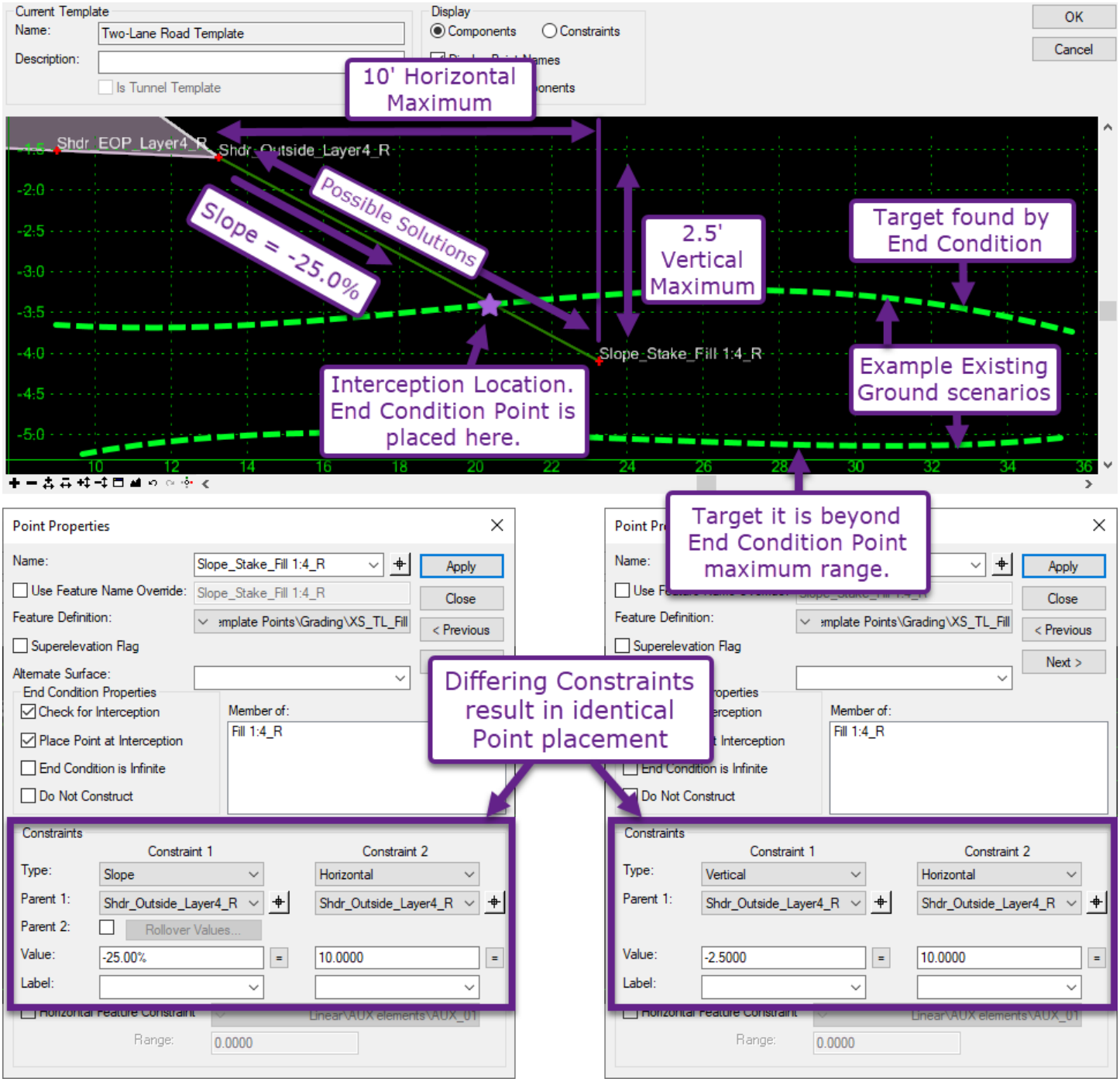
Constraints appear as Blue Vectors drawn from the Parent Point to the Child Point.
White ticks appear in the middle of the Constraint vector to represent the type of constraint.



8C.7 End Condition Template Points

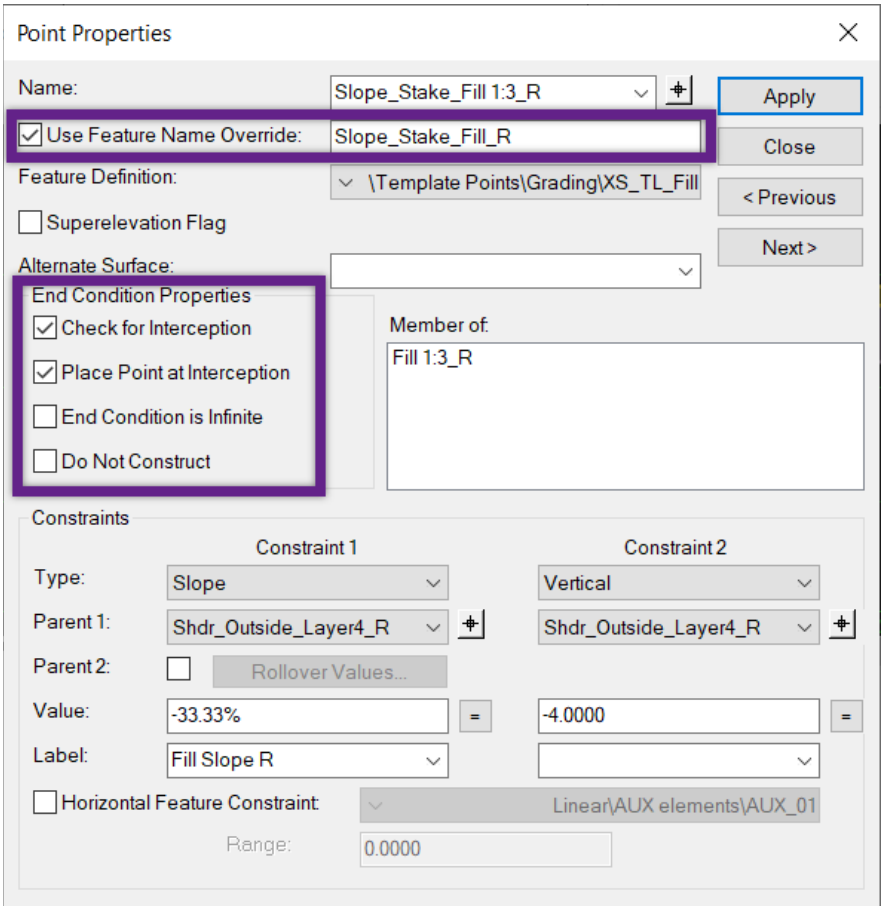
The basic operation of End Condition Points is shown below. End Condition Points are projected along a vector to intercept the set "Target". The Target can be a Terrain Model, single elevation, or a linear element. By default, the Target is set active Terrain Model. The Target can be modified in the Component Properties menu when the End Condition is selected.

If the Target is beyond the constraint range of the End Condition Point, then neither the End Condition Component nor the Point is constructed. The exception to this rule is if the **End Condition is Infinite** box is CHECKED. When CHECKED, the End Condition Point will project as far as necessary to intercept the Target.



8C.7.a End Condition Point Properties

End Condition Points contain a unique set of properties that are NOT found on conventional Template Points. Specifically, the **Check for Interception**, **Place Point at Interception**, **End Condition is Infinite**, and **Do NOT Construct** properties.



Check for Interception: If this box is CHECKED, then the End Condition Point will search for a Target (i.e., the Existing Ground Terrain Model. specified in the End Condition Component Properties.

If this box is UNCHECKED, the End Condition Point behaves like a Conventional Template Point. If UNCHECKED, the End Condition Point will NOT seek out a Target.

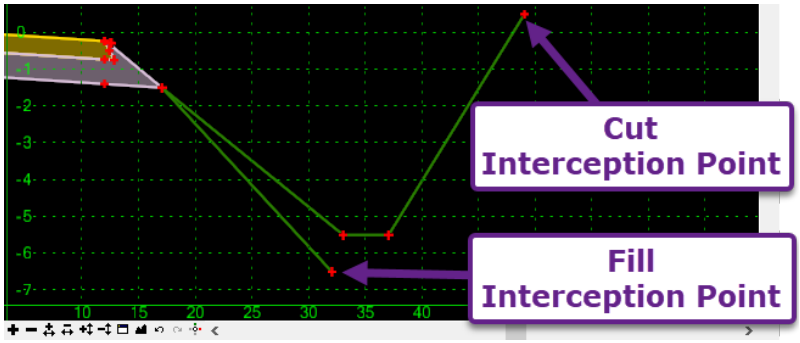
Place Point at Interception: If this box is CHECKED, then the End Condition Point is placed at the interception point between the End Condition vector and the Target (i.e., the Existing Ground Terrain Model).

If this box is UNCHECKED, then the line segment is created at full length. The End Conditions Point is NOT placed to intercept the target.

End Condition is Infinite: If this box is CHECKED, then the End Condition Point will automatically extend to intercept the Target, even if the Target is located beyond the constraint range.

Do Not Construct: If this box is CHECKED, then the End Condition Point is not used (constructed). A line segment is drawn between the previous End Condition Point and the subsequent Point. End Condition Points that use this property are often used as a reference point in advanced template creation. As an example, *see 8C.7.g Place End Condition Point Past First Interception (Do Not Construct - Demonstration)*.

Use Feature Name Override – Feature Name Overrides should ONLY be applied to the Cut and Fill Interception Points. The Feature Name Override ensures that End Condition Points of different slopes are properly joined together when the Corridor is created. For more information on the effects of Feature Name Overrides, see *9C.4.e.ii Effect of Template Point Name Overrides on Corridor Complex Elements*.



8C.7.b Feature Name Override Convention for End Condition Points

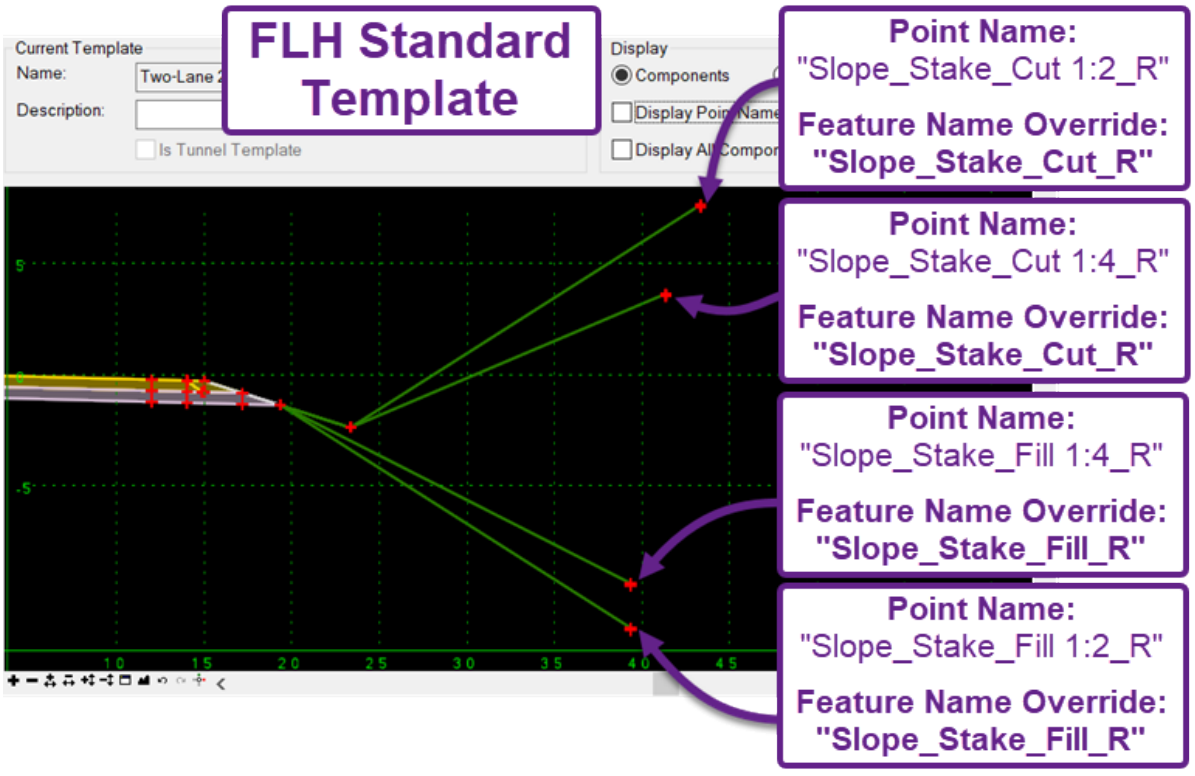
In the FLH Template Library, ALL Cut and Fill End Interception Points are assigned a Feature Name Override.

All **Cut Interception Points** are set to either "Slope_Stake_**Cut**_R" or "Slope_Stake_**Cut**_L", depending on which side of the road alignment the interception point is placed on.

All **Fill Interception Points** are set to "Slope_Stake_**Fill**_R" or "Slope_Stake_**Fill**_L".

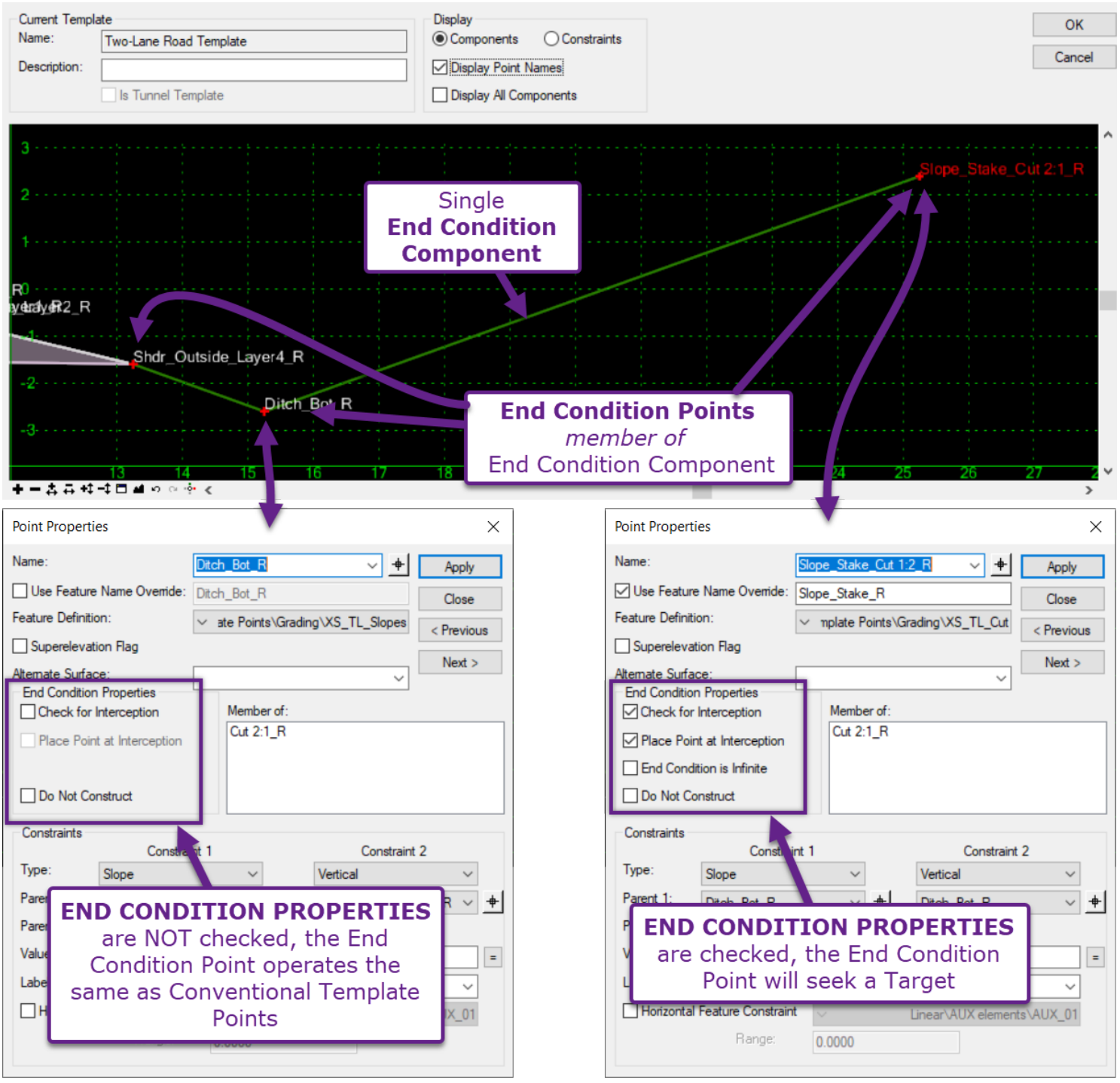
The Feature Name Override ensures that End Condition Points of different slopes are properly joined together when the Corridor is created. For more information on the effects of Feature Name Overrides, see *9C.4.e.ii Effect of Template Point Name Overrides on Corridor Complex Elements*.

When creating custom End Condition Points, assign a Feature Name Override that is consistent with the convention discussed above.



8C.7.c End Conditions with Multiple Line Segments (Ditches)

It is often necessary to create End Condition component with multiple line segments for ditch modeling. The Ditch Point should behave like a Conventional Template Point. By UNCHECKING the "Check for Interception" box, the position of the Ditch Point becomes fixed.



8C.7.d Multiple End Conditions – Slope Stake Tables

Road Templates in the FLH Template Library typically have multiple End Conditions per side – ranging in steepness from 4V:1H (400%) to 1V:6H (16.6%). The End Condition that is constructed for each Corridor station depends on two criteria:

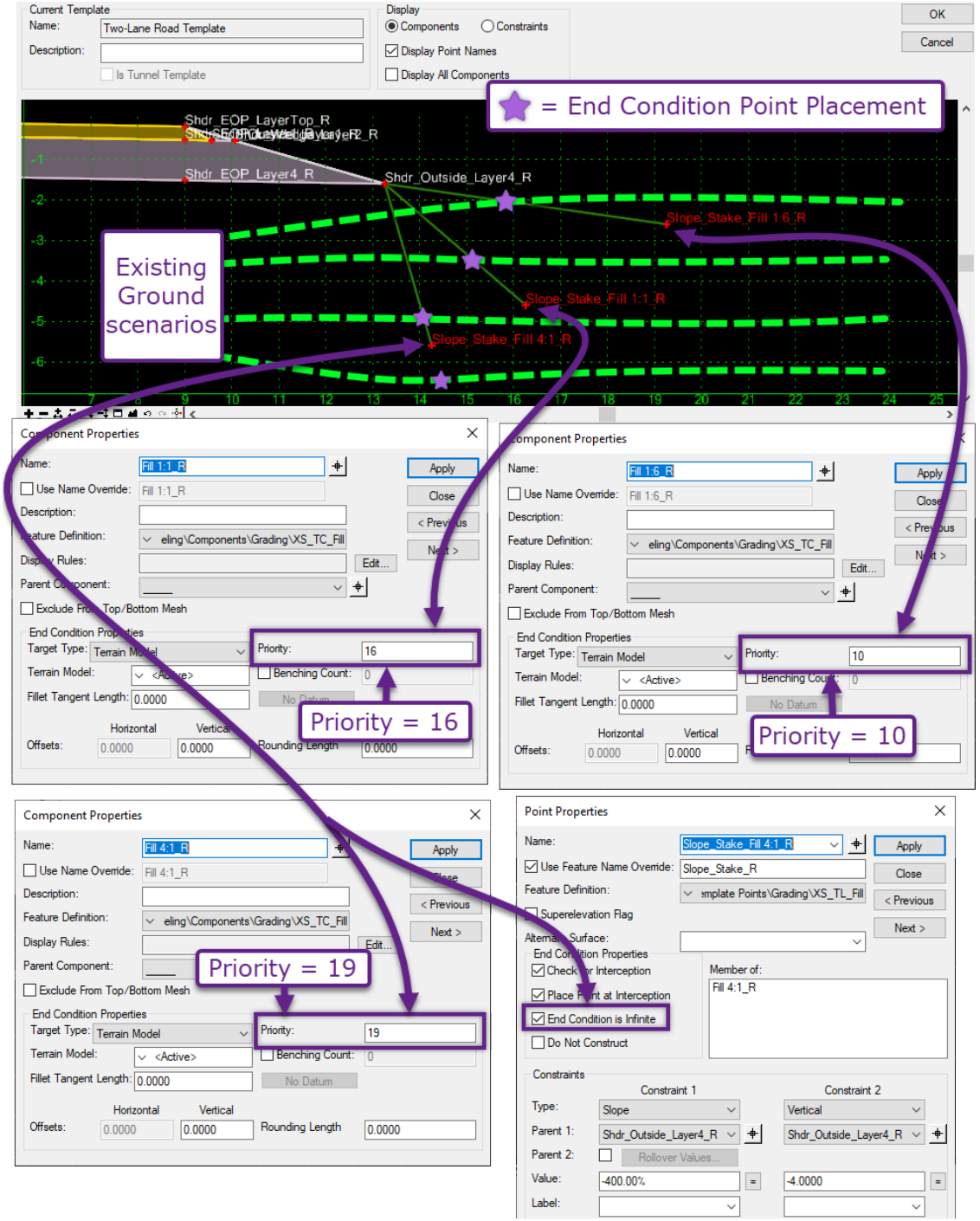
1. The End Condition must be *Solvable* – within the *Constraint* range of the End Condition Point.

2. In the event that several End Conditions are solvable, the End Condition with the lowest-value *Priority* is constructed. *Priority* value is shown in the End Condition Component Properties

The Slope Stake Tables in the FLH Template Library are ranked in *Priority* from flattest to steepest. For example, if the 1V:2H and a 1V:4H Fill Slopes are both solvable, the flatter 1V:4H Fill Slope will be used.

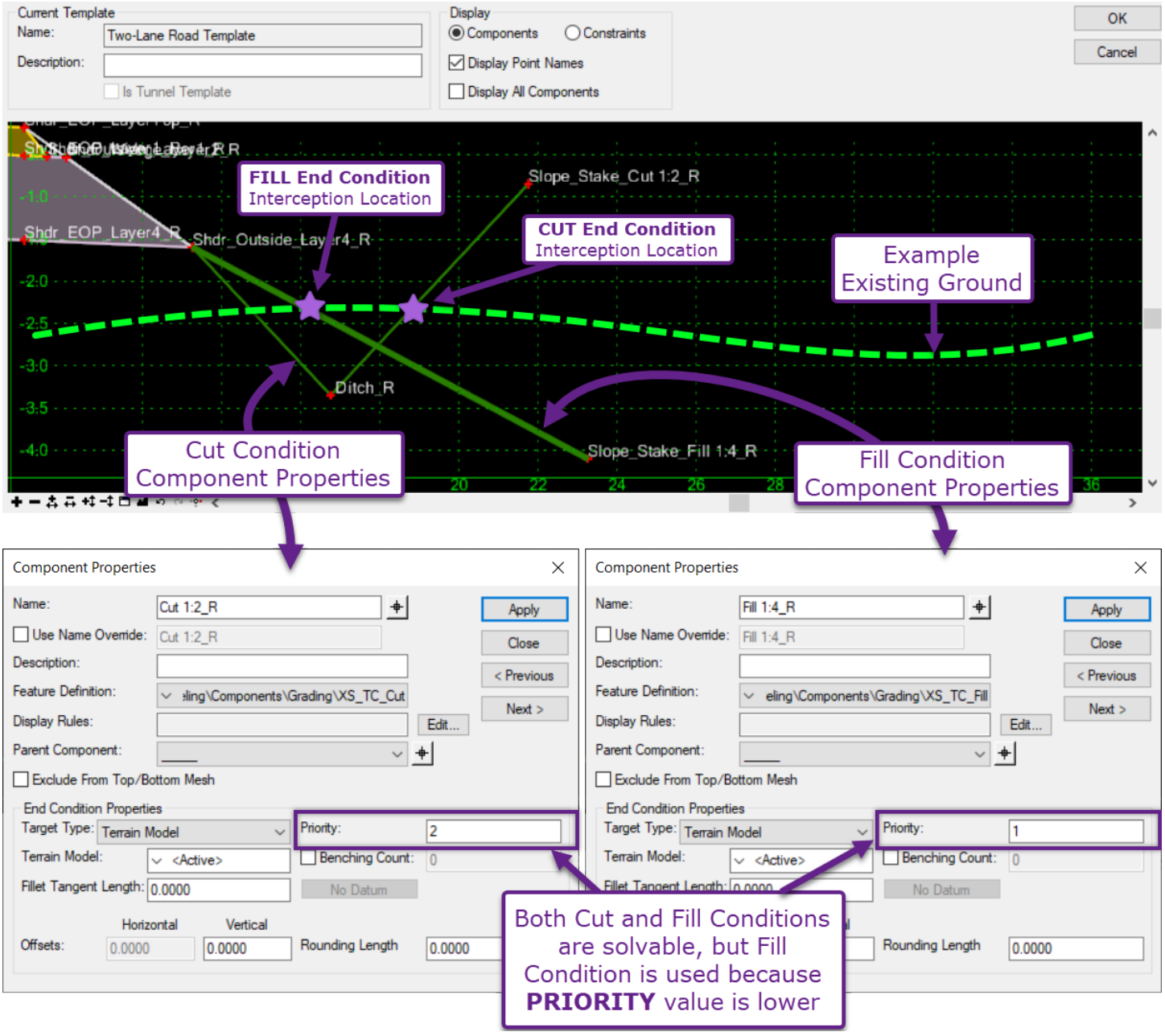
Templates with multiple End Conditions are usually set up to have the steepest End Condition with highest-value *Priority* and the *End Condition is Infinite* box checked. In other words, if all other End Conditions fail, the steepest End Condition will be infinitely extended until the Target is intercepted. If the steepest End Condition can NOT find the Target, then no End Conditions Components are constructed.

The *Test* tool can be used to explore how End Conditions behave when the Target position is moved. See *8C.7.f Testing Template End Conditions*.



8C.7.e Multiple End Conditions and End Condition Priorities

At a minimum, a Template should have two End Conditions Components branching from the hinge point: an End Condition Component for a Cut scenario and another for the Fill scenario. Multiple End Conditions stemming from a common point is referred to as an *End Condition Branch*. Only ONE End Condition in a Branch will be used per Corridor station. If more than one End Condition can be solved for in a Branch, then the End Condition with the lowest **Priority** value will be used.

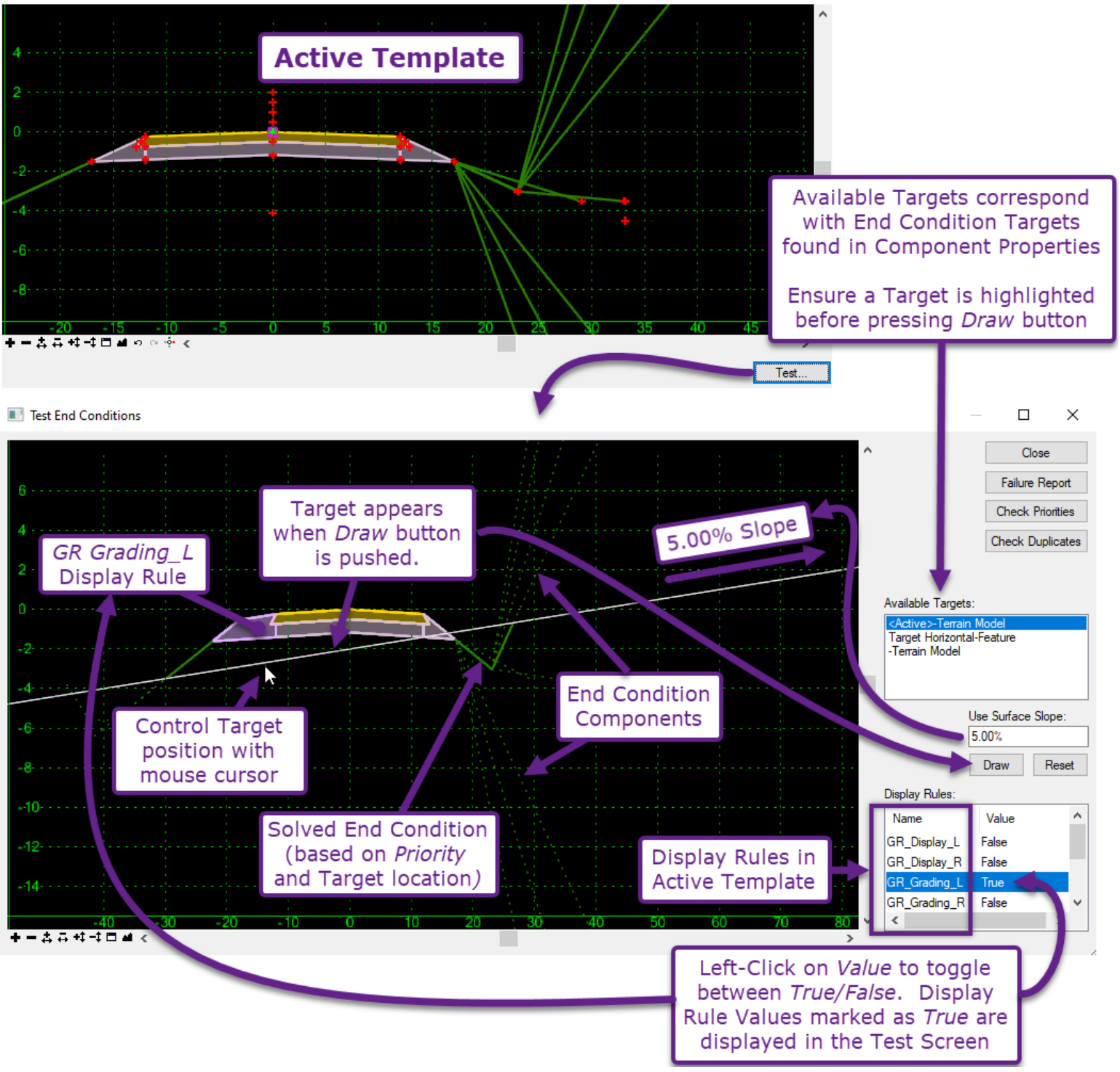


In the example shown above, both the Cut and Fill End Condition are solvable. For this particular Corridor station, the Fill End Condition is constructed used because it has a lower Priority value then the Cut End Condition.

Priority: If a Template has multiple End Conditions, the software will attempt to solve the End Condition with the lowest Priority value first. If a solution is found, no additional End Condition analysis is performed. If a solution is NOT found, then the next End Condition with the second lowest Priority value is analyzed. This process is repeated until a solution is found – assuming a solution is possible.

8C.7.f Testing Template End Conditions

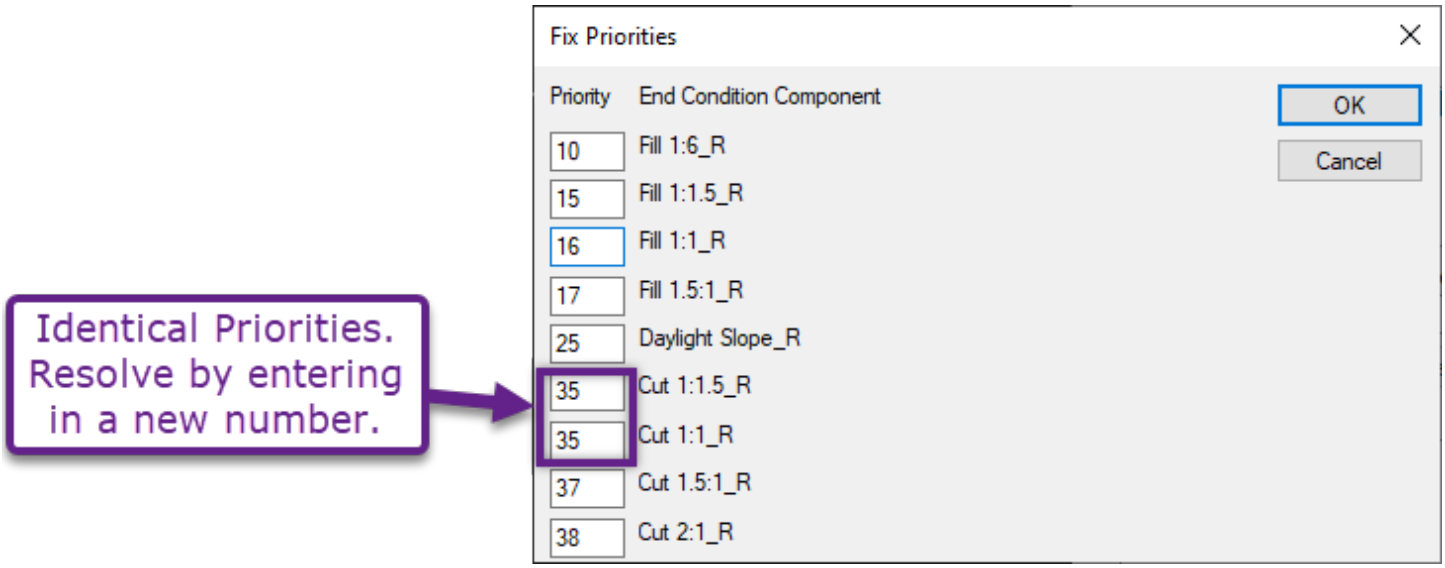
Using the *Test* tool, the Template can be tested to see how End Conditions behave. This tool can be used to quickly identify and resolve *Priority* conflicts within an End Condition Branch. To access the *Test* tool, Left-Click on the *Test…* button below the Active Template Editor Screen.



Failure Report: If an End Condition can NOT be solved for, then the Failure Report can be used to shows which End Condition Component failed and why.

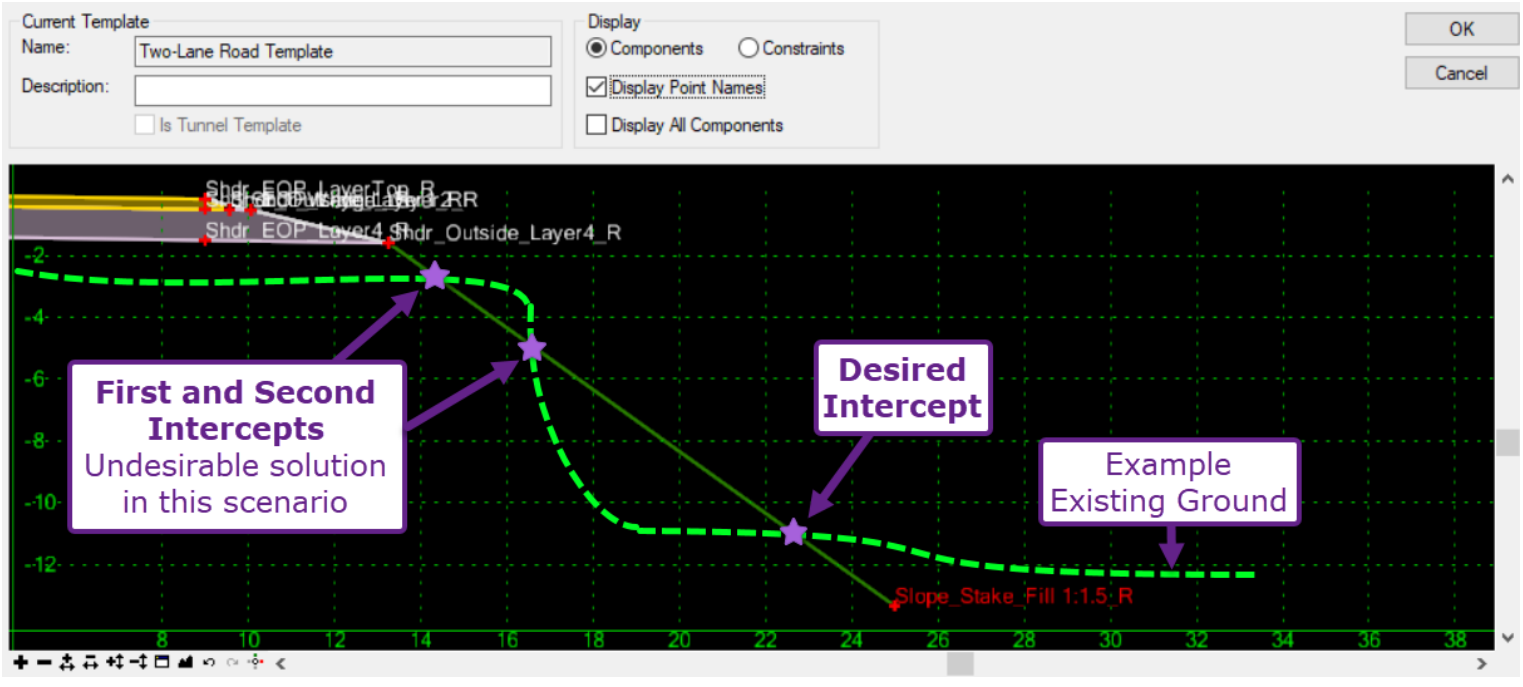
Check Duplicates: Checks for solutions that result in identical Names being used. Identical Names could be due to Feature Name Overrides.

Check Priorities – If an End Condition Branch has two or more End Condition Components with identical *Priority* values, then this option will identify the conflicting *Priorities* and allow the user to resolve them.

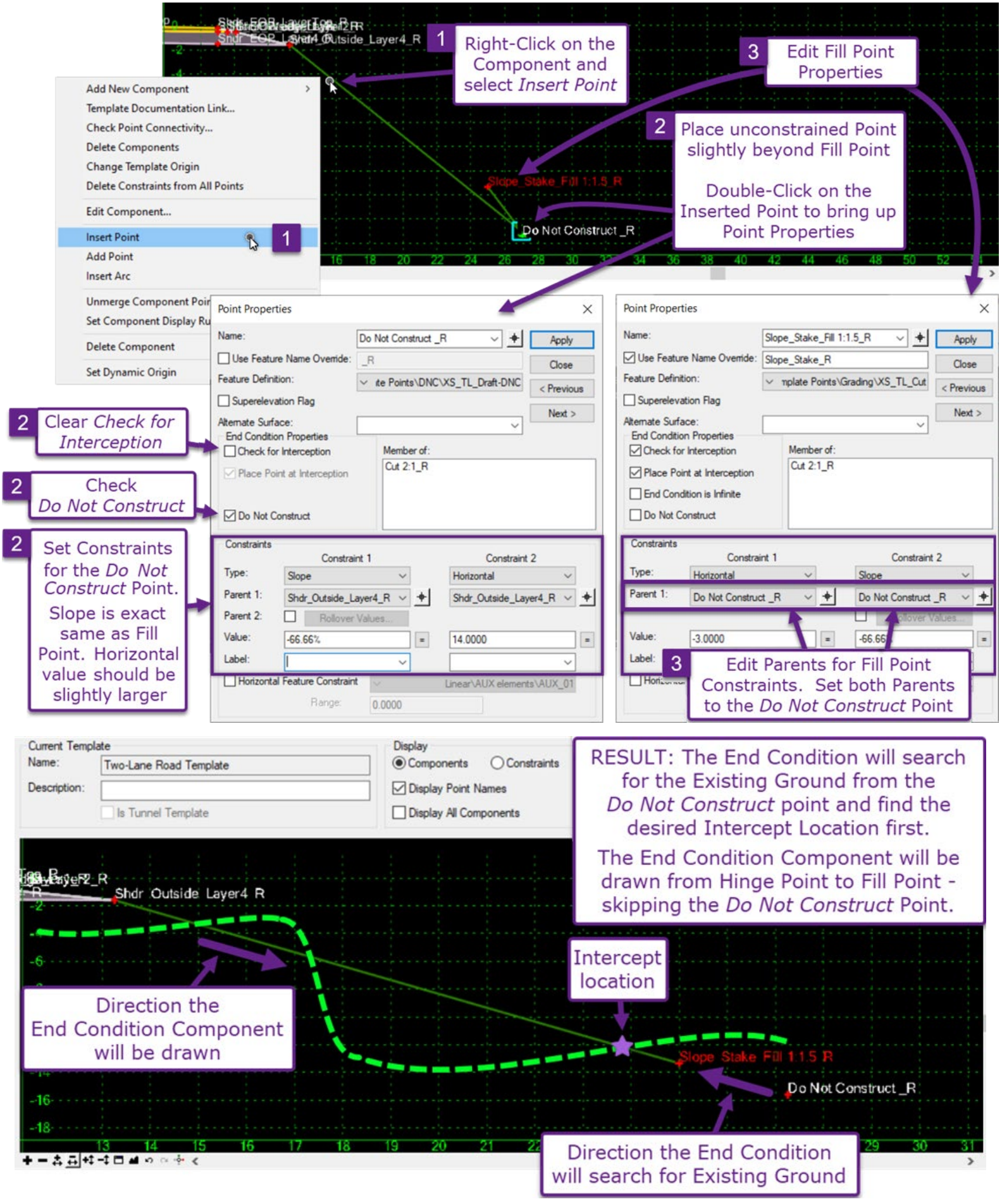


8C.7.g Place End Condition Point Past First Interception (Do Not Construct - Demonstration)

The *Do Not Construct* End Condition Point Property is useful when it is necessary to bypass the first solvable interception and search for a more desirable interception point further in the cross-section. A scenario where this is necessary is shown below. In this example, it necessary to project the fill slope to the bottom of the eroded embankment by passing through the undesirable intercepts located further up.



This scenario can be addressed by creating a *Do Not Construct* point within the End Condition Component.


1
Right-Click on the Component and select *Insert Point*
3
Edit Fill Point Properties
2
Place unconstrained Point slightly beyond Fill Point
Double-Click on the Inserted Point to bring up Point Properties
Add New Component
Template Documentation Link...
Check Point Connectivity...
Delete Components
Change Template Origin
Delete Constraints from All Points
Edit Component...
Insert Point
Add Point
Insert Arc
Unmerge Component Poir
Set Component Display Ru
Delete Component
Set Dynamic Origin
Slope_Stake_Fill 1:1.5_R
Do Not Construct _R
Point Properties
Name:
Do Not Construct _R
Apply
Use Feature Name Override:
_R
Close
Feature Definition:
ite Points\DNC\XS_TL_Draft-DNC
< Previous
Superelevation Flag
Next >
Alternate Surface:
End Condition Properties
Check for Interception
Member of:
Cut 2:1_R
Place Point at Interception
Do Not Construct
Constraints
Constraint 1
Constraint 2
Type:
Slope
Horizontal
Parent 1:
Shdr_Outside_Layer4_R
Parent 2:
Rollover Values...
Value:
-66.66%
14.0000
Label:
Horizontal Feature Constraint
Linear\AUX elements\AUX_01
Range:
0.0000
2
Clear *Check for Interception*
2
Check *Do Not Construct*
2
Set Constraints for the *Do Not Construct* Point.
Slope is exact same as Fill Point. Horizontal value should be slightly larger
Point Properties
Name:
Slope_Stake_Fill 1:1.5_R
Apply
Use Feature Name Override:
Slope_Stake_R
Close
Feature Definition:
mplate Points\Grading\XS_TL_Cut
< Previous
Superelevation Flag
Next >
Alternate Surface:
End Condition Properties
Check for Interception
Member of:
Cut 2:1_R
Place Point at Interception
End Condition is Infinite
Do Not Construct
Constraints
Constraint 1
Constraint 2
Type:
Horizontal
Slope
Parent 1:
Do Not Construct _R
Do Not Construct _R
Value:
-3.0000
Label:
3
Edit Parents for Fill Point Constraints. Set both Parents to the *Do Not Construct* Point
Current Template
Name:
Two-Lane Road Template
Description:
Is Tunnel Template
Display
Components
Constraints
Display Point Names
Display All Components
RESULT: The End Condition will search for the Existing Ground from the *Do Not Construct* point and find the desired Intercept Location first.
The End Condition Component will be drawn from Hinge Point to Fill Point - skipping the *Do Not Construct* Point.
Shdr_Outside_Layer4_R
Intercept location
Direction the End Condition Component will be drawn
Slope_Stake_Fill 1:1.5_R
Do Not Construct _R
Direction the End Condition will search for Existing Ground

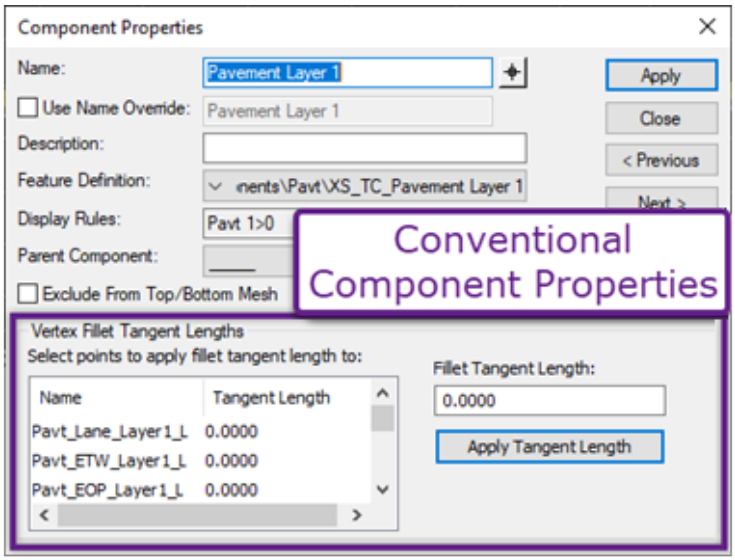
8D – TEMPLATE COMPONENTS

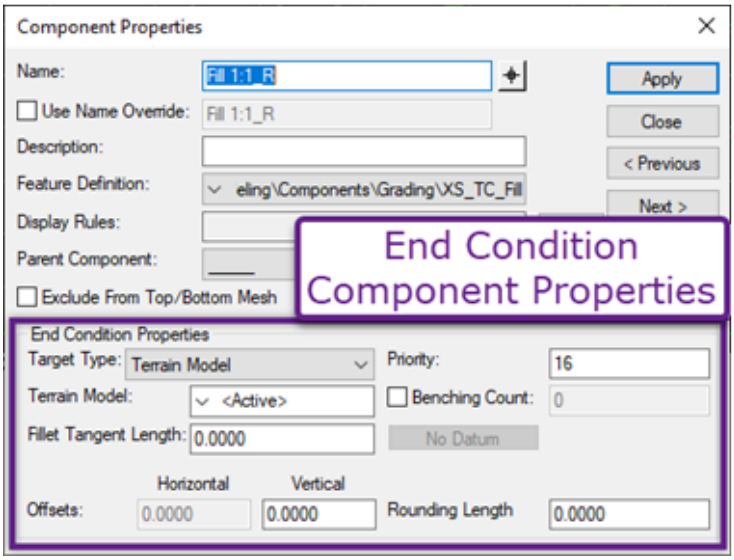
There are three types of Template Components: **Conventional**, **End Condition**, and **Overlay/Stripping**. The Component type can be identified by the options available in the Component Properties.

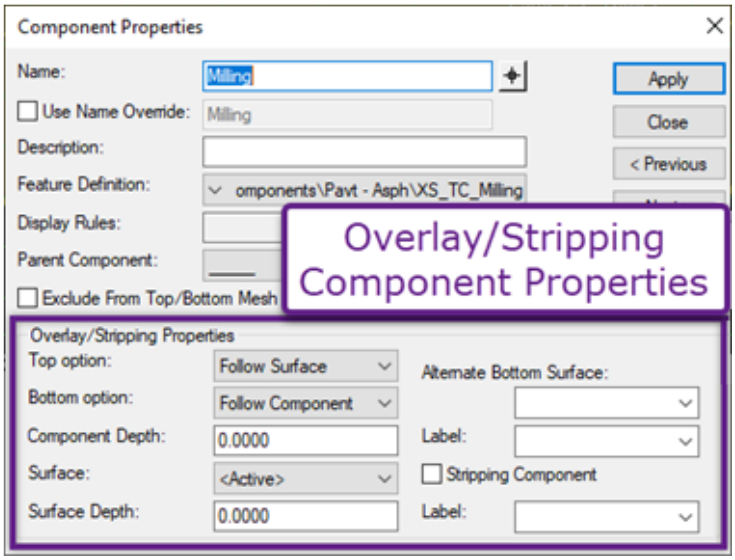
Conventional Components: Conventional Components are simply referred to as Components in this manual. A Conventional Component can be Volumetric (enclosed) or Planar (open). An example of a Volumetric Conventional Component is asphalt or aggregate. An example of a Planar Component is a geotextile fabric.

End Condition Components: End Condition Components are dynamic and used to intercept a Target, such as a Terrain Model, single Elevation, or Linear Element. End Condition Components are always Planar.

Overlay/Stripping Components: Overlay/Stripping Components are dynamic and have the ability to adjust the Component's depth to follow a surface. The most common use of Overlay/Stripping Components is to model milling and leveling course depths in relation to the existing ground surface. Overlay/Stripping Components appear Planar but will become volumetric as the top or bottom of the Overlay/Stripping Component dynamically repositions to follow the surface of a Terrain Model.







8D.1 Component Properties

Component Properties include the Feature Definition, Name, Name Override assigned to a Component.

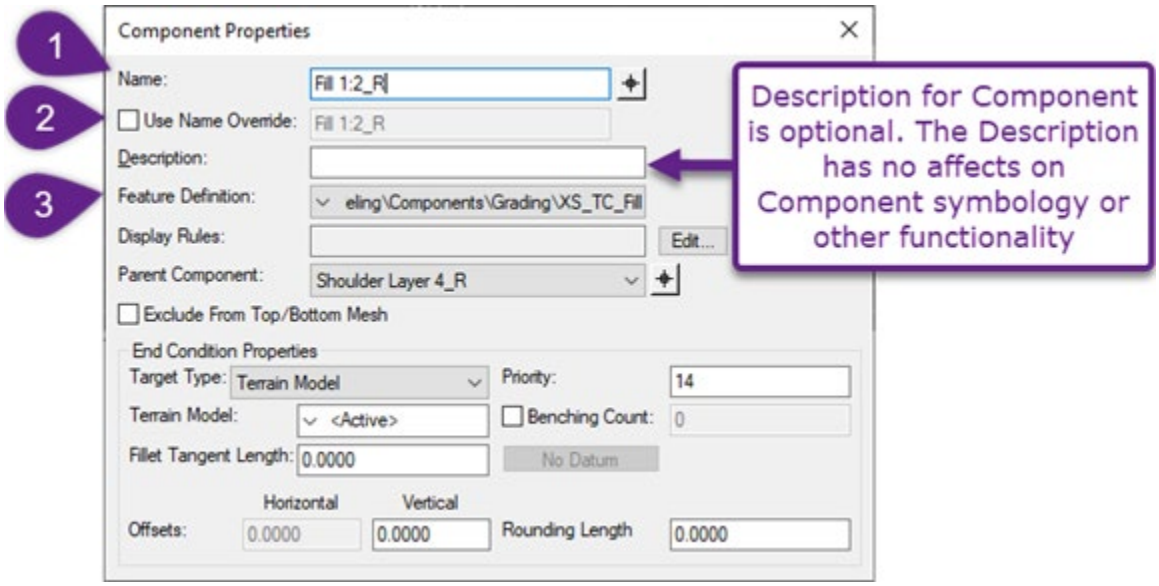
3 **Feature Definition:** The Feature Definition determines the Level and Color of a Component as it is shown in the *3D Design Model*, Dynamic Cross Section Viewer, and Cross Section sheets.

> ***IMPORTANT:*** When quantities are reported for a Corridor, the Component volumes and areas are summed up by Feature Definition. The Feature Definition must correspond with the real-world material that the Component represents. Components of the same material type should be placed on the same Feature Definition.

1 **Name:** The Name is used to identify a Template Component. Every Component within a Template MUST have a unique Name.

> ***BEST PRACTICE:*** Template Components should be given a logical name AND contain the suffix _R or _L when appropriate.
>
> An example of an appropriate name for an End Condition Component is *Fill 1:2_R*. This name describes the type and slope of the End Condition (Fill 1V:2H) and the side it falls on (_R).

2 **Use Name Override:** If this box is CHECKED, then the Component Name is overridden to whatever is typed into the adjacent box. This property option is used to overcome the aforementioned condition that Template Components must have different Names within the same Template. It may be useful to have matching Names for Components that are used conditionally, as a result of Display Rules. However, Name Overrides are NOT used in common practice.



8D.2 Display Rules

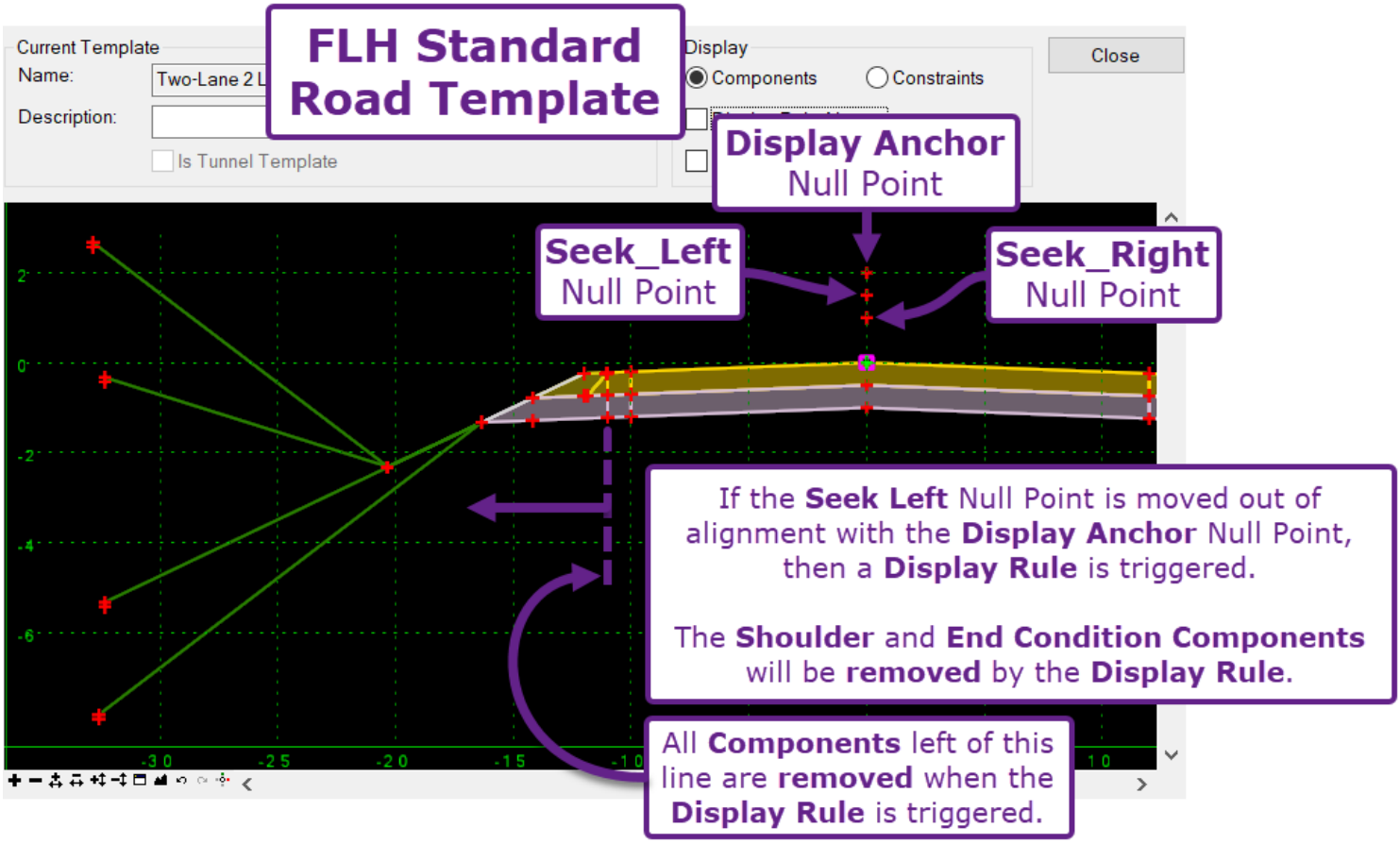
Display Rules are used to conditionally display Components. Typically, Display Rules are used in conjunction with a Null Point. The Null Point is programmed with a *Horizontal Feature Constraint* to seek out Linear Elements of a specified Feature Definition in the *2D Design Model*. If the Null Point can find a specific 2D Linear Element, then the Null Point is displaced from its default position, which triggers the Display Rule to display or undisplay a set of Component(s).

In the most common use, Display Rules are used to turn OFF the Shoulder and End Conditions Components in the vicinity of intersections, approaches, and driveways. As explained in the next section, FLH Standard Road Templates contain Display Rules for this purpose. Alternatively, the process for creating this type of Template is shown in *8F.4 Mainline Road Template with Display Rules for Managing Approach Roads and Driveways*.

Another common use of Display Rules is to turn ON guardrail and shoulder widening components for a stretch of guardrail. The process for creating this type of Template is shown in *8F.3 Advanced Road Template with Guardrail and Display Rules*.

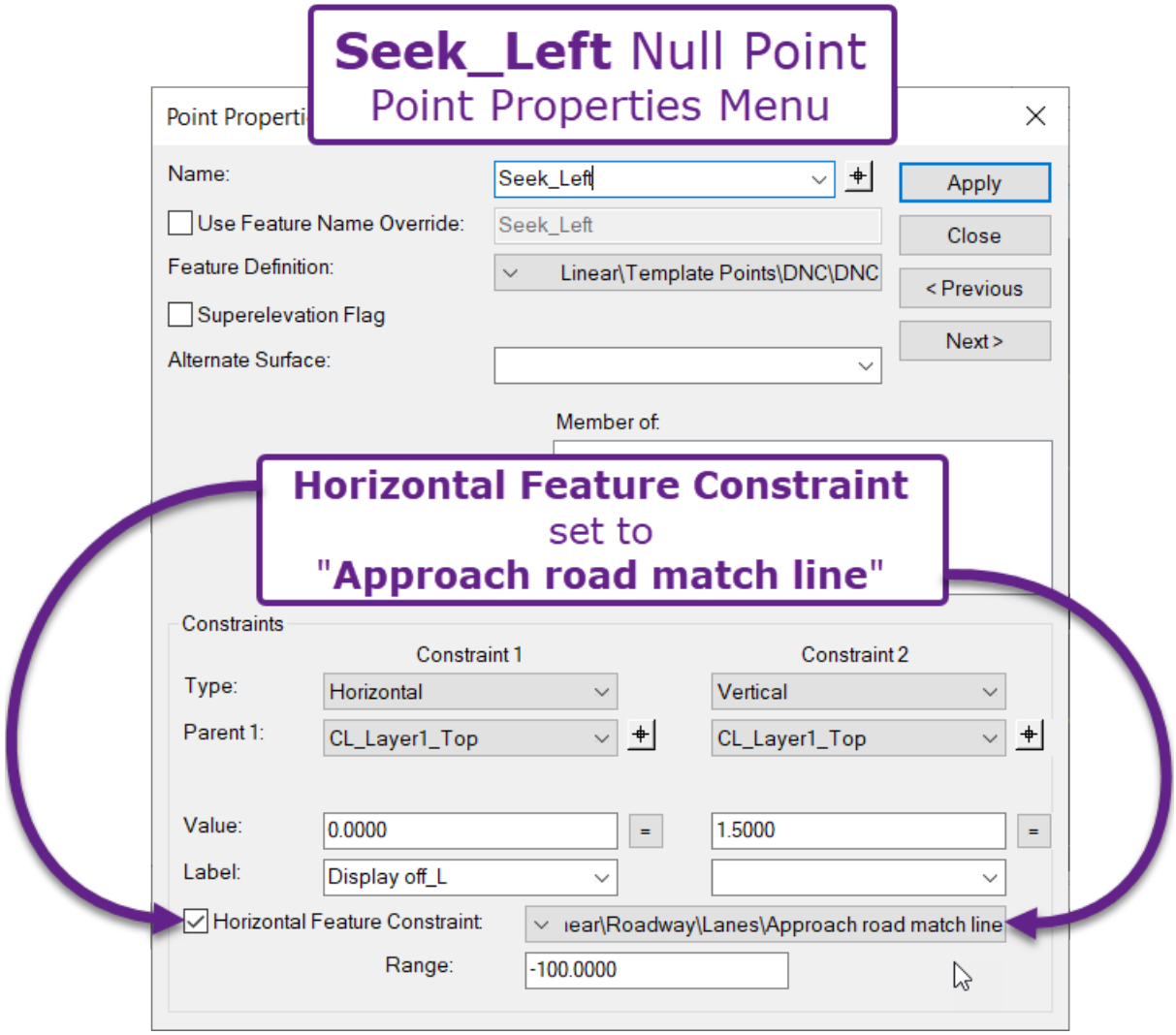
8D.2.a Display Rules in the FLH Standard Road Templates for Approaches

In the FLH Library, all road Templates are built with Display Rules to turn OFF the Shoulder and End Condition Components in the vicinity of intersections, approaches, and driveways.



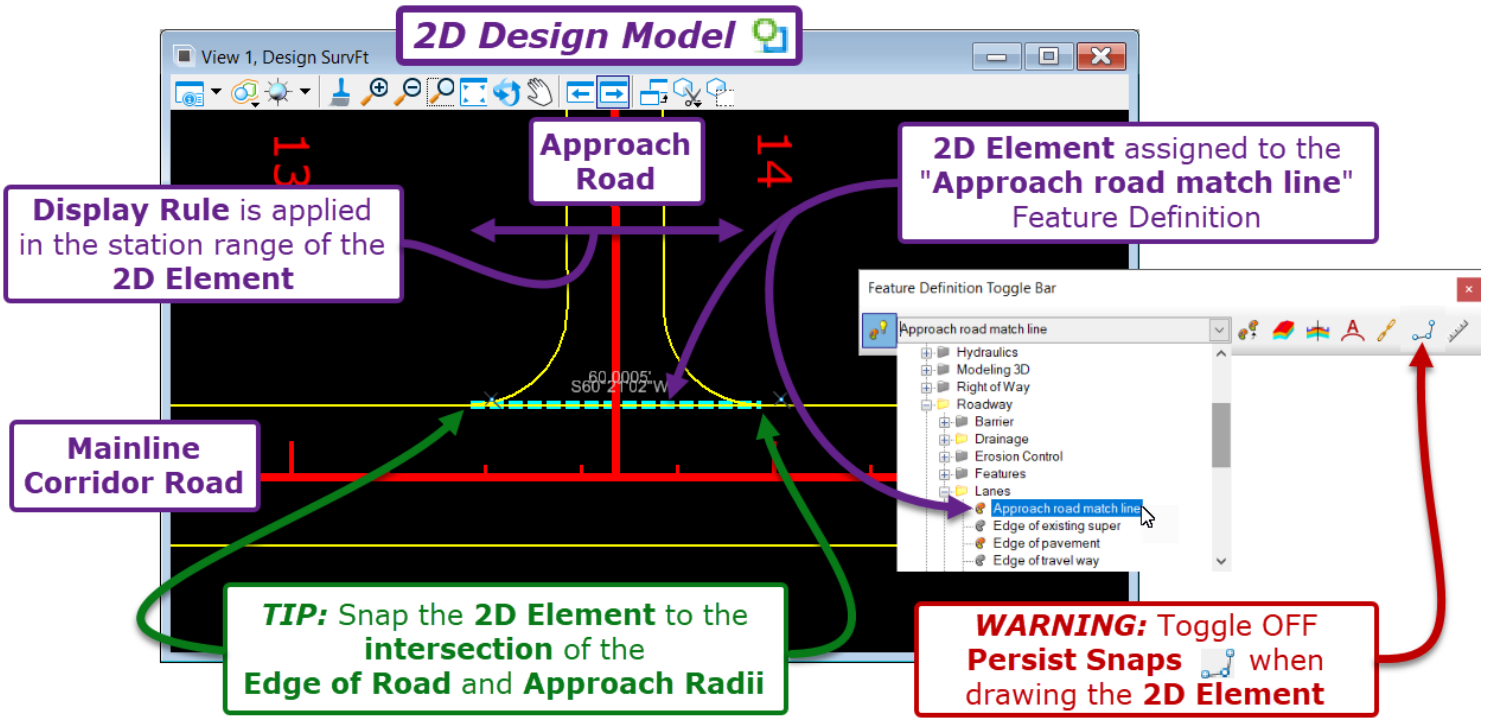
The **Seek_Left** and **Seek_Right** Null Points are programmed with *Horizontal Feature Constraints* that seek out 2D Elements assigned to the "Approach road match line" Feature Definition. The **Display Anchor** Null Point will always remain in its default position. A Display Rule is triggered when either the **Seek_Left** or **Seek_Right** Null Point is moved out of vertical alignment with the **Display Anchor** Null Point.

The Point Properties for the **Seek_Left** Null Point is shown below. The **Seek_Left** Null Point is programmed with a *Horizontal Feature Constraint* with the Feature Definition set to "Approach Road Match Line".



To move the **Seek_Left** Null Point from its default position, a 2D Element must be drawn in the *2D Design Model*. The 2D Element must be assigned to the "Approach Road Match Line" Feature Definition to displace the **Seek_Left** Null Point from its default position, via the *Horizontal Feature Constraint*.

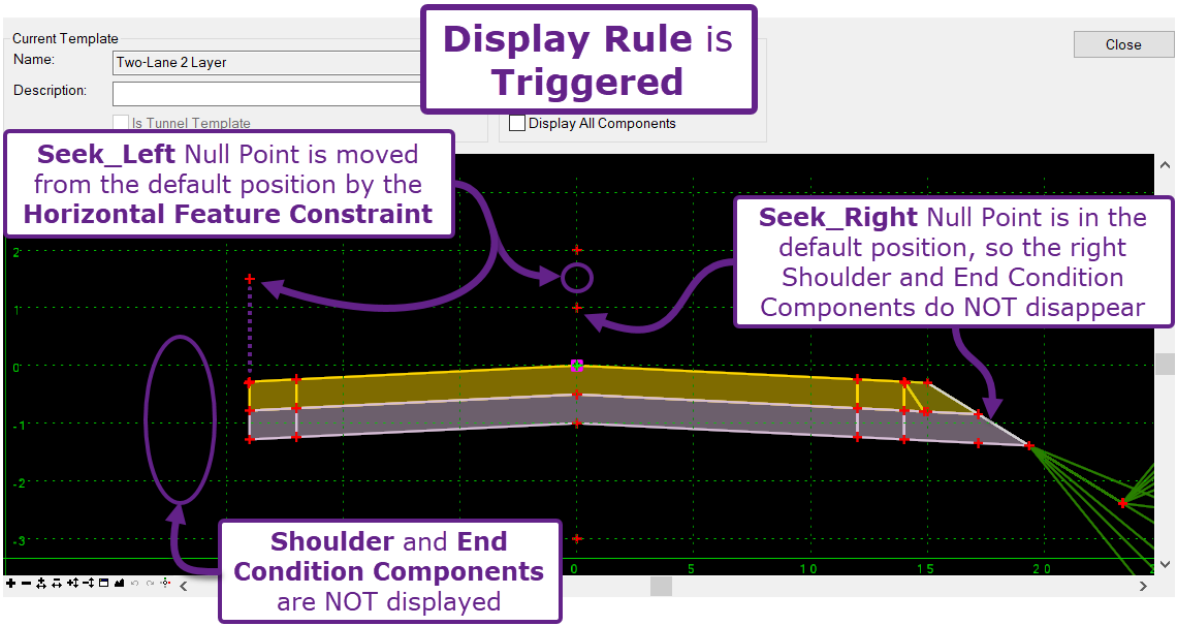
The 2D Element should be drawn after the Corridor is created. After the 2D Element is drawn, it must be added to the Corridor as a Corridor Reference. **A 2D Element will NOT be used as a *Horizontal Feature Constraint* until it is added as a Corridor Reference.** See *9G.9 Corridor References*.



WARNING: **When drawing the 2D Element toggle OFF the Persist Snap setting.** If Persist Snaps are turned ON, then a dynamic Persist Snap dependency is formed between the 2D Element and the Corridor Element (i.e., edge of road line) that was snapped to.

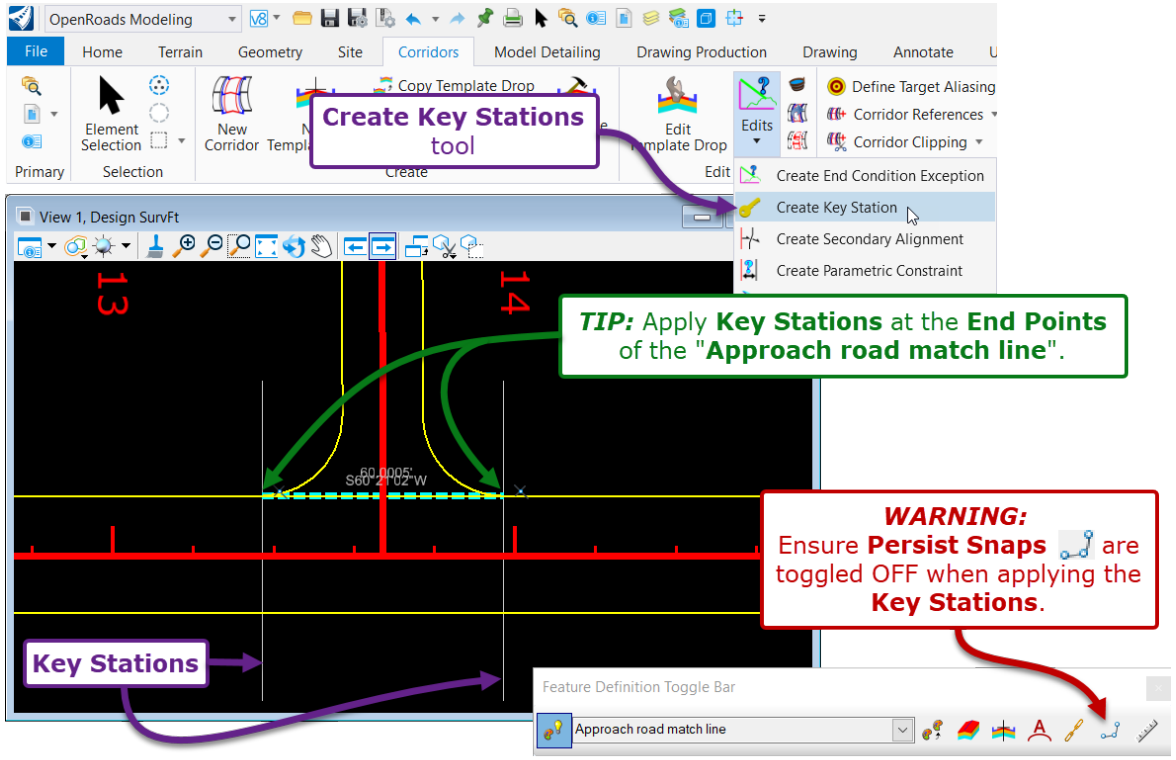
After the 2D Element is drawn, it must be added to the Corridor as a Corridor Reference. The 2D Element is REJECTED as a Corridor Reference if a Persist Snap is formed with a Corridor Element (i.e., edge of road line). If the 2D Element was drawn with Persist Snap, it is rejected because adding the element to the Corridor would result in a circular reference.

As shown below, the *Horizontal Feature Constraint* moves the **Seek_Left** Null Point out of vertical alignment with the **Display Anchor** Null Point, which triggers the Display Rule attached to the Shoulder and End Condition Components. The Shoulder and End Condition Components are removed in the station range of the 2D Element.



TIP: Apply *Key Stations* to the Corridor at the end points of the "Approach road match line" element. The Key Stations ensure the Corridor is processed for the exact full distance of the "Approach road match line" element. Key Stations are discussed in *9G.3 Key Stations*.

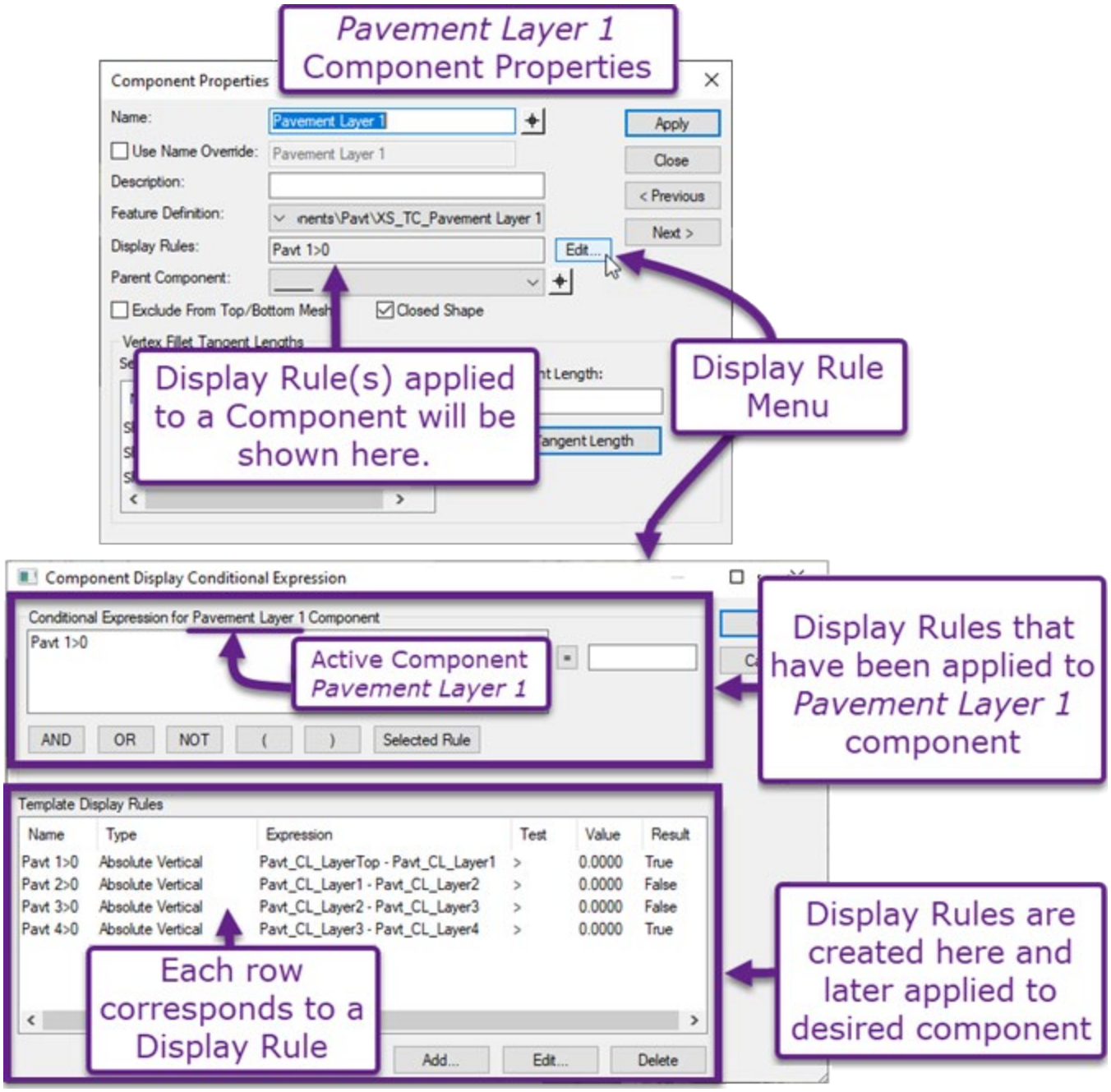
WARNING: The **Persist Snap** toggle must be OFF when placing the Key Stations. If NOT the Key Stations will NOT be created because a circular reference would be formed with the "Approach road match line" Corridor Reference. The Persist Snap toggle is in the Feature Definition Toolbar. See *7B.3 Feature Definition Toolbar*.



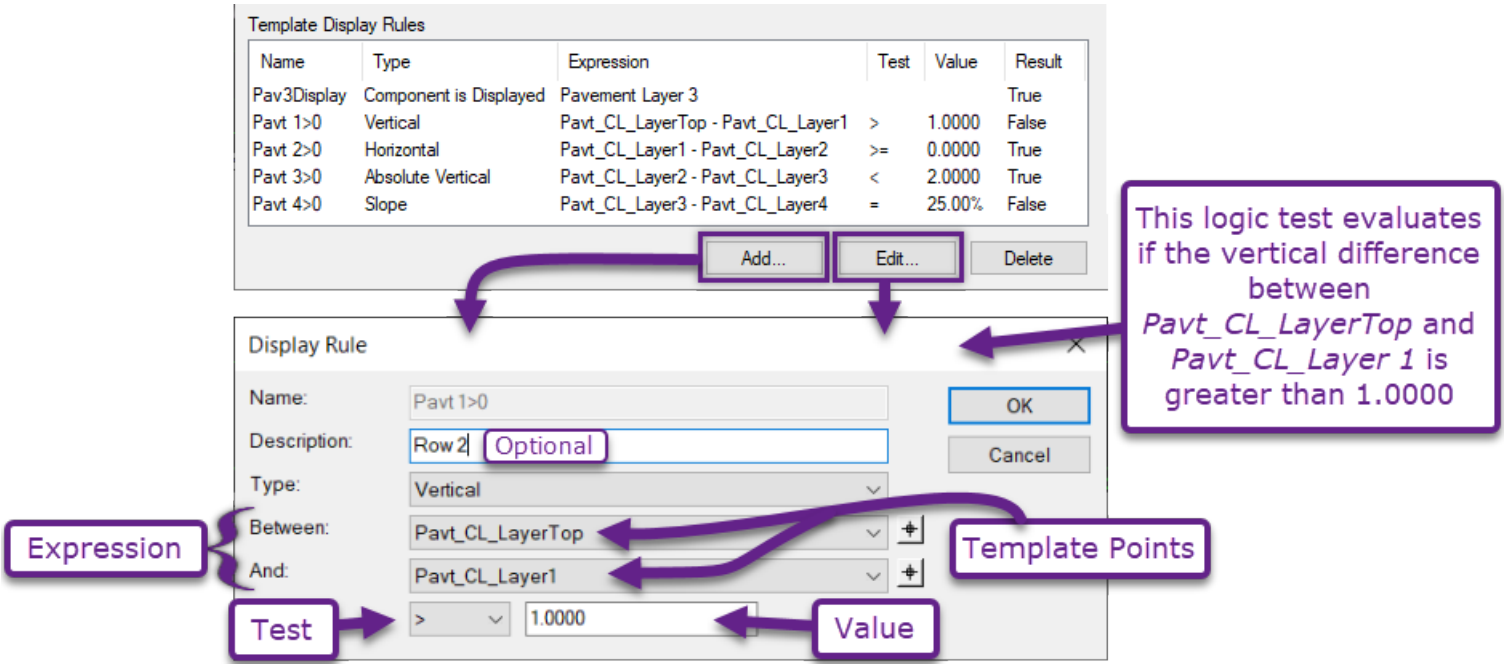
8D.2.b Operation of Display Rules

Display Rules applied to a particular Component is shown in the Component Properties menu. Display Rules will conditionally display the Component if certain criteria are met. Display Rules require a logic test, which is based on the position of two Template Points. If the logic test is found to be *True*, the Component is displayed and constructed in the Corridor model. If the logic test is found to be *False*, the Component is NOT displayed.

Display Rules are created and applied to a component in the *Display Rule Menu*, which is accessed in the Component Properties menu. The Display Rule Menu is divided into two areas. The top portion is where the Display Rules for the Active Component are shown and applied. The bottom portion lists every Display Rule contained in the whole Template. The bottom portion is also where Display Rules are created.



8D.2.c Template Display Rule list



Name – is only used to identify Display Rules. Name has no effect on Display Rule calculations.

Type – denotes the type of logic test and corresponding coordinate calculation.

Type:	Description:
Horizontal	The horizontal difference between two Template Points is calculated. For example, if the first Template Point has a horizontal value of 4 and the second Point has a horizontal value of 6 the calculated value is -2.
Absolute Horizontal	The horizontal difference between two Template Points is calculated and the absolute value is returned. For example, if the first Template Point has a horizontal value of 4 and the second Point has a horizontal value of 6 the calculated value is +2.
Vertical	The vertical difference between two Template Points is calculated.
Absolute Vertical	The vertical difference between two Template Points is calculated and the absolute value is returned.
Slope	The slope between two Template Points is calculated.
Absolute Slope	The horizontal and vertical difference between two Template Points is calculated and the absolute value for the rate of change (slope) is returned.
Component is Displayed	This test simply evaluates whether a different Template Component is displayed. An example may be: show Template Component X if Template Component Y is being displayed. This is useful if Template Component Y has its own set of Display Rules.

Expression – shows the Template Points or Component that are being tested. The Expression is performed with two Template Points or one Component.

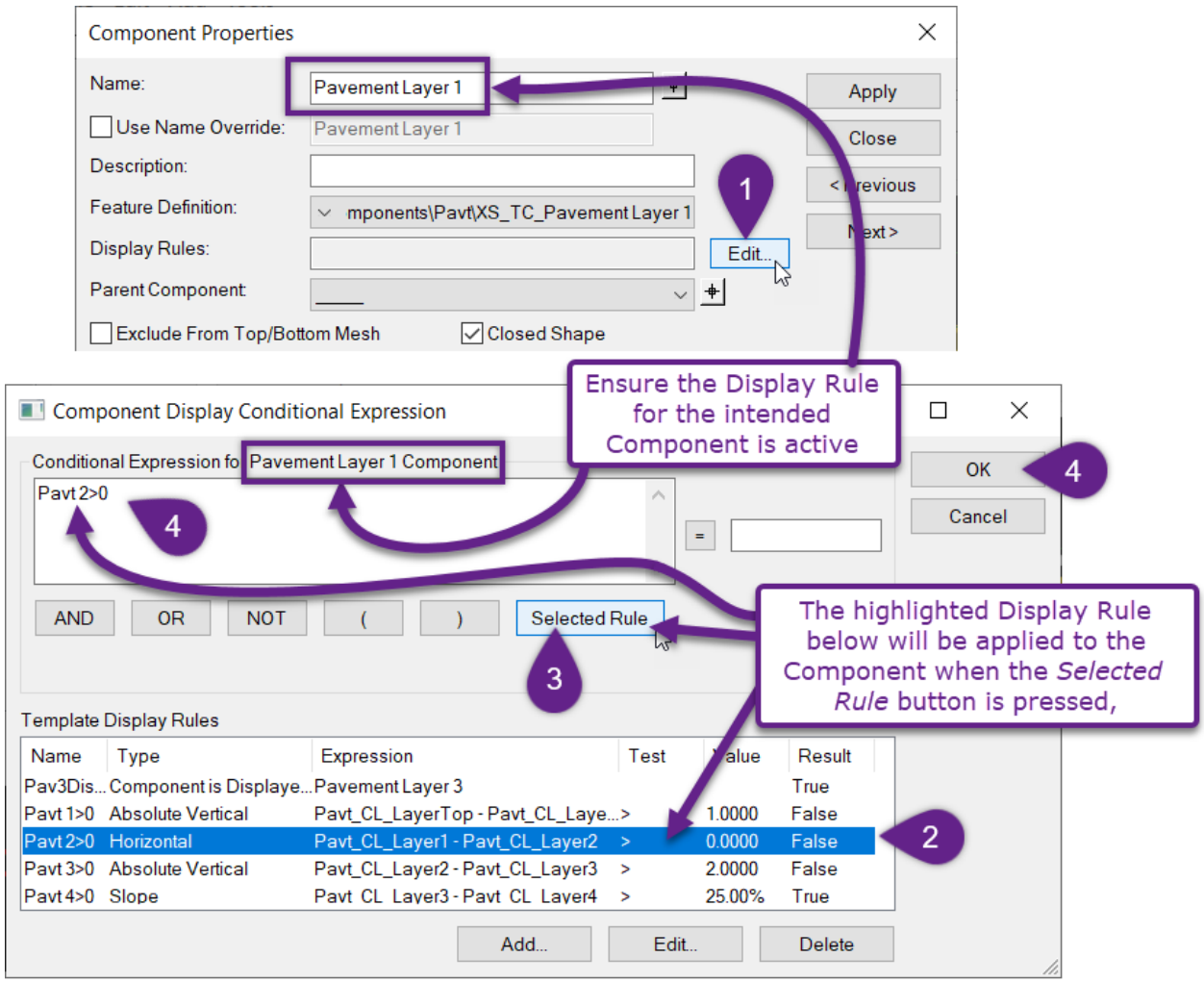
Test – The logic operator used to set the criteria for the test. Operators include:
> greater than >= greater than or equal to = equal to < less than <= less than or equal to

Value – The numerical range or slope value used to set the criteria for the test.

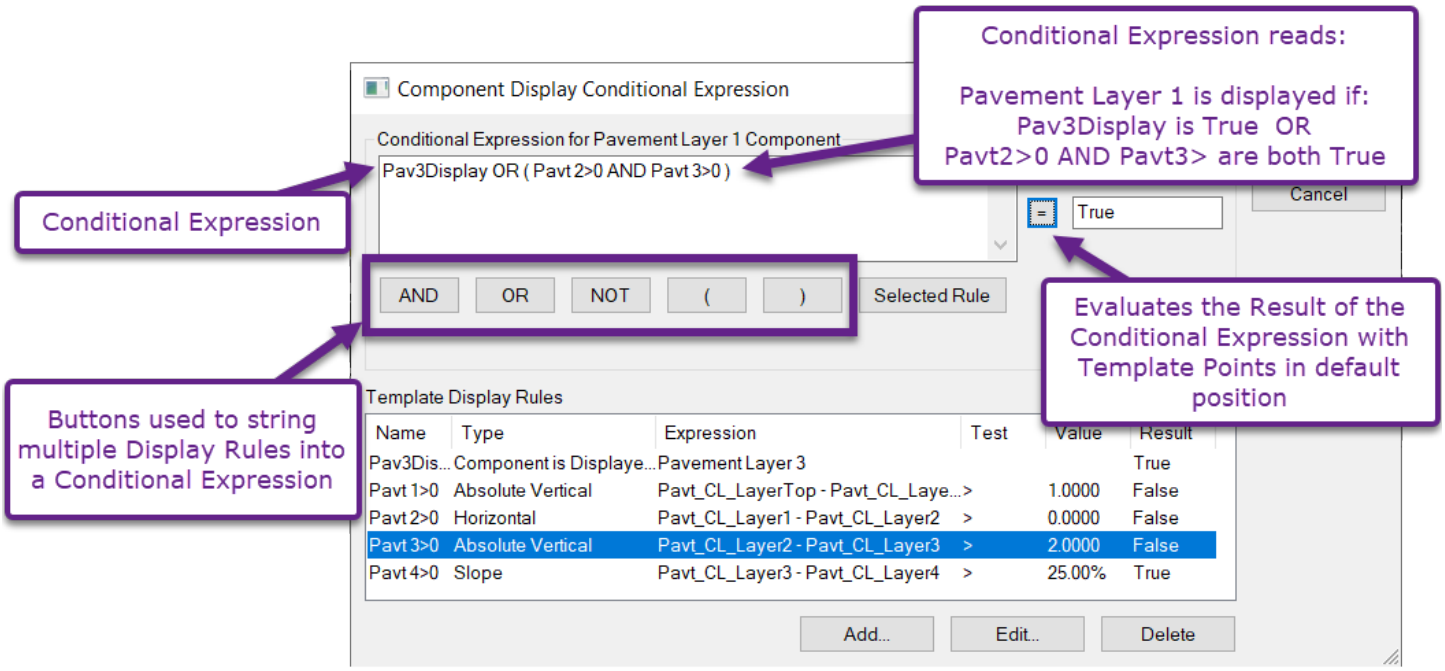
Result – Evaluates the result of the Test with Template Points as shown in the Template. If the logic test is valid with the Template configuration, as displayed in the Template Editor, then the Result will show as True. If the logic test is failed, then the Result will show as False.

8D.2.d Apply Display Rules to a Component

The graphic below shows the basic workflow for applying a Display Rule to a component:



1	Open the *Component Properties* for the desired Component to apply Display Rules to. Left-Click on the *Edit* box that is adjacent to the *Display Rules* property.
2	Left-Click on (highlight) the specific Display Rule to be applied to the Component in the Template Display Rule list.
3	Left-Click on the *Selected Rule* button to add it to the Component Display Rule list.
4	Ensure that the Name of the intended Display Rule has been added to the Component Display Rule list and push the *OK* button to apply the Display Rule.



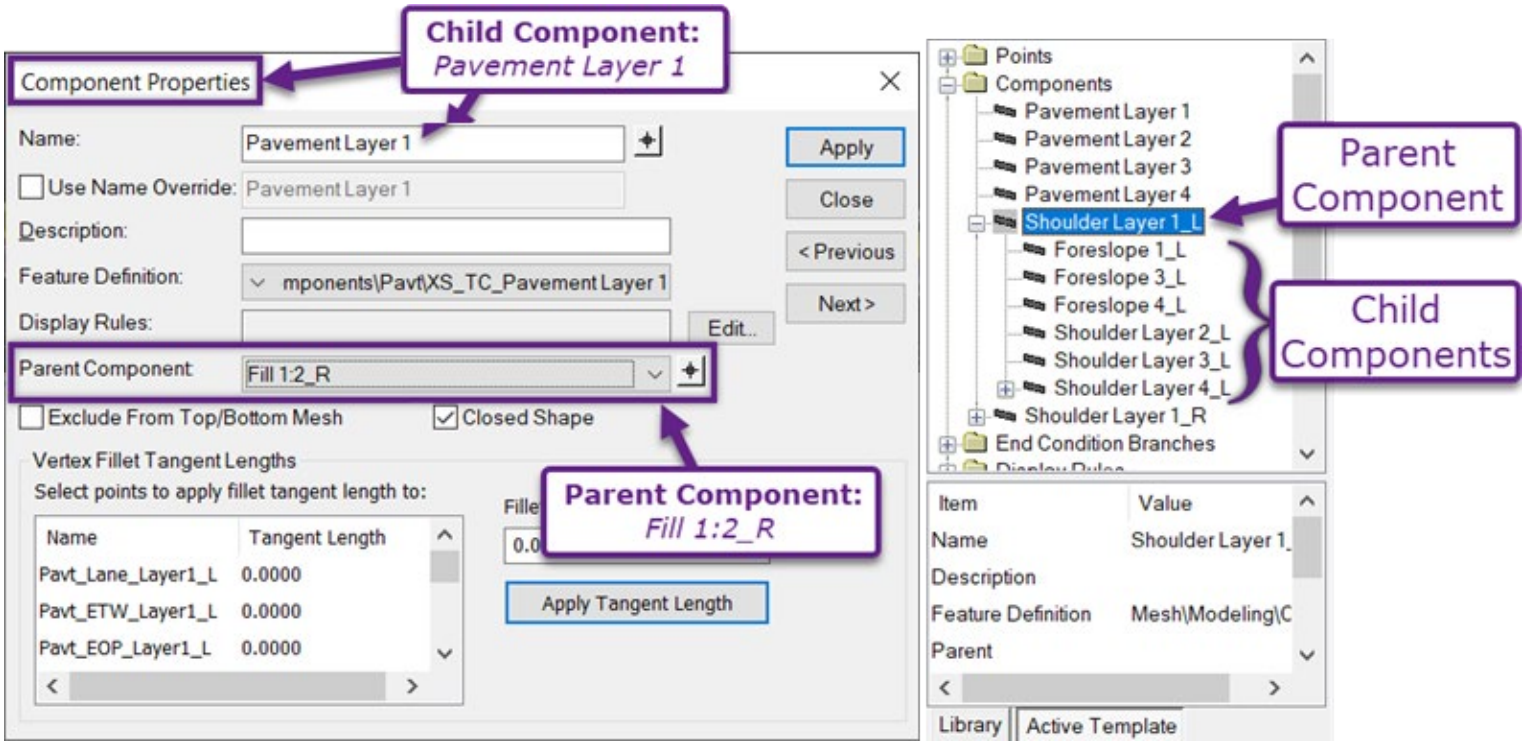
8D.3 Parent Components

A *Child Component* will only be displayed if it's *Parent Component* is displayed.

Parent Component functionality is used in conjunction with Display Rules to create Advanced Templates. Similarly, *Child Components* can be assigned to an End Condition *Parent Component*. The *Child Components* are only displayed if the End Condition *Parent Component* is solved for.

Parent and Child relationships are shown in the *Active Template* list.

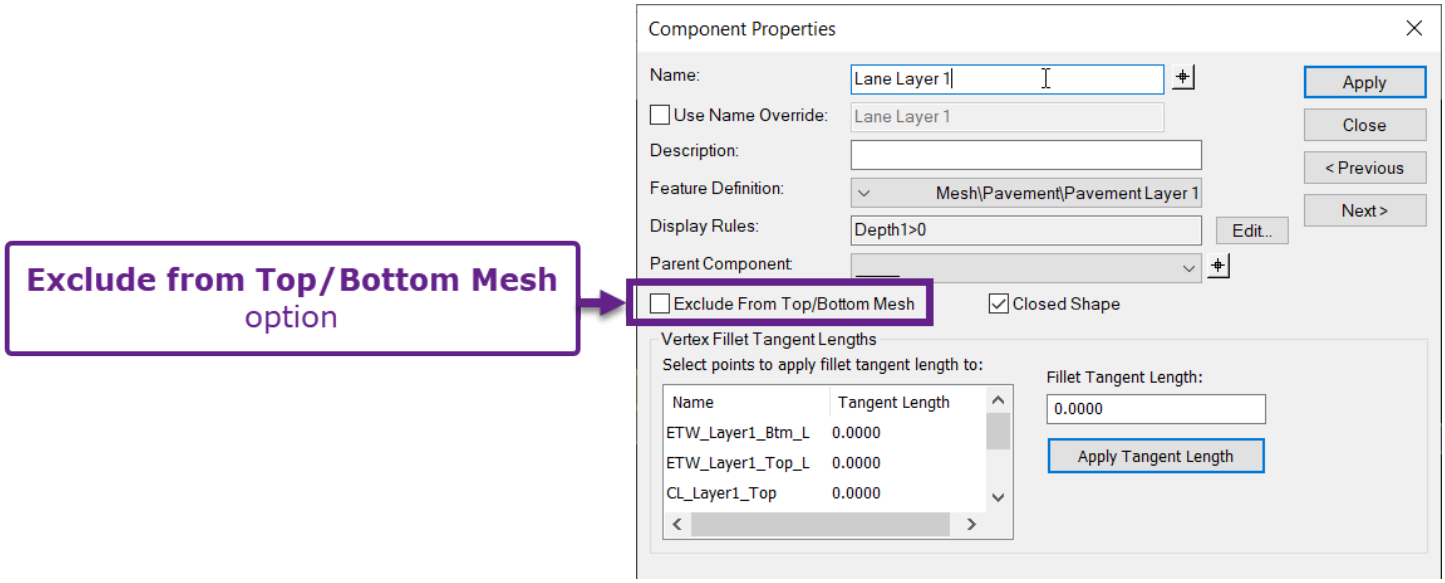
In graphic below, the Child Component [*Pavement Layer 1*] will only be displayed if the Parent Component [*Fill 1:2_R*] is displayed.

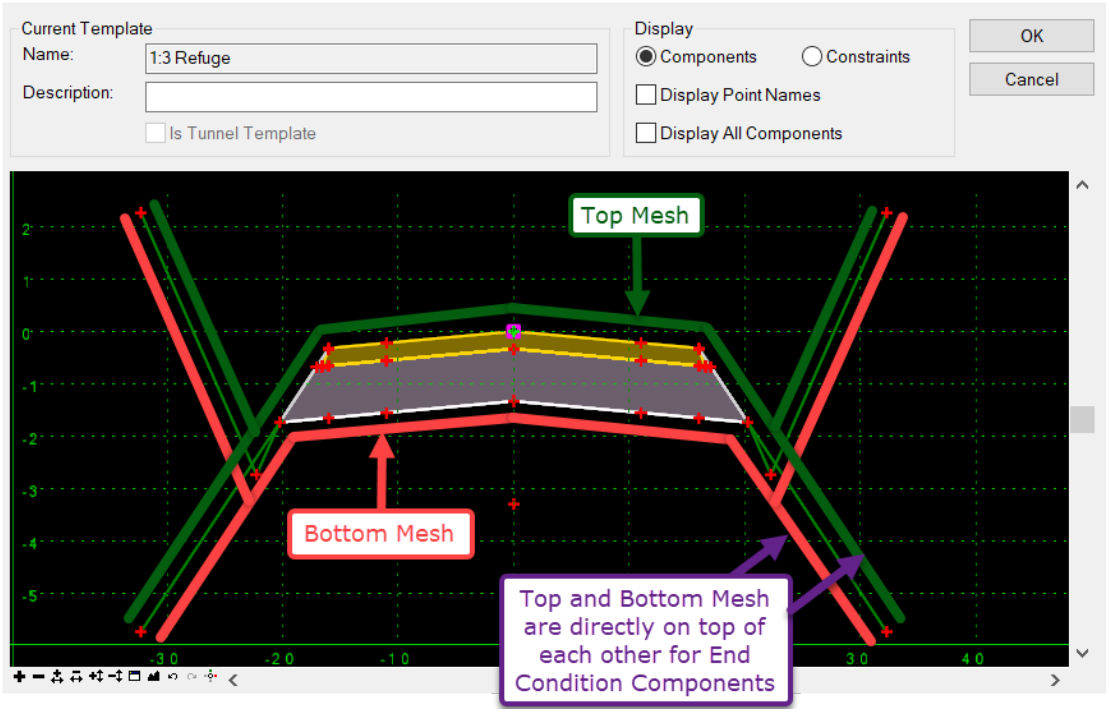


8D.4 Exclude From Top/Bottom Mesh

Top and Bottom Mesh elements are created with the Corridor. The Top Mesh represents the top surface of the Corridor. The Bottom Mesh represents the bottom surface of the Corridor. This option determines if the selected Component is included in either the Top or Bottom Mesh. This option is only consequential if the Component lies on the Top or Bottom Mesh line. Interior or intermediate Components are NOT affected by this option.

For a detailed explanation of what comprises the Top and Bottom Mesh - See *9I.1 Top and Bottom Meshes*.





If this box is UNCHECKED, the selected Component will be included in the Top or Bottom Mesh – assuming the component falls on the Top or Bottom Mesh line.

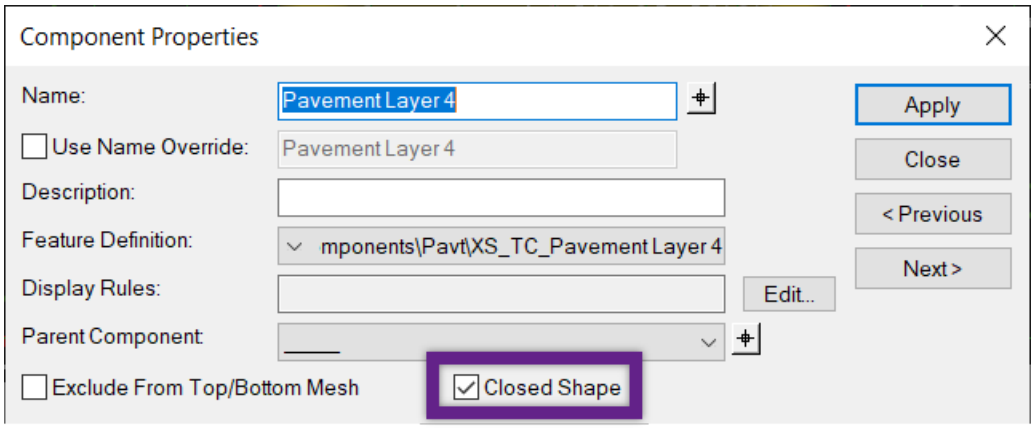
If this box is CHECKED, the component is skipped and NOT included into the Top or Bottom Mesh.

NOTE: Top and Bottom Mesh elements are rarely utilized in recent versions of the ORD Software. For versions of the software released before the year 2020, the Top and Bottom Mesh elements were used to calculate earthwork quantities and create Finished Grade and Subgrade Terrain Models. Using the Top and Bottom Mesh elements to create a Terrain Model is shown in *22A.2 Select Mesh Elements to Create the Terrain Model*. However, this workflow is discouraged.

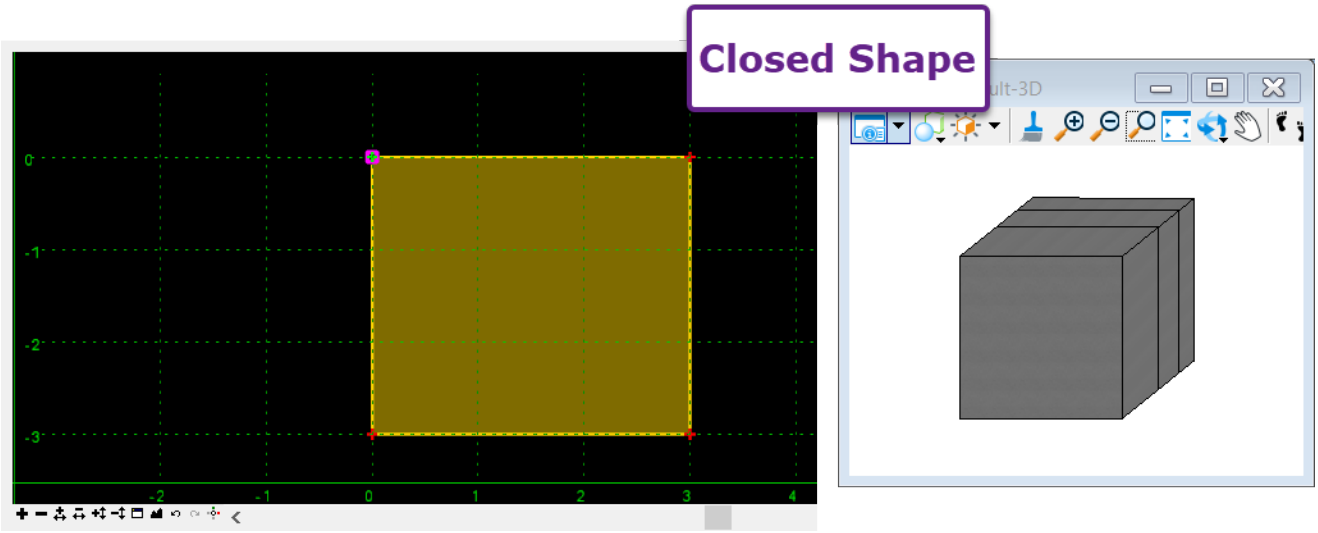
In recent versions of the ORD software (year 2020 and later), the recommended workflow for creating a Terrain Model is with the *Create Terrain Model from Design Meshes* tool, which is shown in *22A.1 Create Terrain Model from Design Meshes tool*. Similarly, the Top/Bottom Meshes are NOT relevant to earthwork calculations in recent versions of the ORD Software. For more information on quantity calculations, see *Chapter 20 – Quantities*.

8D.5 Closed Shape

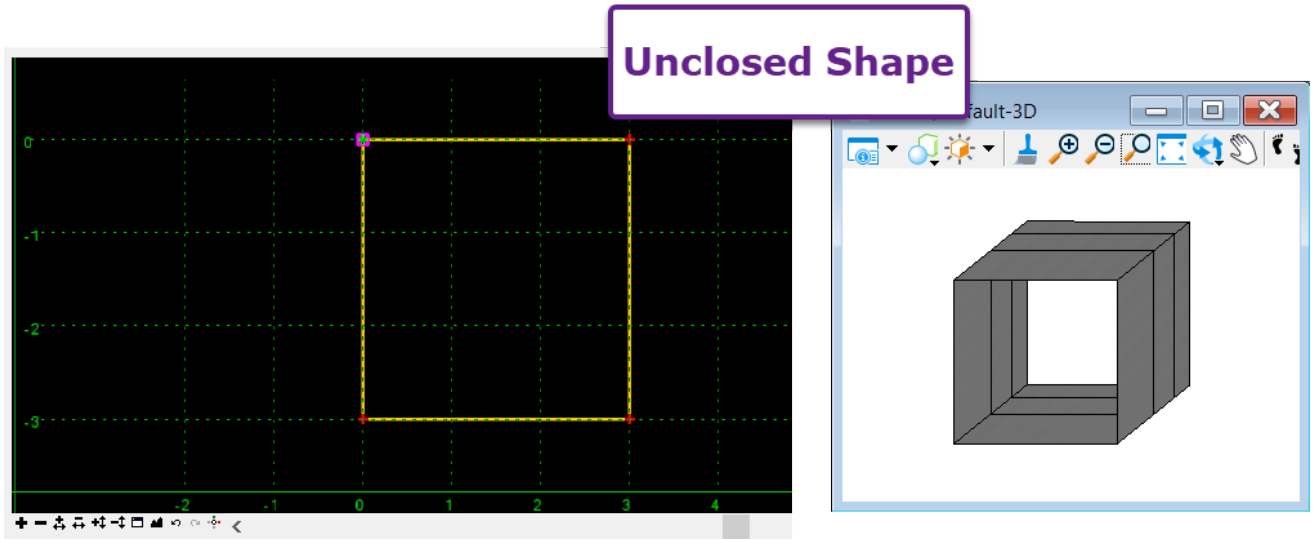
This option is only available for Components that appear *enclosed* in the *Template Editor*. In quantity calculations, a Closed Shape Component will produce a volume and a planar area. An Unclosed Shape component will only produce to the surface area. An enclosed shape could be used to represent hollow objects, such as a pipe-arch culvert.



If CHECKED, the component will be Volumetric when processed by the Corridor.

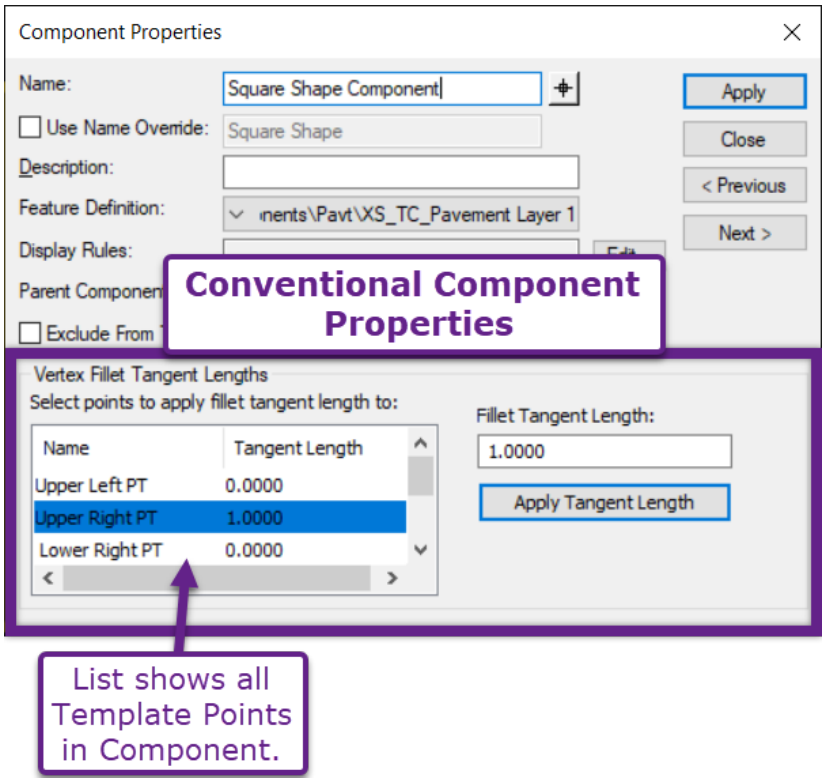


If UNCHECKED, the component will be Planar when processed by the Corridor.



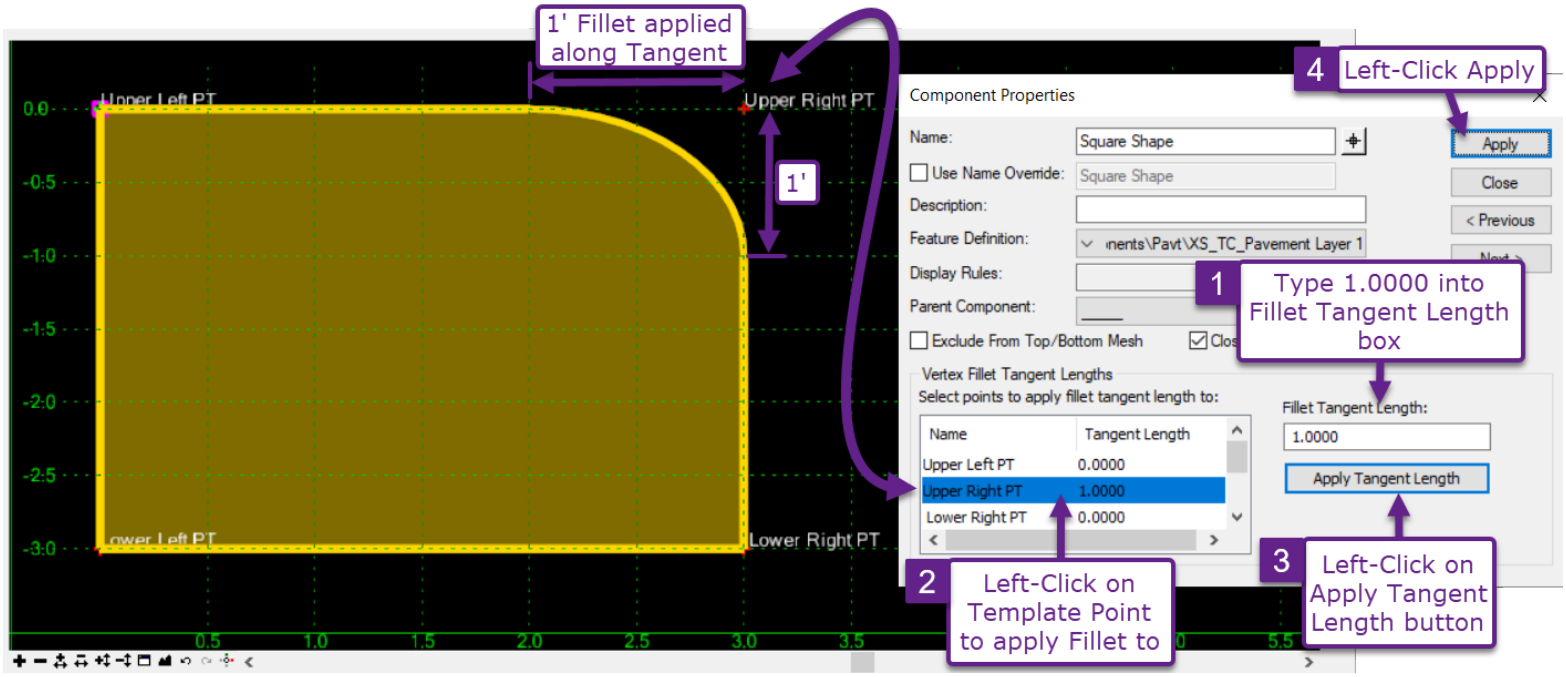
8D.6 Conventional Components Properties – Circularly Fillet a Component

Conventional Components have a unique property called **Vertex Fillet Tangent Lengths:**



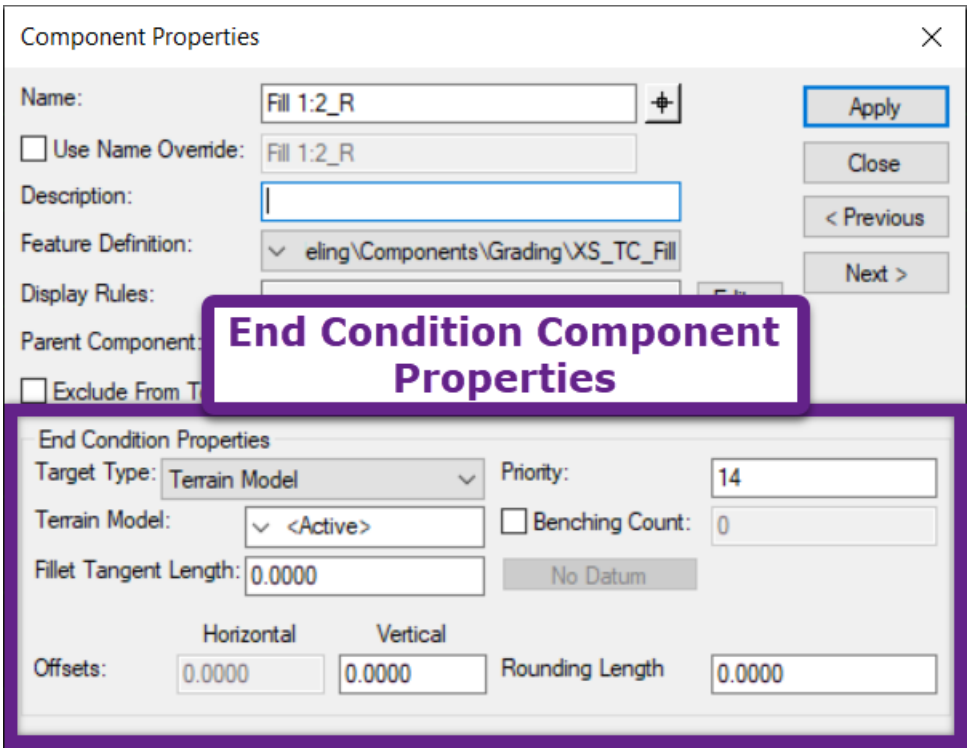
The sole function of this property is to place a circular fillet (round over) on the corner of a Conventional Components. The workflow shown below demonstrates how to place a 1’ on the corner of a Conventional Component:

WARNING: Placing Fillets on a Component may increase corridor processing time.



8D.7 End Condition Component Properties

The properties shown below are unique to End Condition Components:



Target Type: Specifies the Target Type which the End Condition will search for. See the next page for more information on Target Types.

Fillet Tangent Length: This property is only consequential if a Linear Element is chosen as the Target Type. The angle between the End Condition Component segment and Linear Element Target will be rounded according to value entered in this box.

Offsets: If the User wishes for the End Component to intercept the Target above or below the actual Target elevation, then the Vertical Offset box can be utilized. An example use would be setting this Vertical Offset value to -0.5' to account for 6" of topsoil stripping which may occur before road reconstruction.

Horizontal Offsets are only available for *Feature Definition* and *Linear* Target Types. If this setting is used, the End Condition will intercept the ORD Element at a horizontal offset position specified by the User.

Priority: If multiple End Conditions Components are present in a Template, the various End Conditions will be solved and drawn according to Priority value. See *8C.7.e Multiple End Conditions and End Condition Priorities*.

Benching Count: If this box is CHECKED, benching settings can be enabled. See *8D.7.b End Condition Benching*.

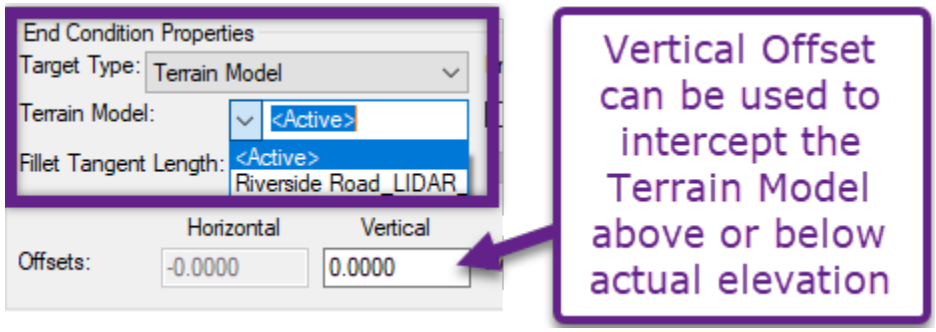
Rounding Length: The angle between the final End Condition Component line segment and Target will be rounded according to the length typed into this box.

WARNING: Rounding Lengths other than zero may increase corridor processing.

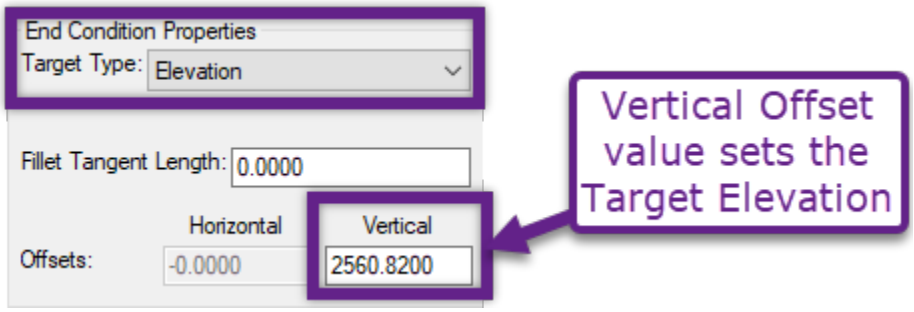
8D.7.a End Condition Target Types

Terrain Model: By default, the Target Type is set to <Active> Terrain Model. When set, the *active* Terrain Model will be intercepted by the End Conditions. Typically, the Existing Ground Terrain Model is set to *active*. However, through the "Terrain Model" drop-down, any Terrain Model within the current ORD File can be set for targeting.

TIP: The *Target Aliasing* tool can be used to target multiple Terrain Models with a single End Condition. For more information on Target Aliasing, see *9G.8 Target Aliasing*.



Elevation: If this Target Type is selected, the End Condition will Target the Elevation specified in the *Vertical Offset* box. An application of this Target Type would be grading a detention basin if the basin bottom elevation is already known.

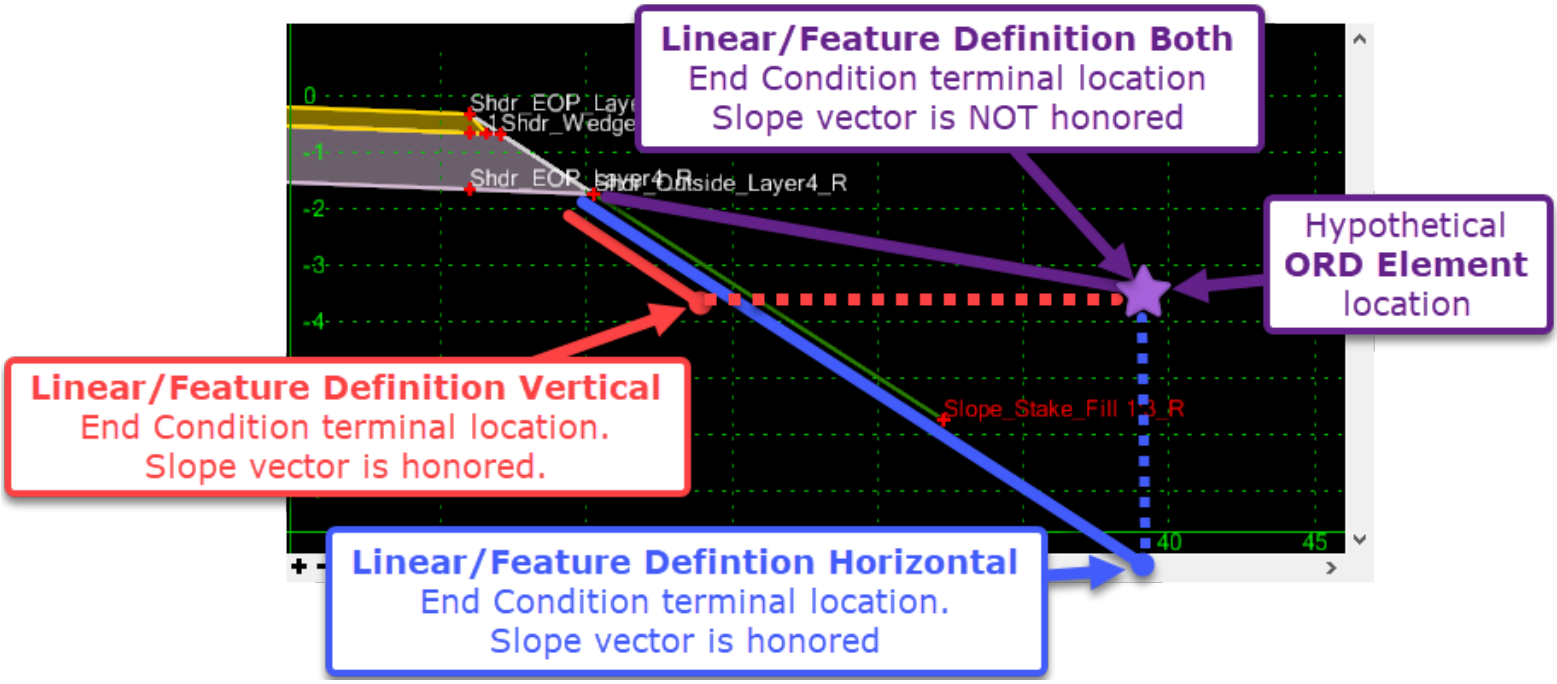


Linear Targets: If any of these Target Types are selected, the End Condition will Target one or more ORD Elements that are found in the *2D Design Model* . Linear Targets must be selected with the *Target Aliasing* tool for the Corridor to locate the Linear Elements to target. See *9G.8 Target Aliasing*.

If **Linear Horizontal** is selected, then the End Condition will terminate at the Horizontal Position of the ORD Element. The End Condition slope vector is honored. *(See graphic on next page)*

If **Linear Vertical** is selected, then the End Condition will terminate at the same Vertical datum as the ORD Element. The End Condition slope vector is honored.

If **Linear Both** is selected, then the End Condition will terminate at the Horizontal and Vertical location of the ORD Element. The End Condition vector is NOT honored. The End Condition will adjust the slope vector as necessary to intercept the ORD Element.

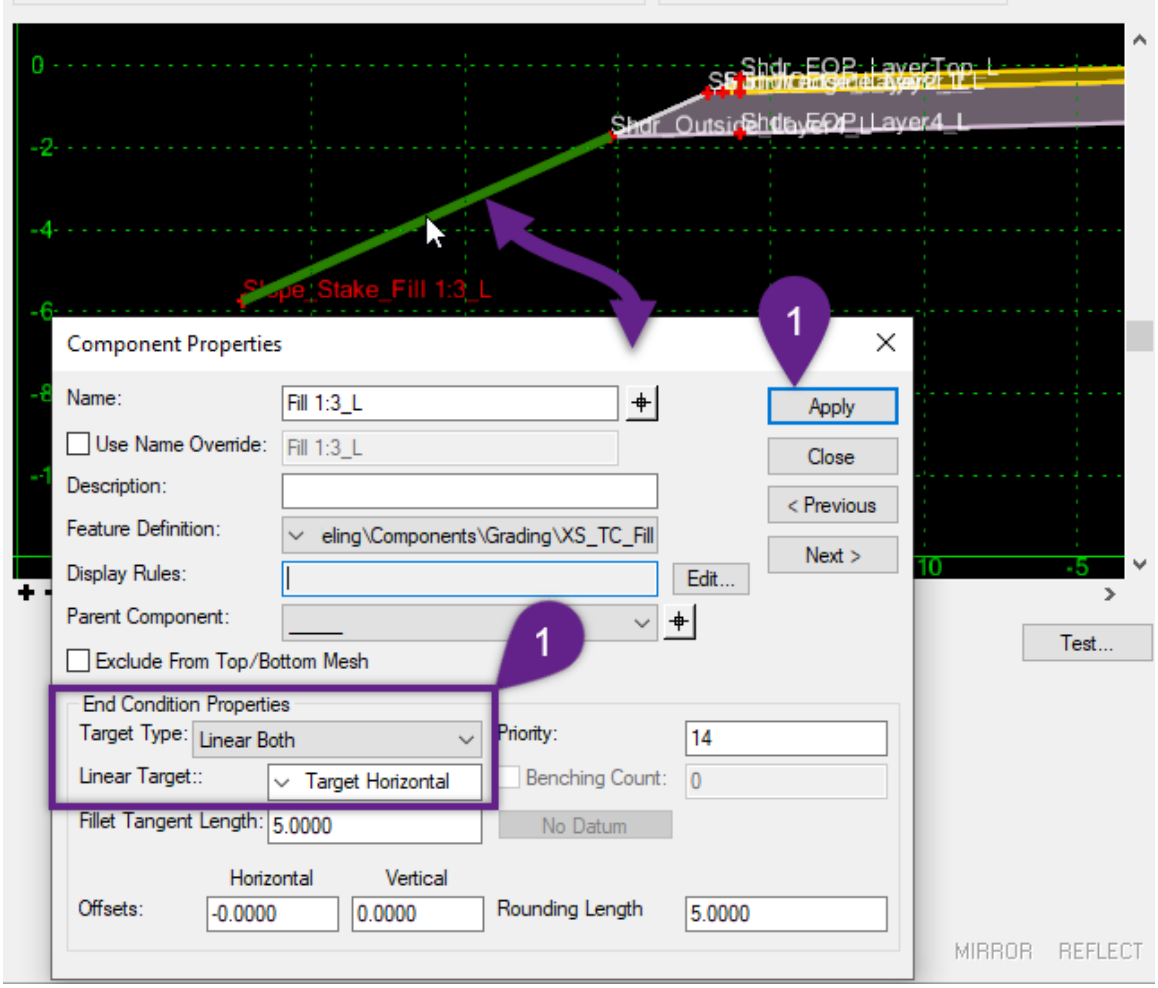


Feature Definition Targets: This setting operates similarly to Linear Targets to target ORD Elements. Instead of selecting individual Linear Elements to target, a Feature Definition is specified. All ORD Elements assigned to the set Feature Definition are targeted.

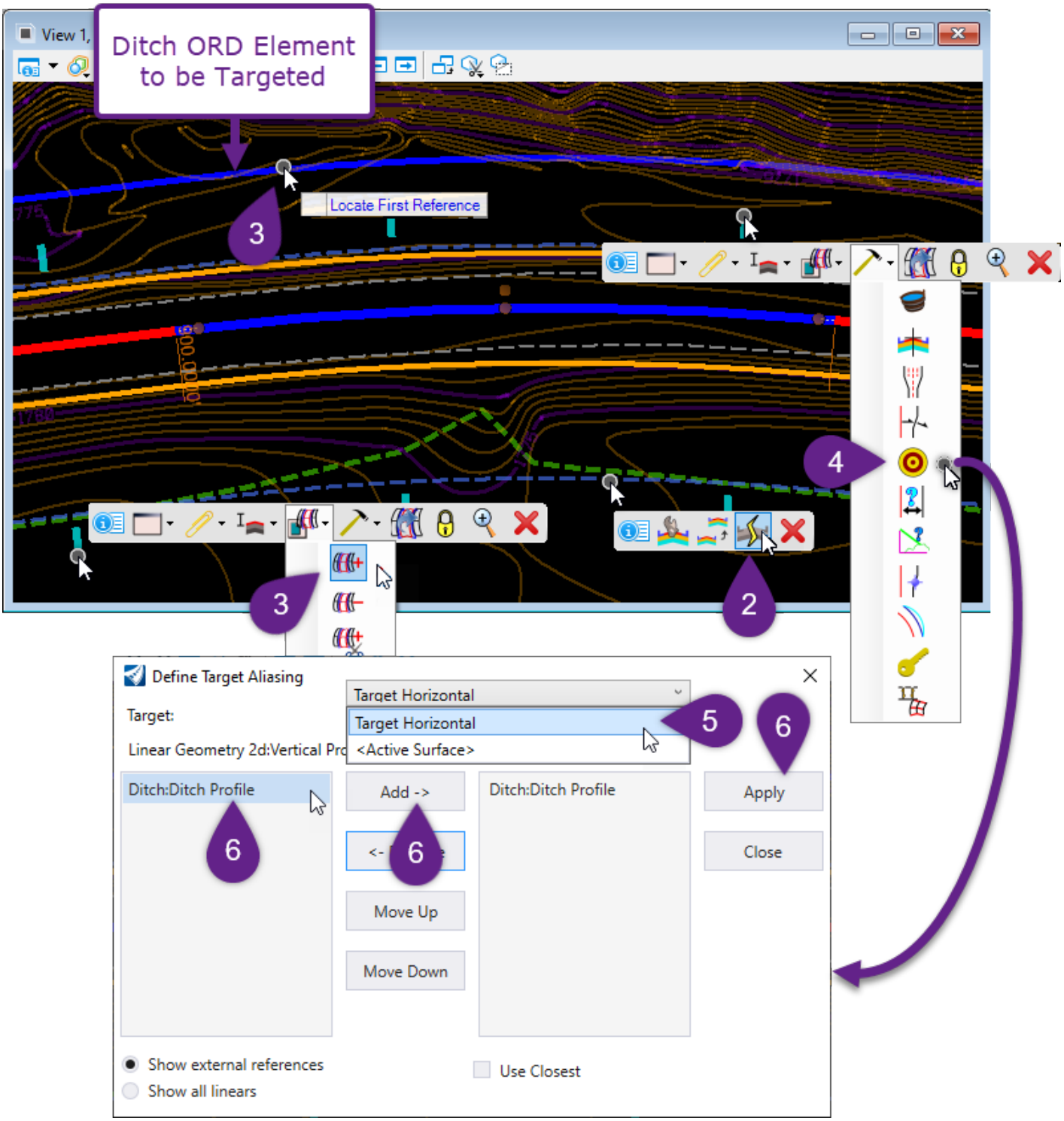
WARNING: Linear and Feature Definitions Targets must be added to the Corridor as *Corridor References* to function. See *9G.9 Corridor References*.

8D.7.a.i Target an ORD Element with End Conditions - Workflow

This workflow demonstrates how to Target an ORD Element with an End Conditions Component. This workflow is valid for both *Linear* and *Feature Definition* Target Types. However, if the *Feature Definition* Target Type is used, steps 3–5 are NOT necessary.



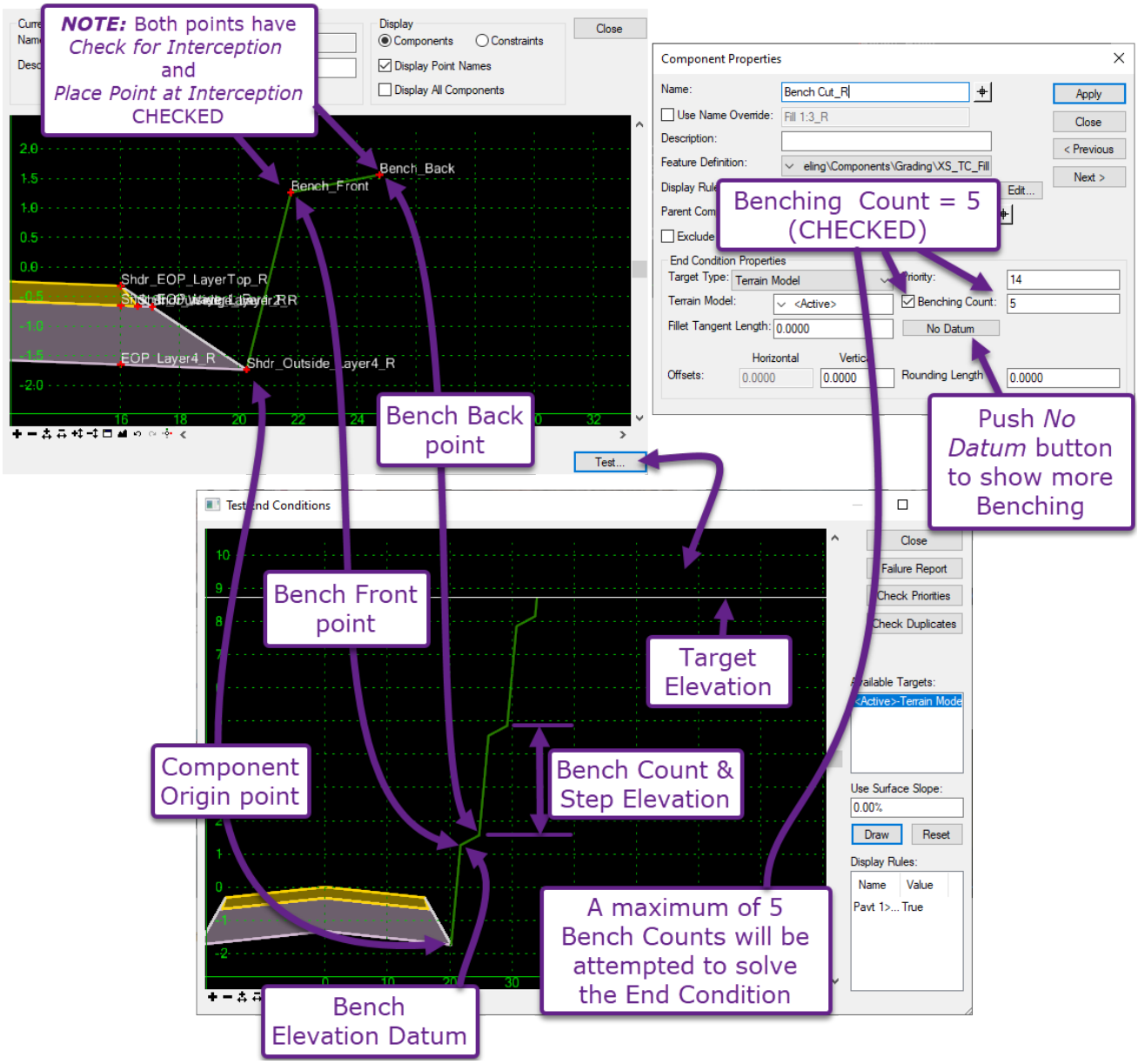
1	Open the Component Properties for the End Condition that will Target the Linear Element. Set the Target Type to *Linear Horizontal, Linear Vertical,* or *Linear Both.* Push the *Apply* button.
2	From the template drop Pop-Up Icon Menu, reapply the edited Template to the Corridor with the *Synchronize with Library* tool. See *9E.8 Synchronize with Library tool*.



3	From the corridor's Pop-Up Icon Menu, add the ORD Element(s) as a reference to the Corridor with the *Add Reference to Corridor* tool. See *9G.1 WARNING – Creating Circular References (Recursive Solutions)*.
4	From the corridor Pop-Up Icon Menu, open the *Define Target Aliasing* menu. See *9G.8 Target Aliasing*.
5	In the *Target* drop-down, select *Target Horizontal*.
6	In the left column, select the ORD Element to be targeted and push the *Add* -> button. When the ORD Element is shown in the right column, click the *Apply* button.

8D.7.b End Condition Benching

The Bench Count options in the End Condition Component Properties allows for Benching operations. In the ORD Software, Benching occurs if the End Condition Template Point does not find its Target within the range determined by its constraints. The End Condition Component will be repeated until the Target is found OR the maximum Bench Count iterations is reached – in which case, no solution is returned.



Elevation Datum: The FIRST *Bench Front point* will be placed at the value listed in the Elevation Datum box. Consecutive *Bench Front points* in the sequence will be vertically spaced according to the *Step Elevation*.

Point Datum: The FIRST *Bench Front point* will be placed at the same elevation as a specified Point in the Template. Consecutive *Bench Front points* in the sequence will be vertically spaced according to the *Step Elevation*.

Step Elevation: The Step Elevation is the vertical distance between the *Component Origin point* and the *Bench Back point.* In other words, Step Elevation is the height of an individual Benching Component. Step Elevation is only used with Elevation Datum and Point Datum options – after the initial datum is intercepted.

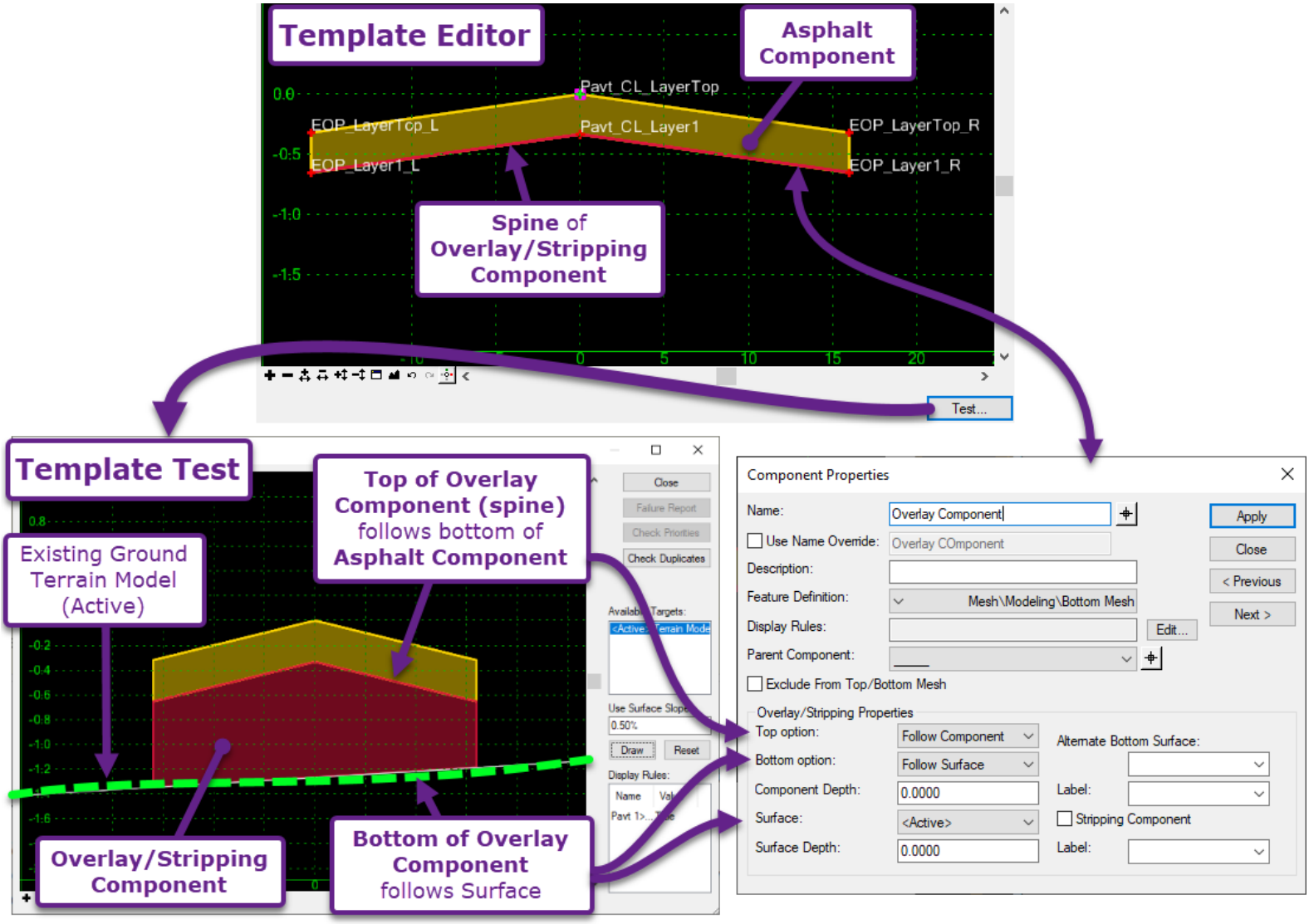
8D.8 Overlay/Stripping Components

Overlay/Stripping Components have the unique ability to be draped onto a Terrain Model. Overlay/Stripping Components can be thought of as Components with a variable depth. The depth is dictated by the Alignment Profile Elevation and the *active* Terrain Model.

In the Template Editor Overlay/Stripping Components always appears as a *Planar Component*, which is referred to as the *Spine*. However, when processed by a corridor, the Overlay/Stripping Component is draped on the terrain model and actually becomes a *Volumetric Component*. When this happens, the depth of the Overlay/Striping Component is measured from the *Spine* to the Terrain Model.

In the graphic below, the top of the Overlay/Stripping Component follows the bottom of the Asphalt Component. The bottom of the Overlay/Stripping Component follows the existing ground terrain model.

NOTE: Component elevations are relative to the Corridor Profile.



8D.8.a Overlay/Stripping Component Properties

Top Option: Specifies how the top of the Overlay/Stripping Component behaves.

Follow Surface: The Top of the Overlay/Stripping Component will follow the Surface (Terrain Model) specified in the drop-down.
Follow Component: The Top of the Overlay/Stripping will follow the bottom surface of the vertically adjacent Conventional Component. The spine shape will be maintained along the top of the Overlay/Stripping Component.
Follow Highest: The Top of the Overlay/Stripping Component will follow the a Terrain Model or the bottom surface of the vertically adjacent Component; whichever one is higher.

Bottom Option: Specifies how the bottom portion of the Overlay/Striping Component behaves.

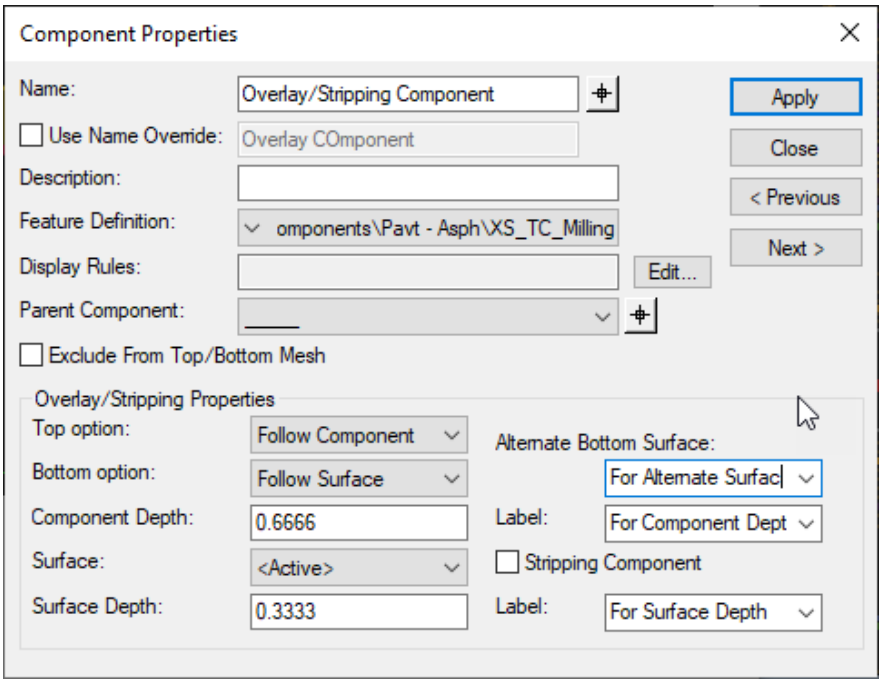
Follow Surface: The Bottom of the Overlay/Stripping Component will follow the Surface (Terrain Model) specified in the drop-down.
Follow Component: The Bottom of the Overlay/Stripping Component will follow the top surface of the vertically adjacent Component. The spine shape will be maintained along the bottom of the Overlay/Stripping Component.
Follow Lowest: The Bottom of the Overlay/Stripping Component will follow the Terrain Model or top surface of the adjacent component; whichever one is higher.

Component Depth and Label: This value cannot be negative. Component Depth refers to the depth of the Overlay/Stripping Component. By default, this is set to 0.000, which means the Top and Bottom Option will control the depth of the component. This parameter sets a minimum depth for the component.

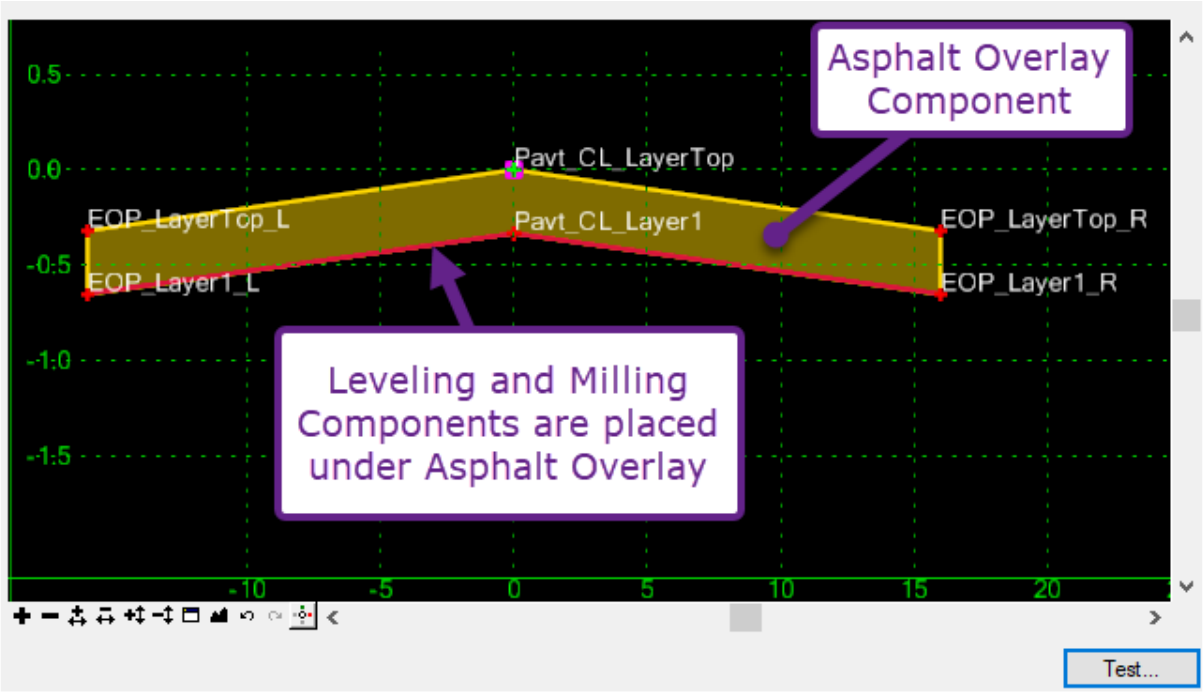
Surface Depth and Label: This value cannot be negative. The Overlay/Stripping Component will follow be offset from the surface at a defined Depth.

Alternate Bottom Surface: By populating this field, an Alternate Surface can be created for the BOTTOM of the Overlay/Stripping Component. The Alternate Surface is not created until the *Create Corridor Alternate Surface* tool is used. See *8C.4 Alternate Surface*.

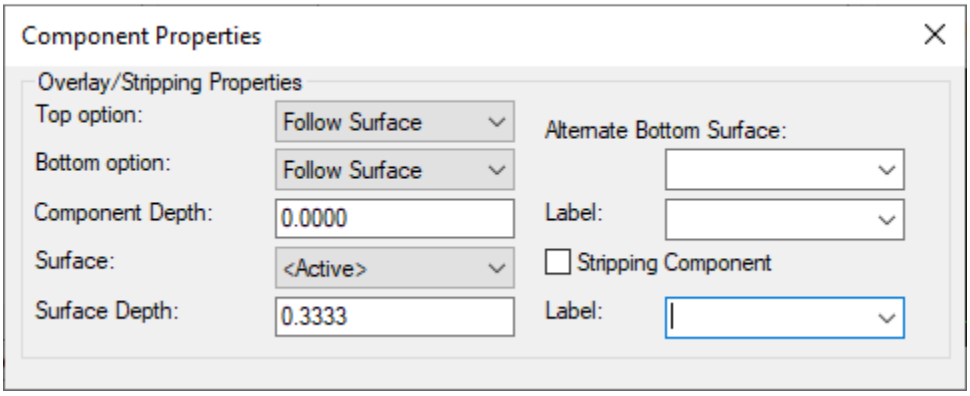
Stripping Component: If this box is checked, then the Overlay/Stripping Component is NOT included in Material and Earthwork calculations performed by the software.



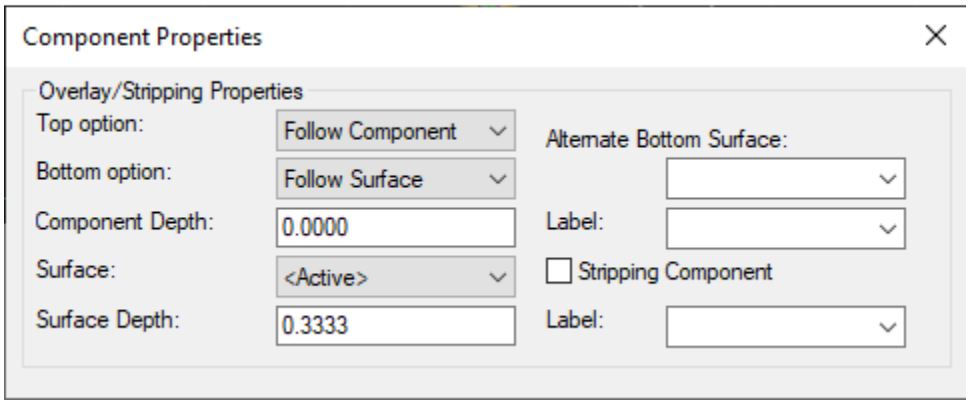
8D.8.b Milling and Leveling Component - Example Configurations

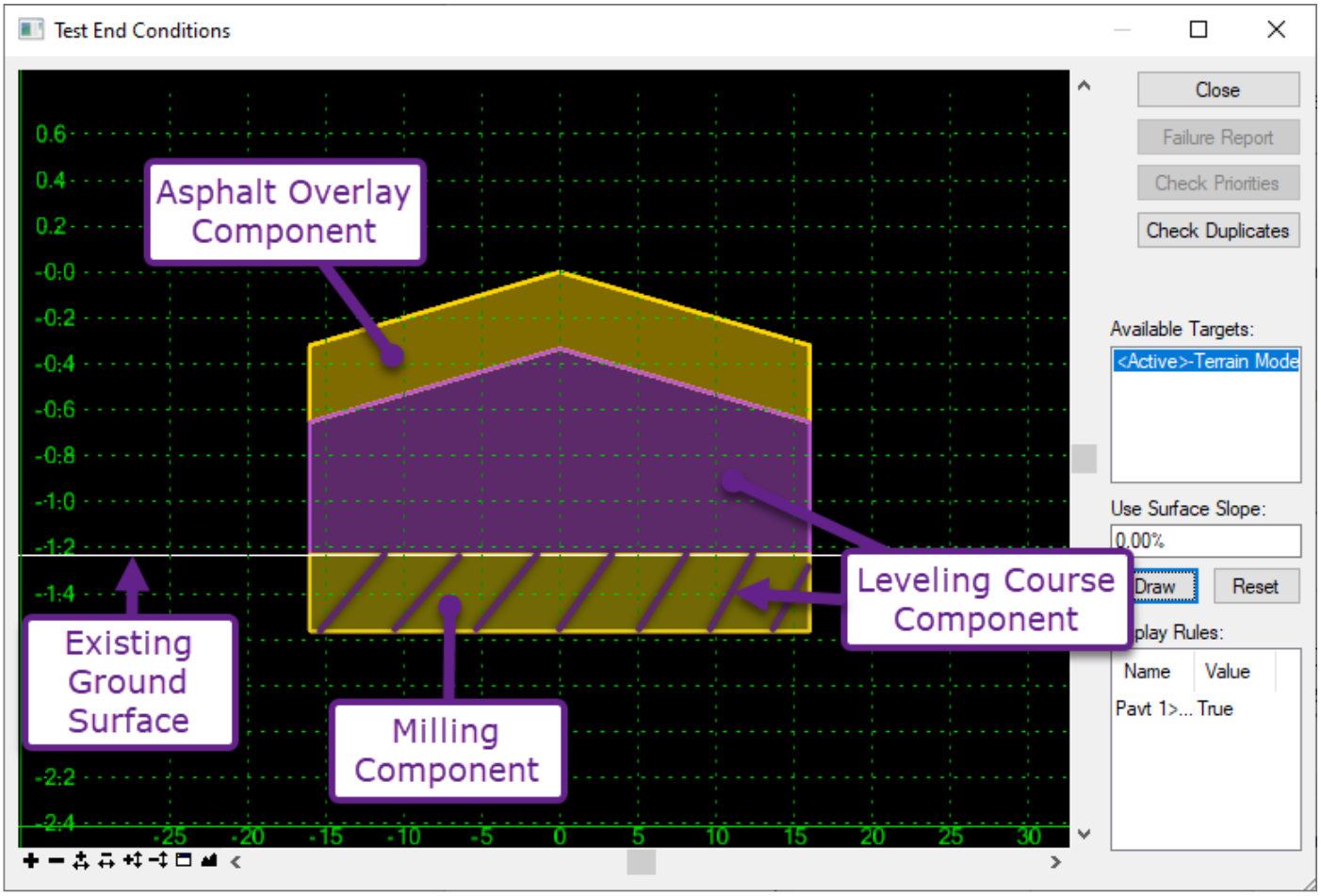


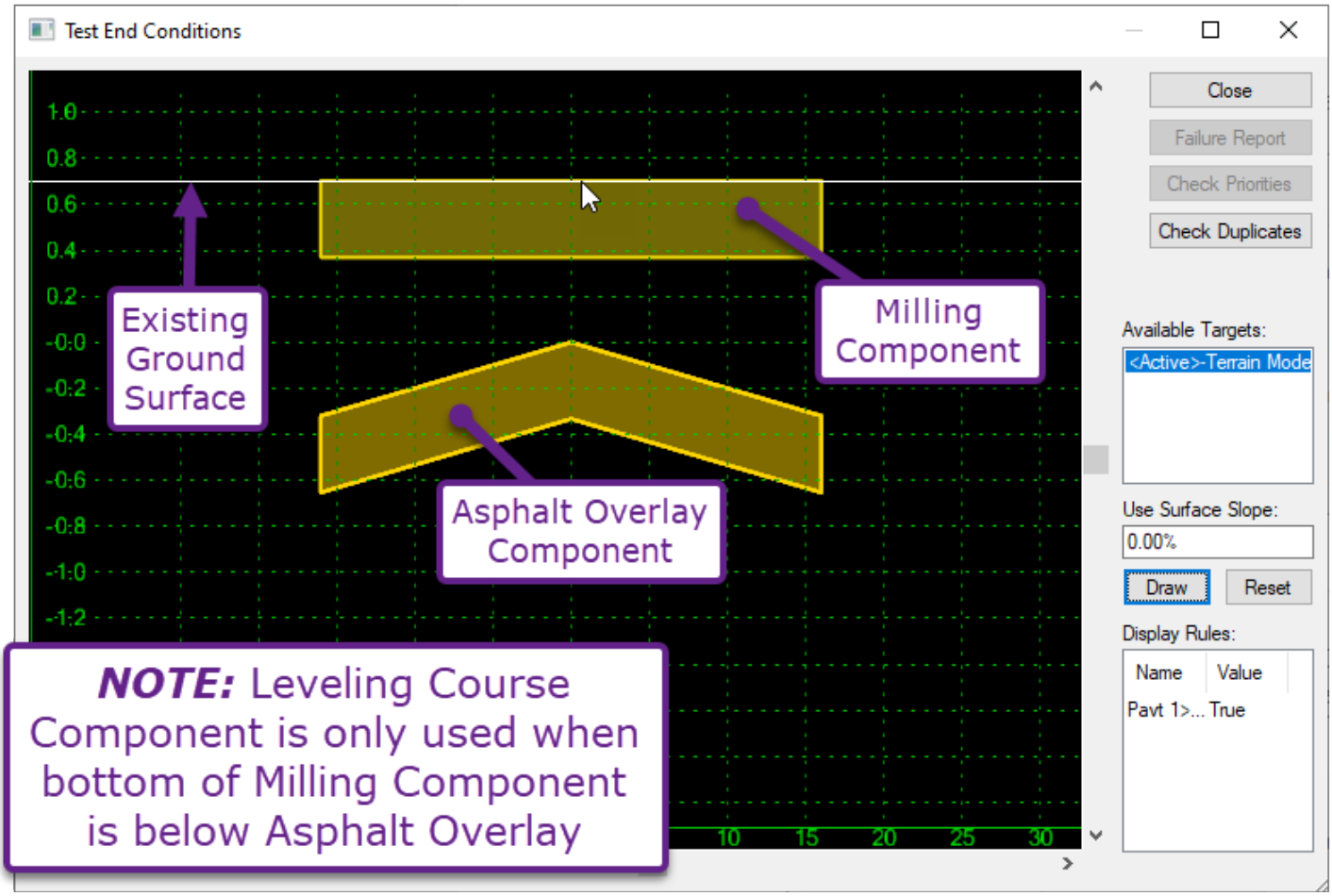
Milling Component: The Overlay/Stripping Component Properties show how to account for a 4” milling of existing asphalt.



Leveling Component: The Overlay/Stripping Component Properties show how to account for a leveling course. The Bottom of the Leveling Component will be at the bottom of the 4” mill. The Top will follow the bottom of the Asphalt Overlay Component. The Leveling Component is only used if the bottom of the 4” mill falls below the bottom of the Asphalt Overlay Component.



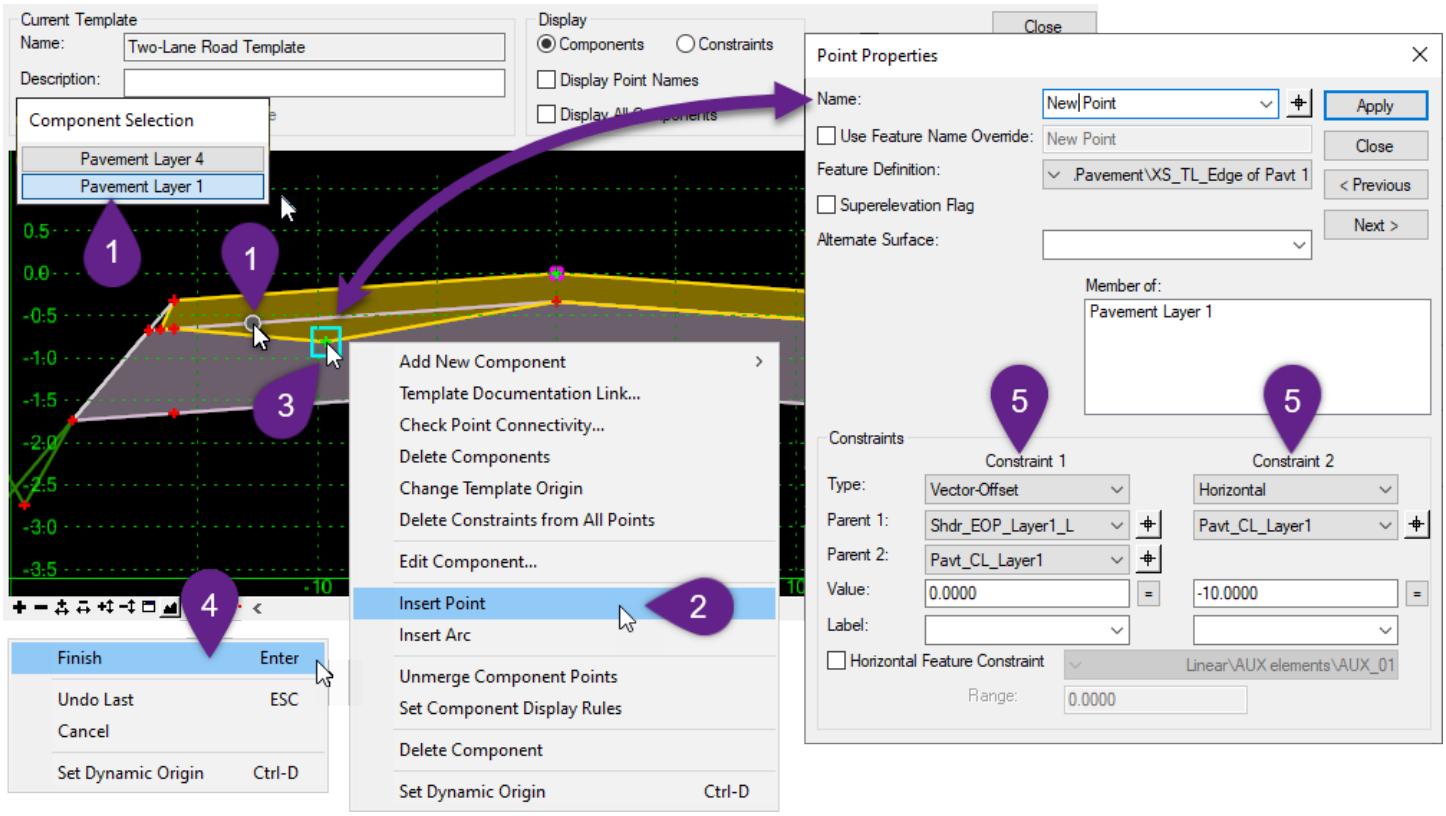




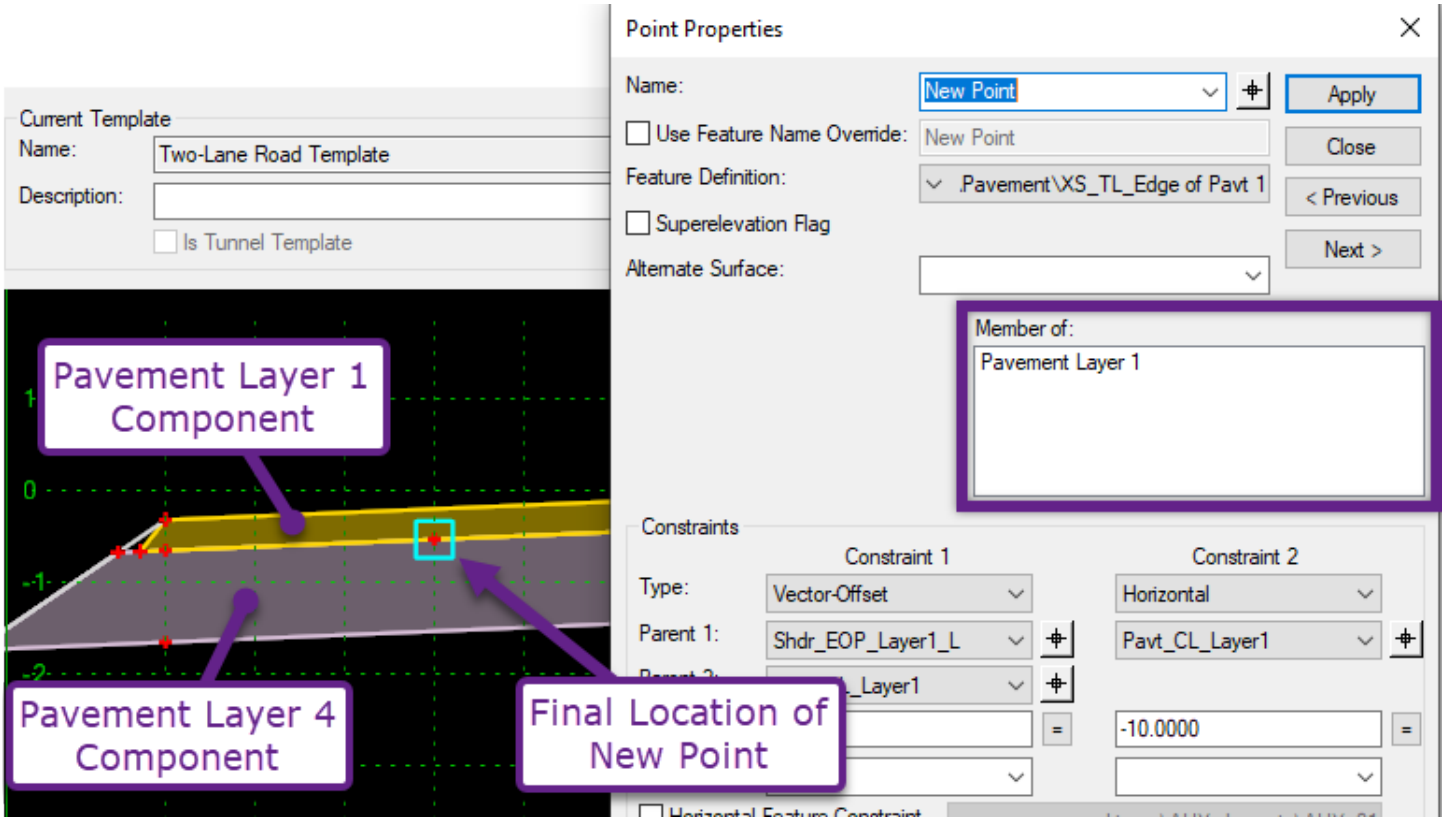
NOTE: Leveling Course Component is only used when bottom of Milling Component is below Asphalt Overlay

8E – CREATING AND MANIPULATING POINTS AND COMPONENTS

8E.1 Insert a Point into a Template Component

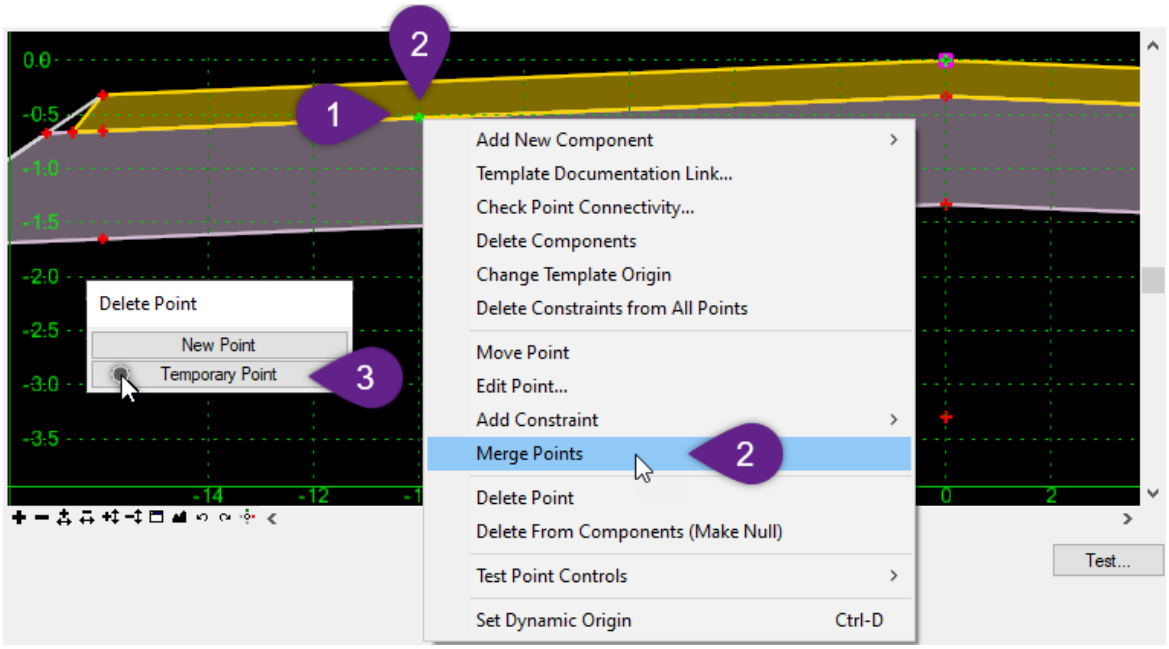


1	Right-Click on the Template Component to which the New Point is to be added – in the approximate desired location for the New Point. If there are adjacent Template Components, the Component Selection window will be shown. In this example, the desire is to add the Point to Pavement Layer 1.
2	Select the *Insert Point* tool from the Right-Click menu.
3	In the Active Template Editor Screen, Left-Click at the approximate location for the new Point.
4	Additional Points can be added by Left-Clicking in the desired location. If no more Points are to be added, Right-Click in the Template Editor and select *Finish*.
5	Newly-added Points will be *Unconstrained*. Enter the new Point Properties and assign Constraints.


Current Template
Name:
Two-Lane Road Template
Description:
Is Tunnel Template
Pavement Layer 1 Component
Pavement Layer 4 Component
Final Location of New Point
Point Properties
Name:
New Point
Use Feature Name Override:
New Point
Feature Definition:
.Pavement\XS_TL_Edge of Pavt 1
Superelevation Flag
Alternate Surface:
Apply
Close
< Previous
Next >
Member of:
Pavement Layer 1
Constraints
Constraint 1
Constraint 2
Type:
Vector-Offset
Horizontal
Parent 1:
Shdr_EOP_Layer1_L
Pavt_CL_Layer1
-10.0000

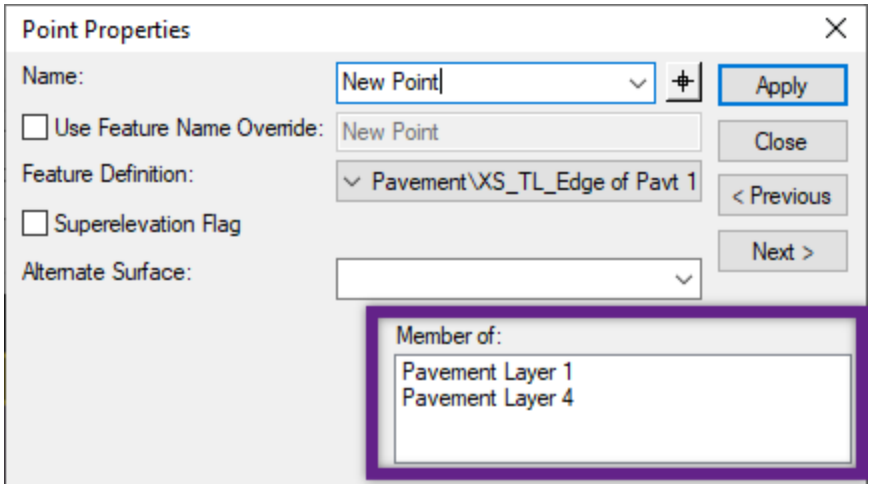
8E.2 Merge Template Points

In the *Add a Point to a Template Component* workflow (shown on the previous page), the final location of the New Point is on the shared border of Pavement Layer 1 and Pavement Layer 4 Components. However, the New Point is only *Member of* Pavement Layer 1 Component – as shown in the Point Properties. In order to make the New Point a *Member of* both components, a temporary point has to be added at the same location and then the two points can be *Merged*. The final Merged Point will be *Member of* both Components.



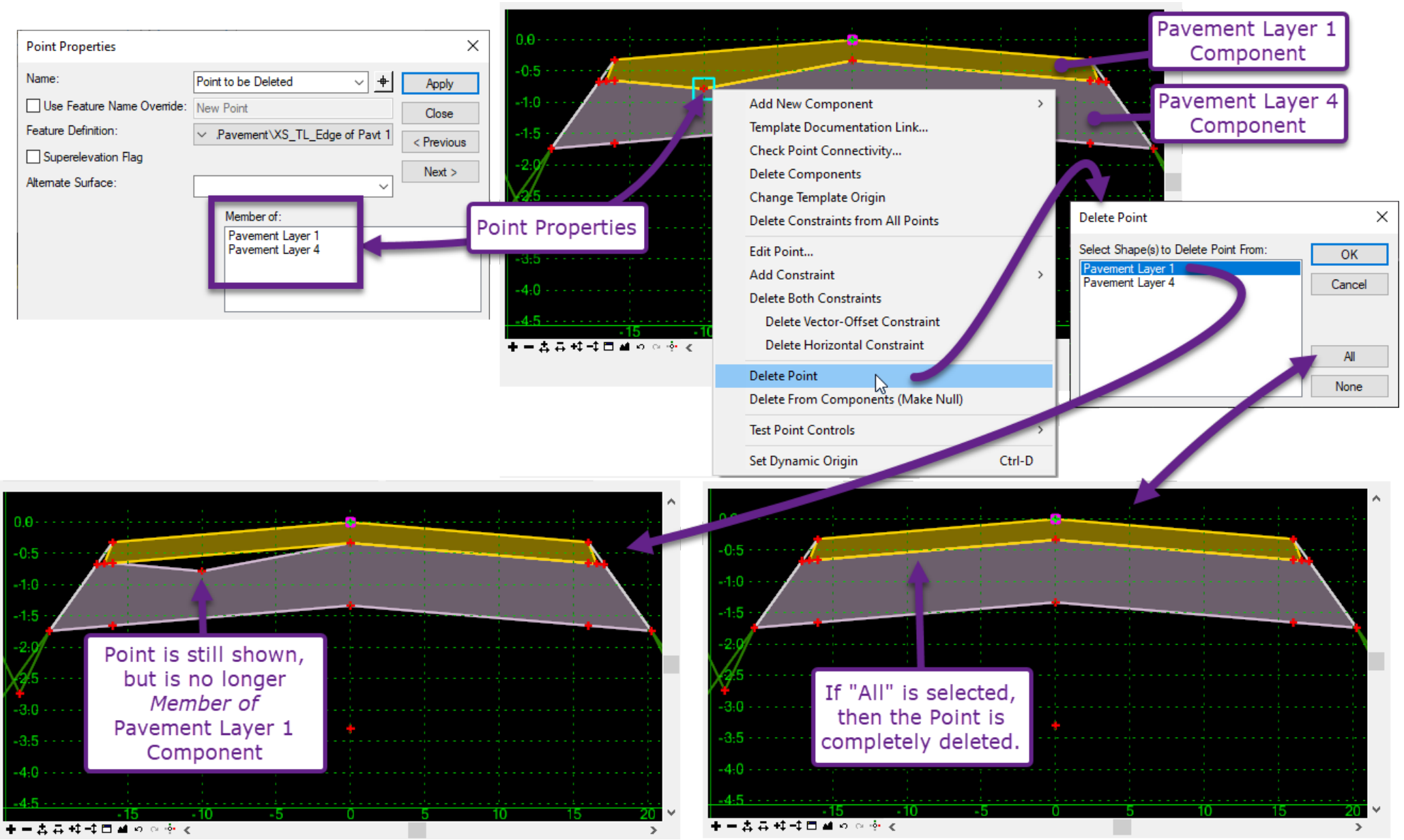
1	Following the procedures shown in *Add a Point to a Template Component*, add a Temporary Point at the same location as the New Point. The Temporary Point is inserted to the Pavement Layer 4 Component.
2	Right-Click on the overlapping Points and select *Merge Points*
3	The *Delete Point* box will appear. Whichever point is selected in this dialog will be deleted. The other point(s) will remain. For this example, the Temporary Point is selected for deletion.

After this procedure is completed, the New Point is *Member of* both Components – as shown in the Point Properties of the New Point:



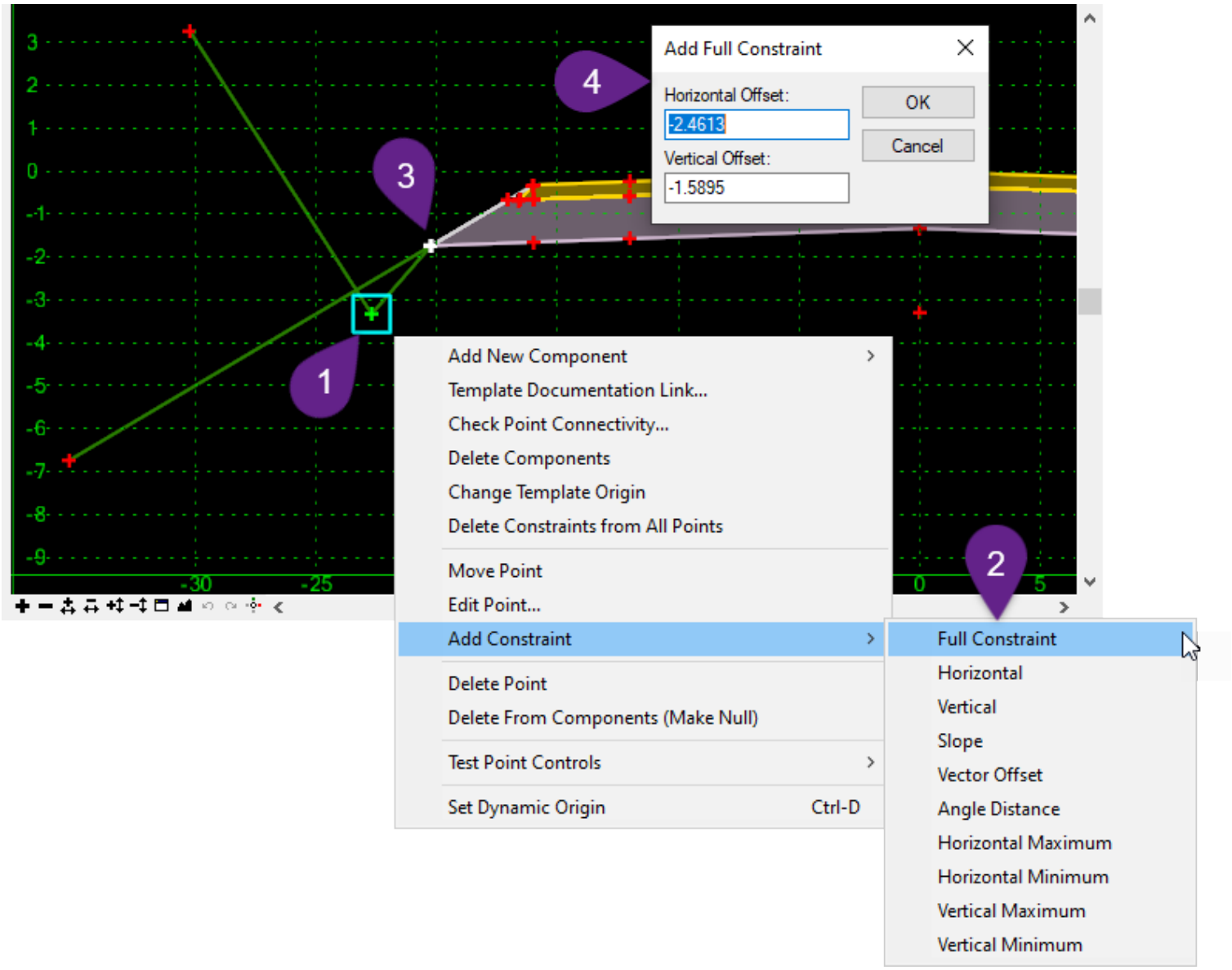
8E.3 Delete Template Point

Template Points are deleted by simply Right-Clicking on the Point and selecting "Delete Point". If the Template Point to be deleted is *Member of* multiple Components, the User is given the option to delete the point from a single Component or from *All* Components.



8E.4 Assign Constraints from the Right-Click Menu

Template Point Constraints can be conveniently assigned through the Right-Click Menu.



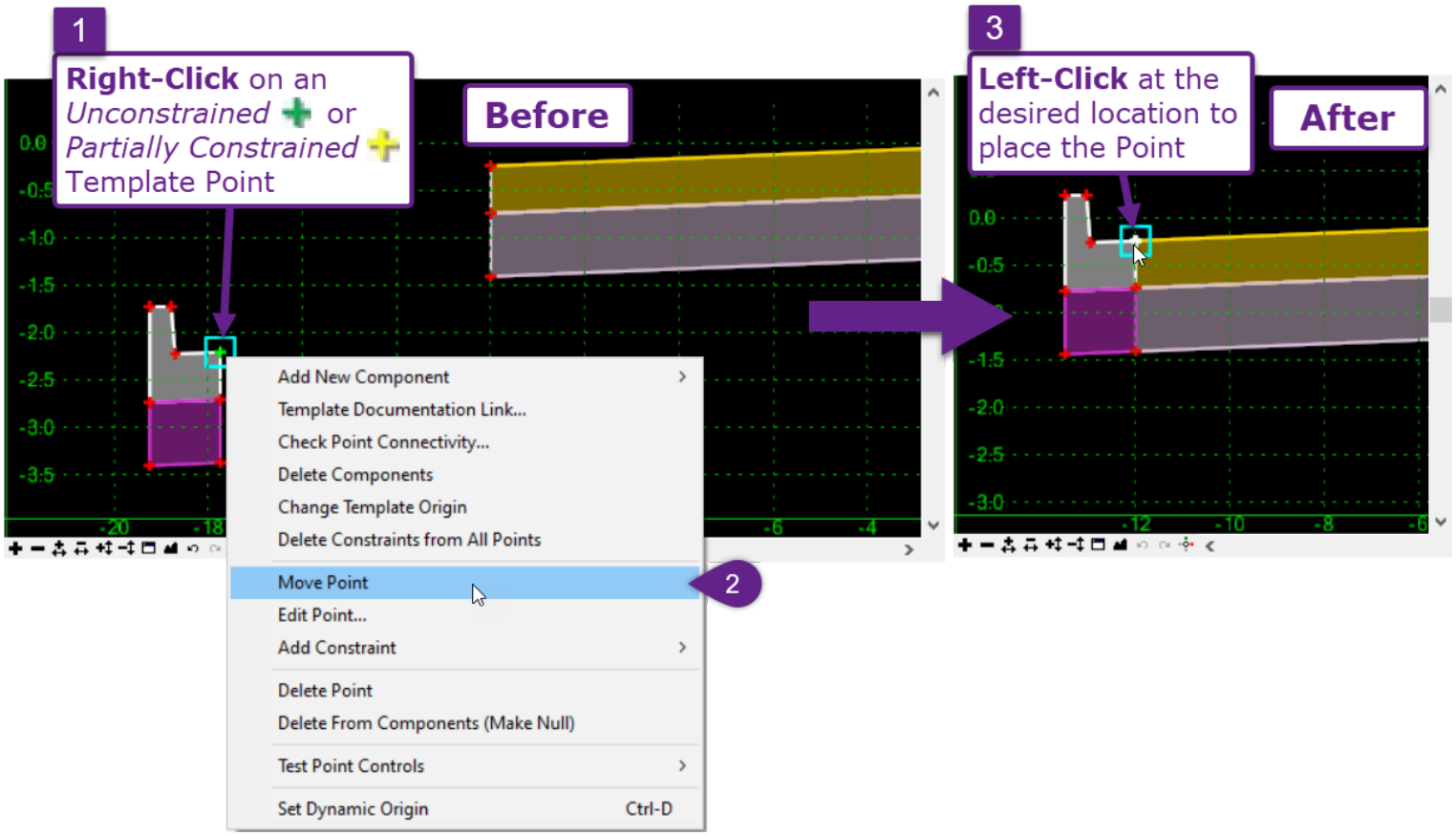
1	Right-Click on the Template Point to assign Constraints to it. In this case, the Ditch Point will be assigned Constraints.
2	In the Right-Click Menu, navigate to the *Add Constraint* and select the desired Constraint type. ***NOTE:*** The *Full Constraint* type will assign both a Horizontal and Vertical Constraint to the Point. All other types will only add the individual Constraint – potentially leaving the Point *Partially Constrained.*
3	In the Active Template Editor Screen, select the *Parent Point* for the constraint. In this case, the Road Hinge Point is selected.
4	In the *Add Full Constraint* dialog box, key-in the desired Constraint Values and Left-Click on *OK.*

8E.5 Move Template Points Graphically

If a Template Point is *Unconstrained* or *Partially Constrained*, then the Point can be moved around in the Active Template Editor Screen with the mouse cursor. *Unconstrained* points can be moved anywhere in the Active Template Editor screen. *Partially constrained* points are moved along the fixed direction/path that honors the constraint.

To access the *Move Point* tool, simply Right-Click on an *Unconstrained* or *Partially Constrained* Point and select *Move Point* from the Right-Click Menu. The *Move Point* tool will NOT be displayed in the Right-Click menu of a *Constrained* Point.

Use the mouse cursor to move the Template Point around. When satisfied with the location in the Active Template Editor Screen, Left-Click to place Point.

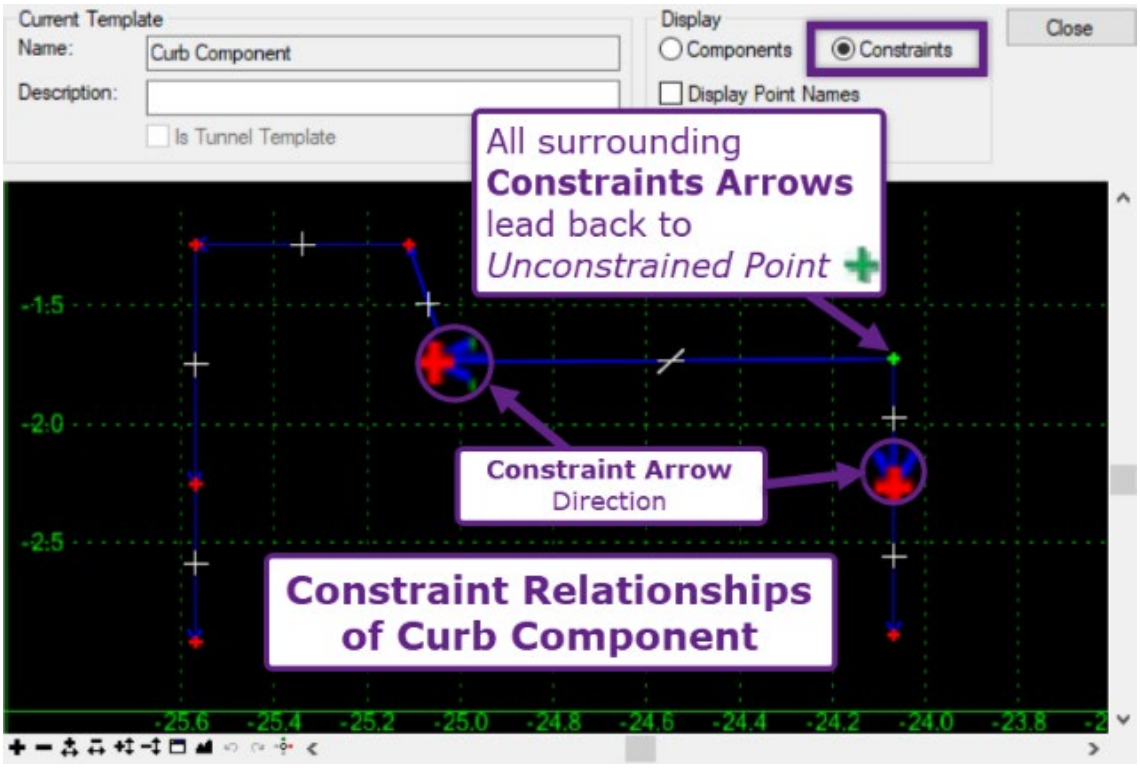


TIP: Enable *Step Options* to move Point at round grid intervals and assist in snapping to other Points. See *8B.4.a.ii Step Options*.

Before Moving a Point, the Constraints of the other surrounding Template Points should be understood to avoid unintended results.

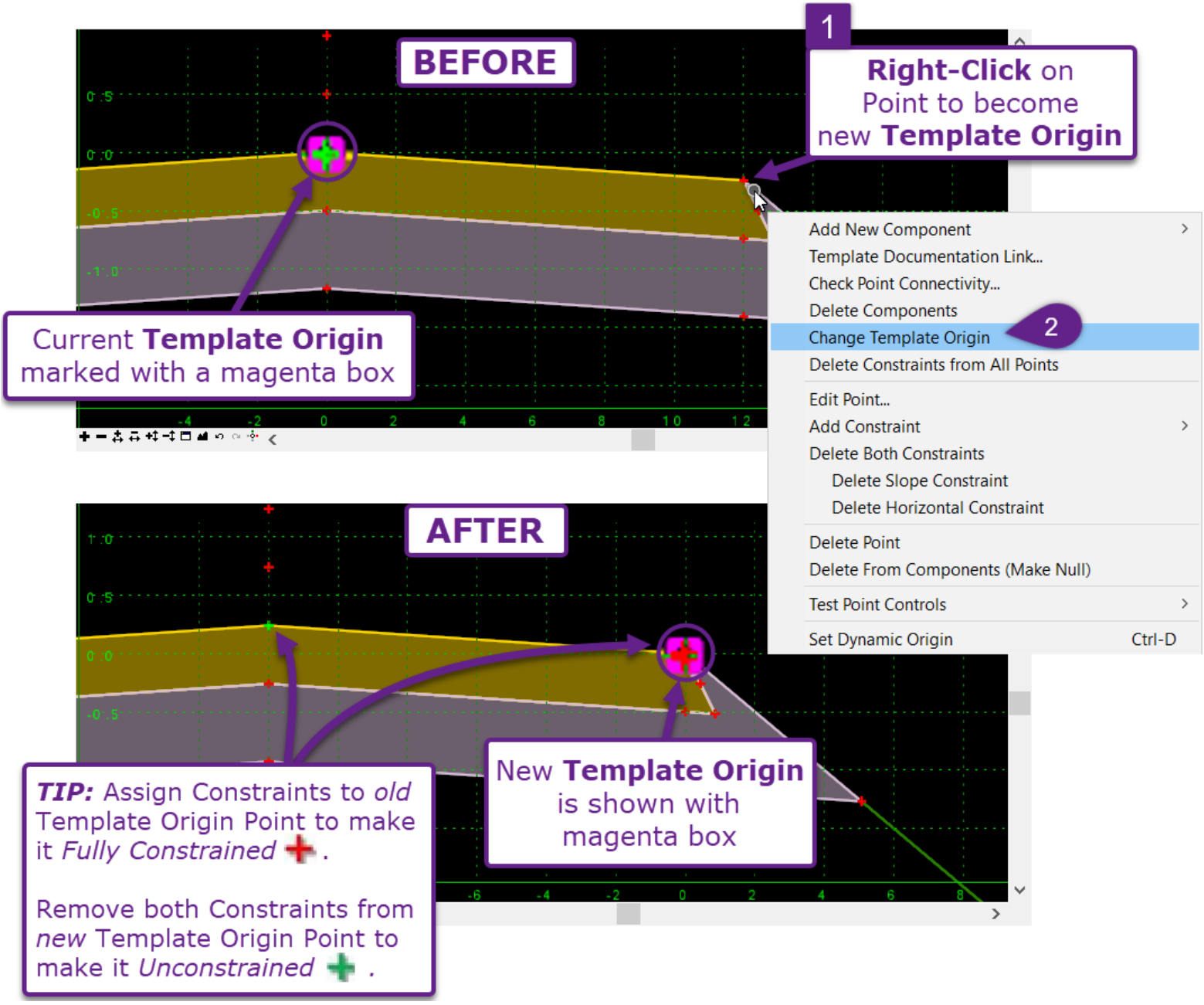
In the curb component shown on the previous page, all surrounding Template Points have constraints leading back to the Moved Point. When the Point is moved, all surrounding Template Points will follow to maintain Constraint relationships.

TIP: In the Active Template Editor Screen, turn on the Constraint Display to better understand Constraint relationships within the Template. See *8C.6.b Constraint Display in Active Template Editor Screen*.



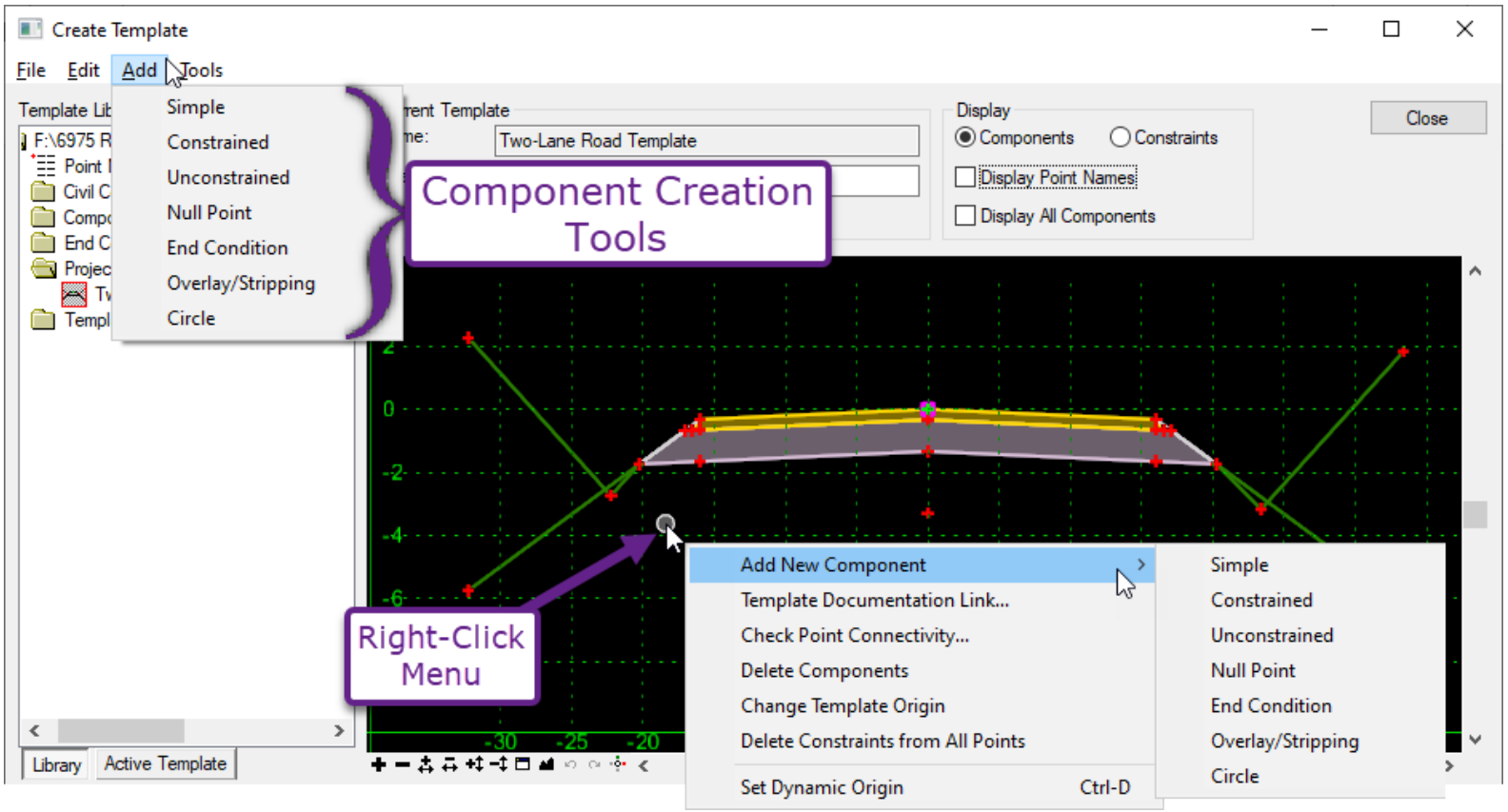
8E.6 Change Template Origin Point

To change the Template Origin Point for a Template, simply Right-Click on a Point and select *Change Template Origin*. After the *Change Template Origin* tool is used, the Template Grid will rearrange to set the (0,0) coordinate to the new Template Origin.



8E.7 Create New Template Components

New Template Components creation tools are accessed by Right-Clicking in the Active Template Editor Screen or through the Add drop-down located in the upper-left corner. Both locations contain identical tools.



With the exception of the *Simple* Component tool, New Components are created by placing new Points in order to define the shape of the New Component. If a New Point is placed atop an existing Point, the points will be *merged*.

Simple: This tool creates a rectangular component. See *8E.9 Create a Simple Component - Workflow*.

Constrained: Creates a Closed or Open Component by placing new Template Points. The Point placed first will be Unconstrained. All successive Points will be horizontally and vertically constrained to the preceding Point.

Unconstrained: Creates a Closed or Open Component by placing new Template Points. All Points will be Unconstrained, except for Points placed atop of existing points.

Null Point: Creates a single Null Point.

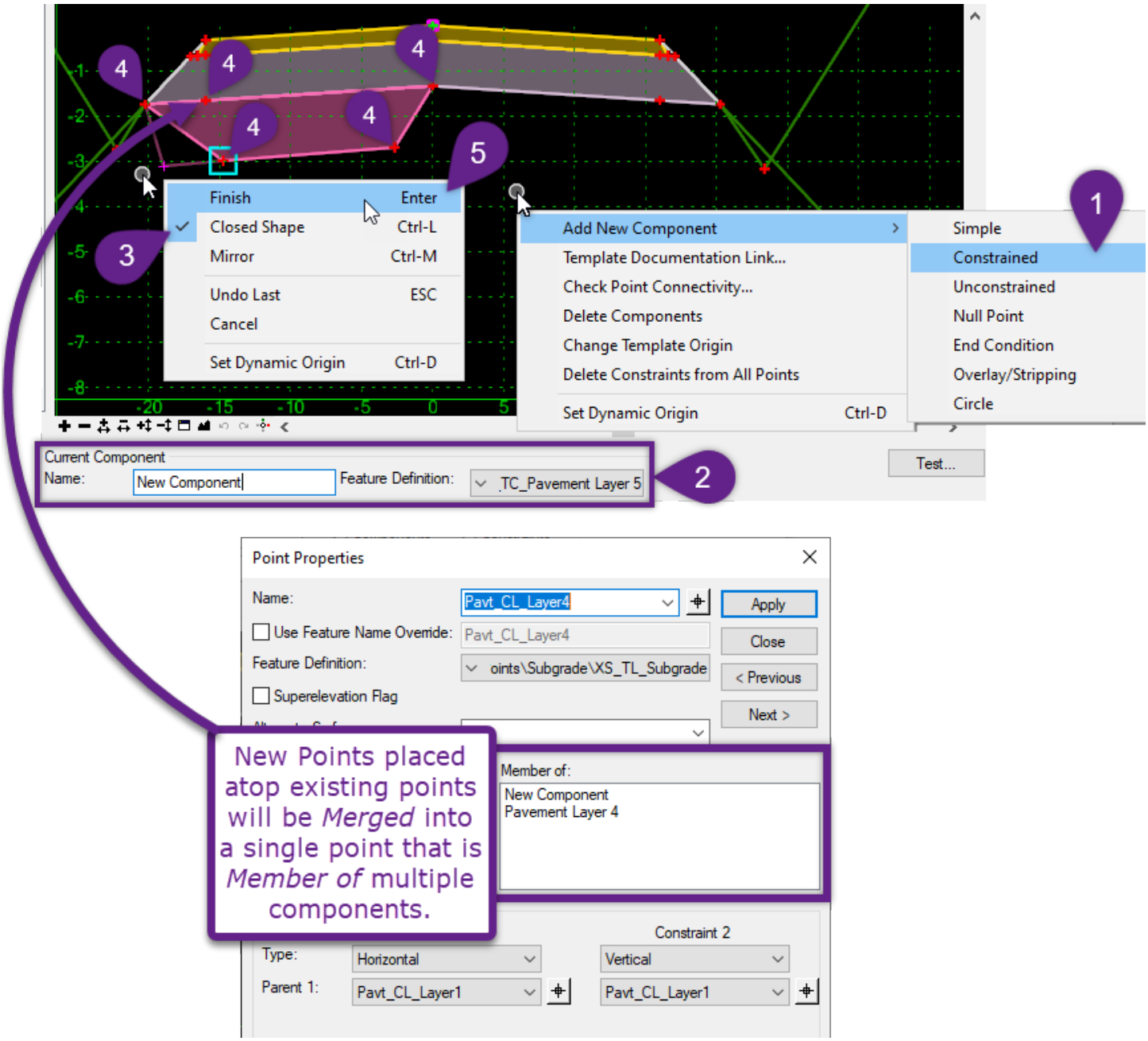
End Condition: Creates a single End Condition Component by placing new Template Points.

Overlay/Stripping – Creates an Overlay/Stripping Component by locating Template Points that comprise the spine of the Component.

Circle: Creates a Circular Component by locating the radius point and entering a Radius value.

8E.8 Create a New Component - Workflow

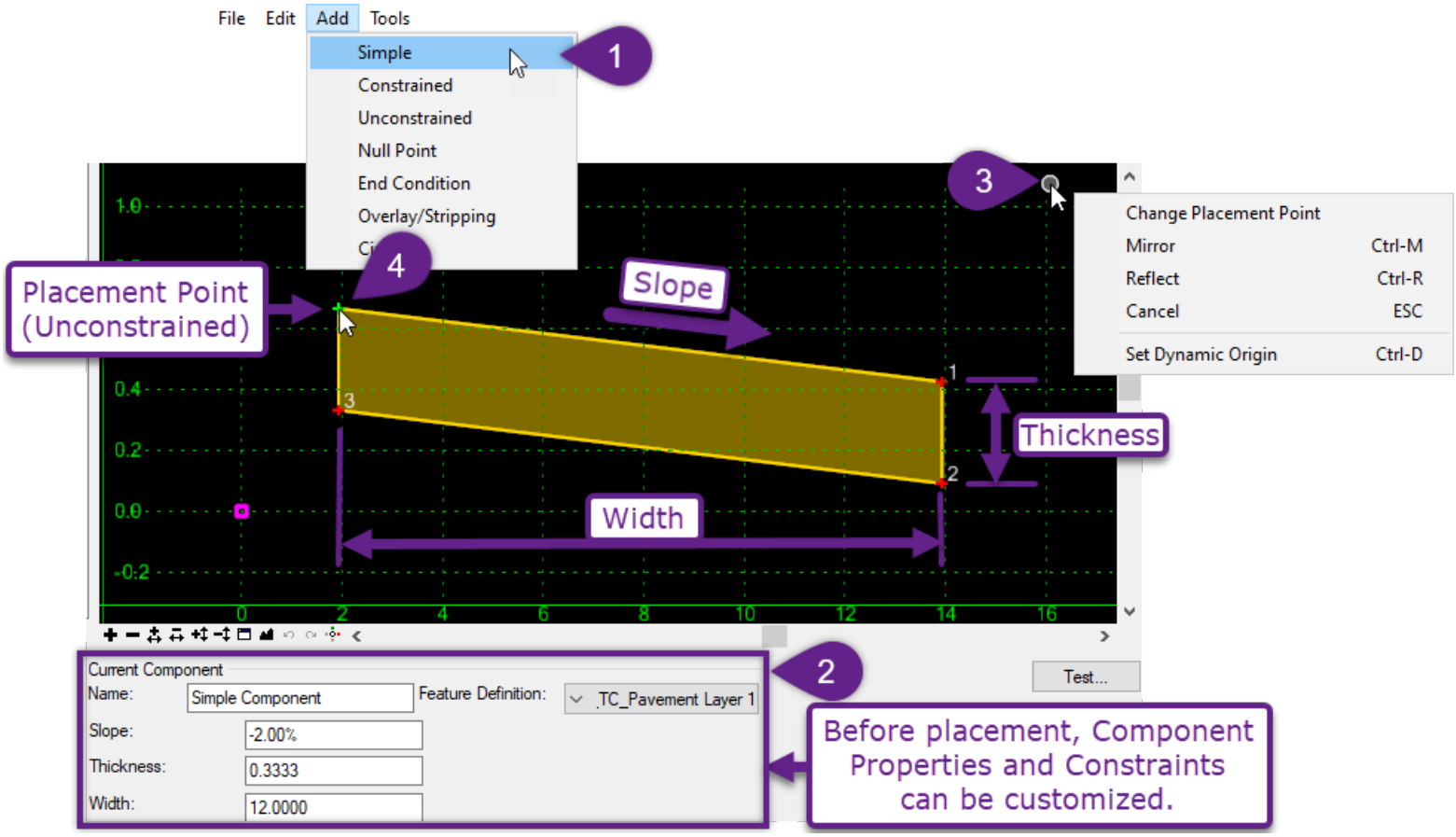
This workflow is applicable to all *Add New Component* tools except for the *Simple Component* tool.



1	Right-Click in the Active Template Editor Screen and select *Add New Component → Constrained.*
2	Assign the New Component an appropriate Name and Feature Definition.
3	Right-Click in the Active Template Editor Screen and identify whether the New Component should have a Closed Shape or Open Shape.
4	In the Template Grid, Left-Click on the point locations that will determine the general Component shape.
5	When satisfied with the general shape of the Component, Right-Click in the screen and select *Finish*.

8E.9 Create a Simple Component - Workflow

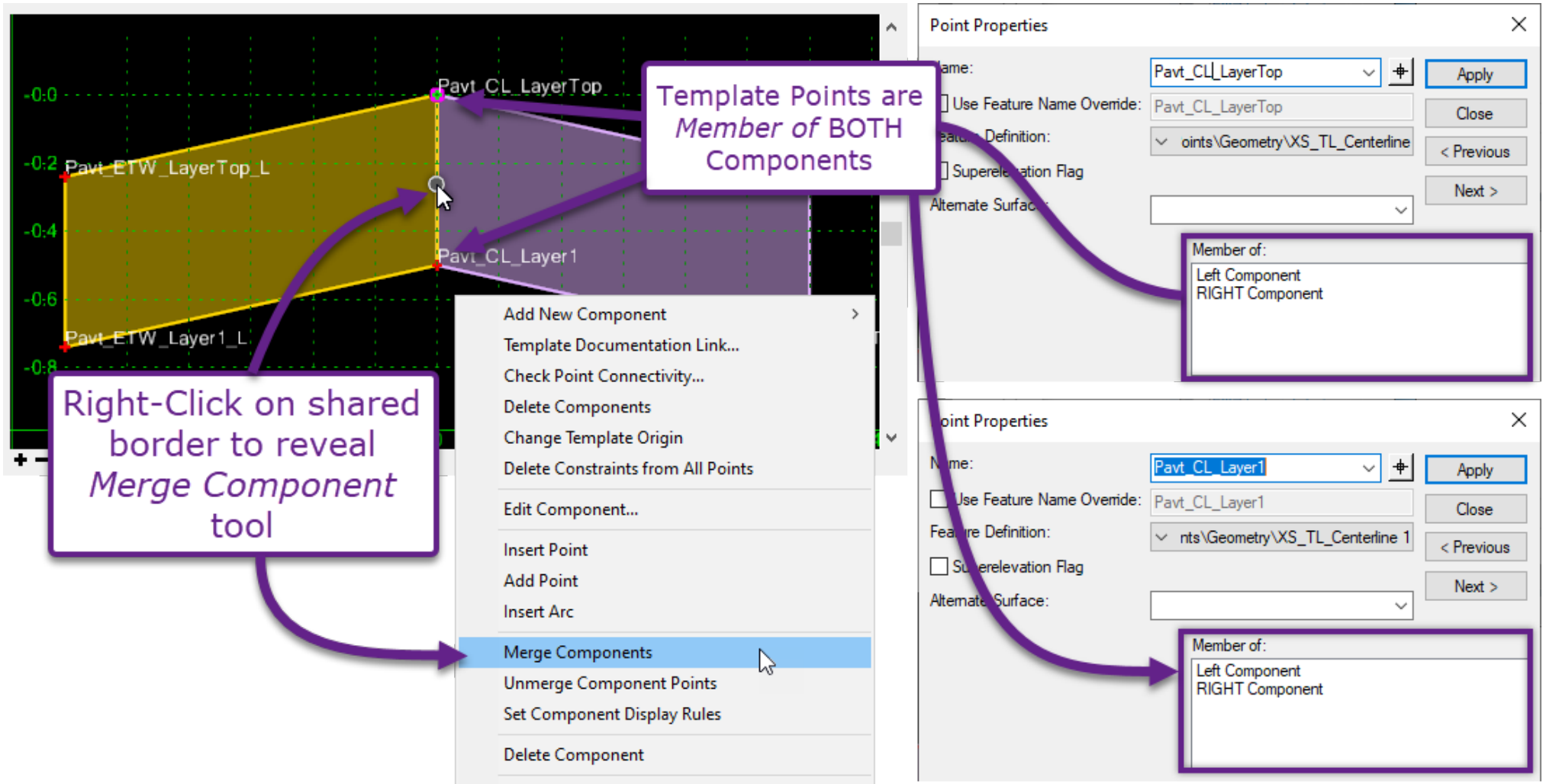
The workflow below shows how to create a *Simple Component.* Simple Components are rectangular in shape when first created.



1	From the *Add* drop-down in the upper-right corner of the Template Editor, select *Simple*.
2	Assign the New Component an appropriate Name and Feature Definition.
3	Right-Click in the Active Template Editor Screen and identify whether the New Component should have a Closed Shape or Unclosed Shape.
4	Left-Click in the Editor Screen where the Component shall be placed.

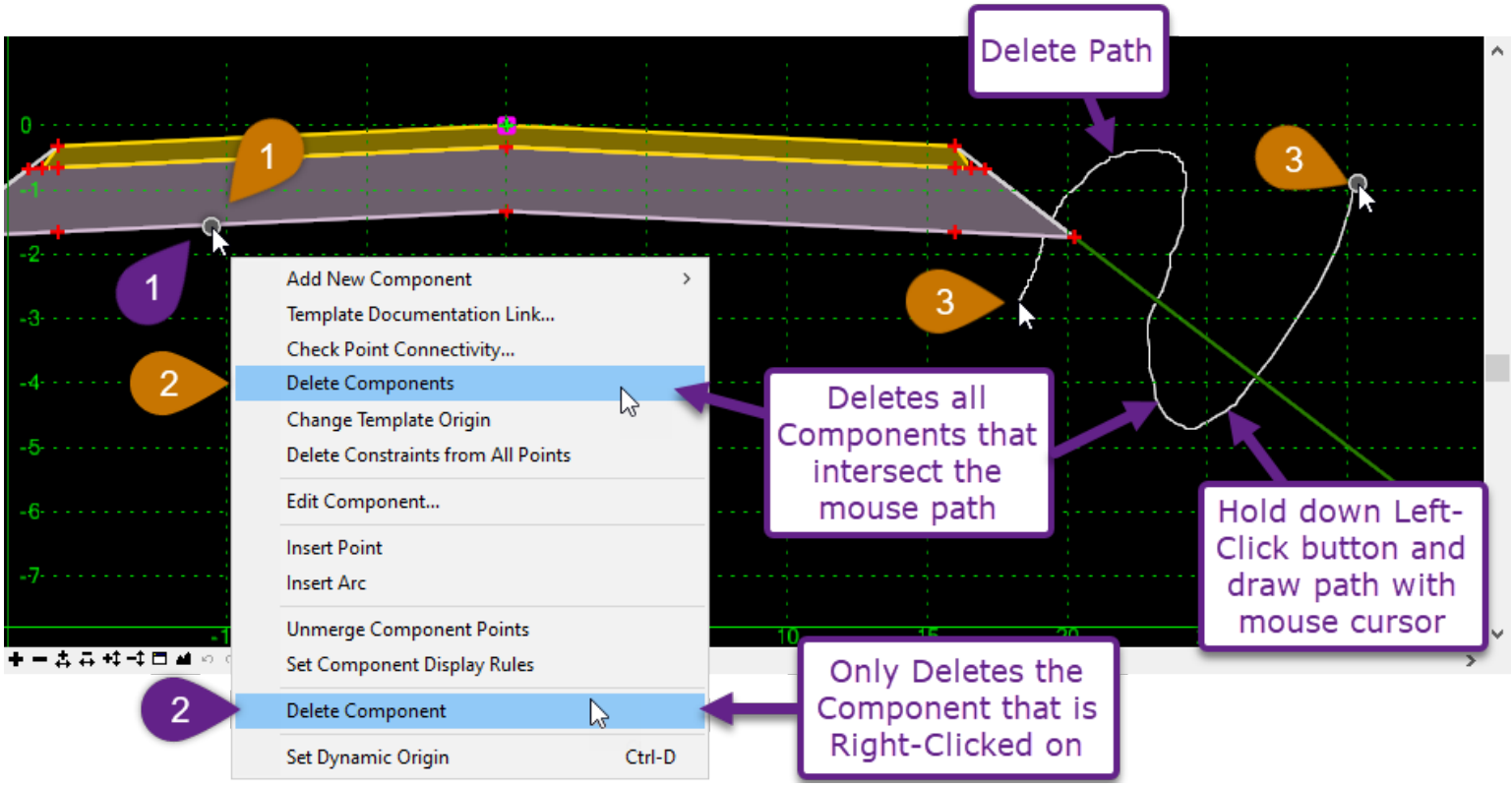
8E.10 Merge Template Components

Two Template Components can be Merged together simply by Right-Clicking on the shared border between the two Template Components and selecting *Merge Components.* For this tool to function, the Template Points located on the shared border must be *Member of* both components.



8E.11 Delete Template Components

Components can be deleted individually using the *Delete Component* tool OR multiple Components can be deleted in a single gesture with the with the *Delete Components* tool. The *Delete Components* tool allows the User to draw a path with the mouse cursor. Any component that crosses the path will be deleted.



Delete Component (Single Component)

1	Right-Click on the Component to be deleted.
2	In the Right-Click Menu, select *Delete Component*.

Delete Component (Multiple Components)

1	Right-Click anywhere in the Active Template Editor Screen.
2	In the Right-Click Menu, select *Delete Components*.
3	Left-Click at the start position for the Delete path. While holding down the Left-Click button, draw out the Delete path. Release the Left-Click button to complete the command.

8F – TEMPLATE CREATION WORKFLOWS

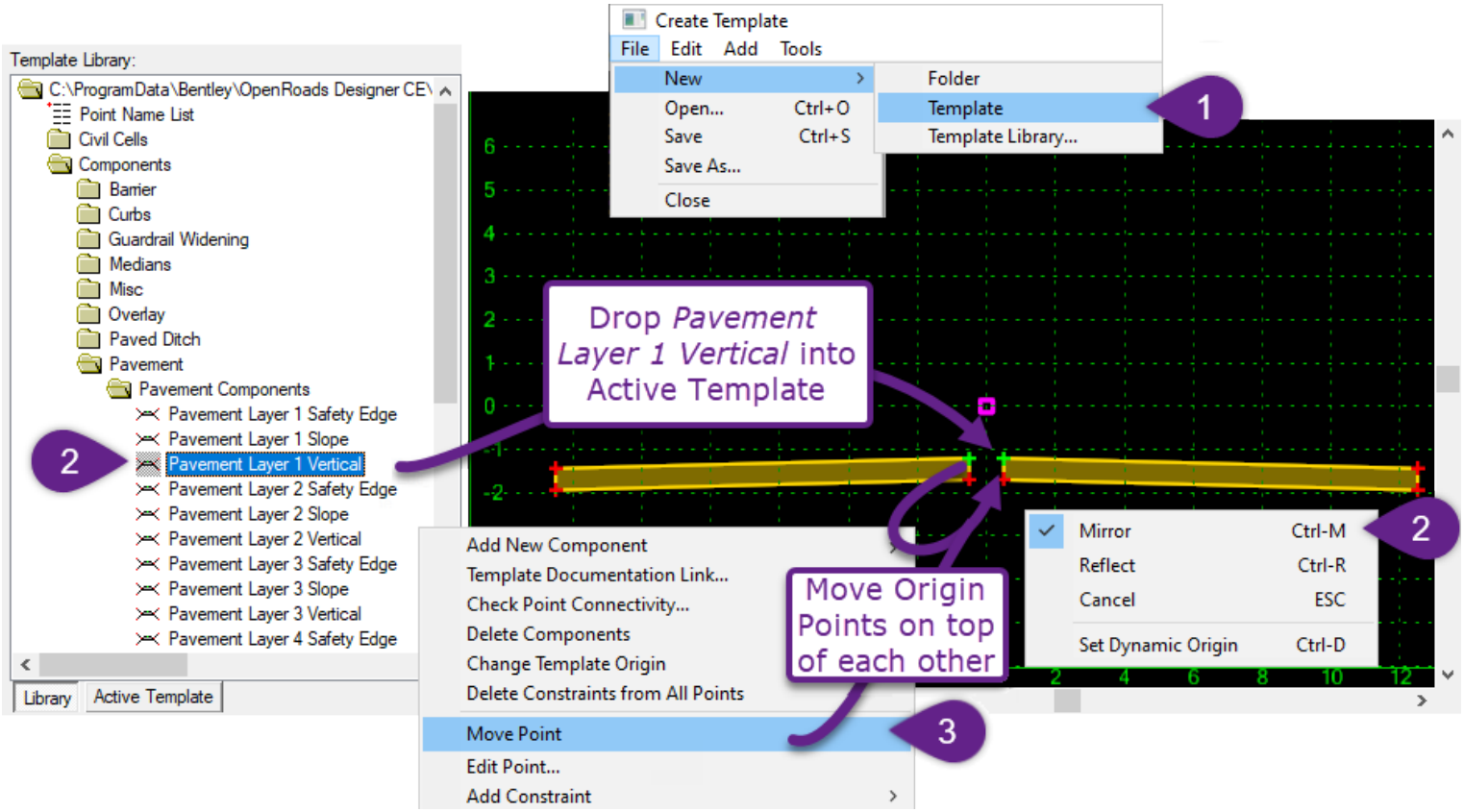
8F.1 Template Points/Components Naming and Feature Definition - Management Warning

When creating Template Points and Components from scratch, a major challenge is assigning points an appropriate **Name** and **Feature Definition**.

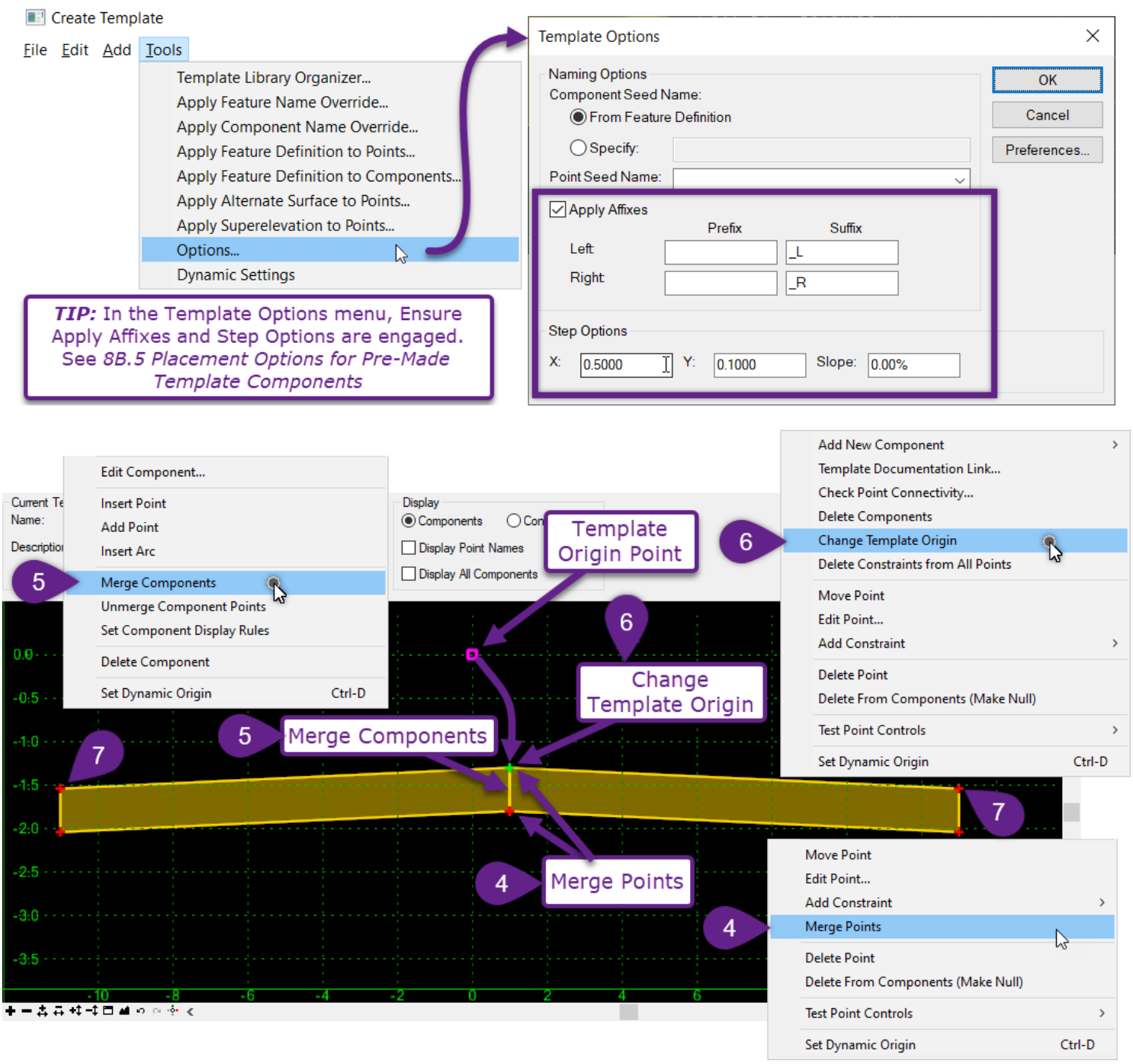
All Template Points and Components must contain an appropriate Name and Feature Definition. Using pre-made Template Components can assist in this effort because Names and Feature Definitions are a pre-set. However, before placing pre-made Template Components, ensure that Template Options have been configured. See *8B.4.a Placement Options for Pre-Made Template Components*. To assist with Naming, the Template Options will automatically place the _R and _L suffix to pre-made Template Components when placed in the Active Template.

8F.2 Simple Road Template for Superelevation and Safety Edge

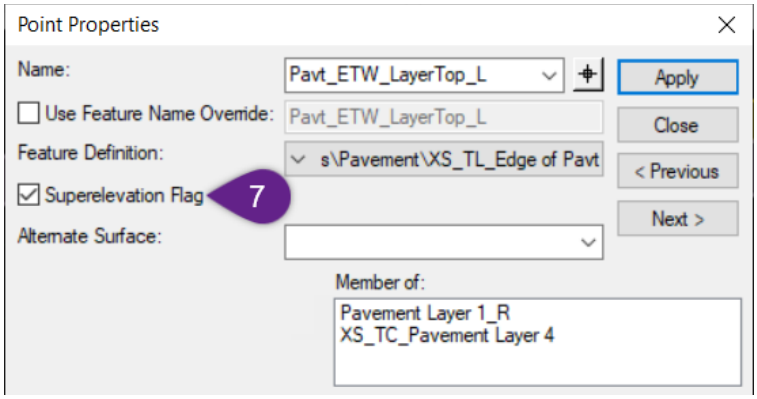
This example shows how to build a Template that will react to Superelevation and contains the proper asphalt pavement Safety Edge configuration.

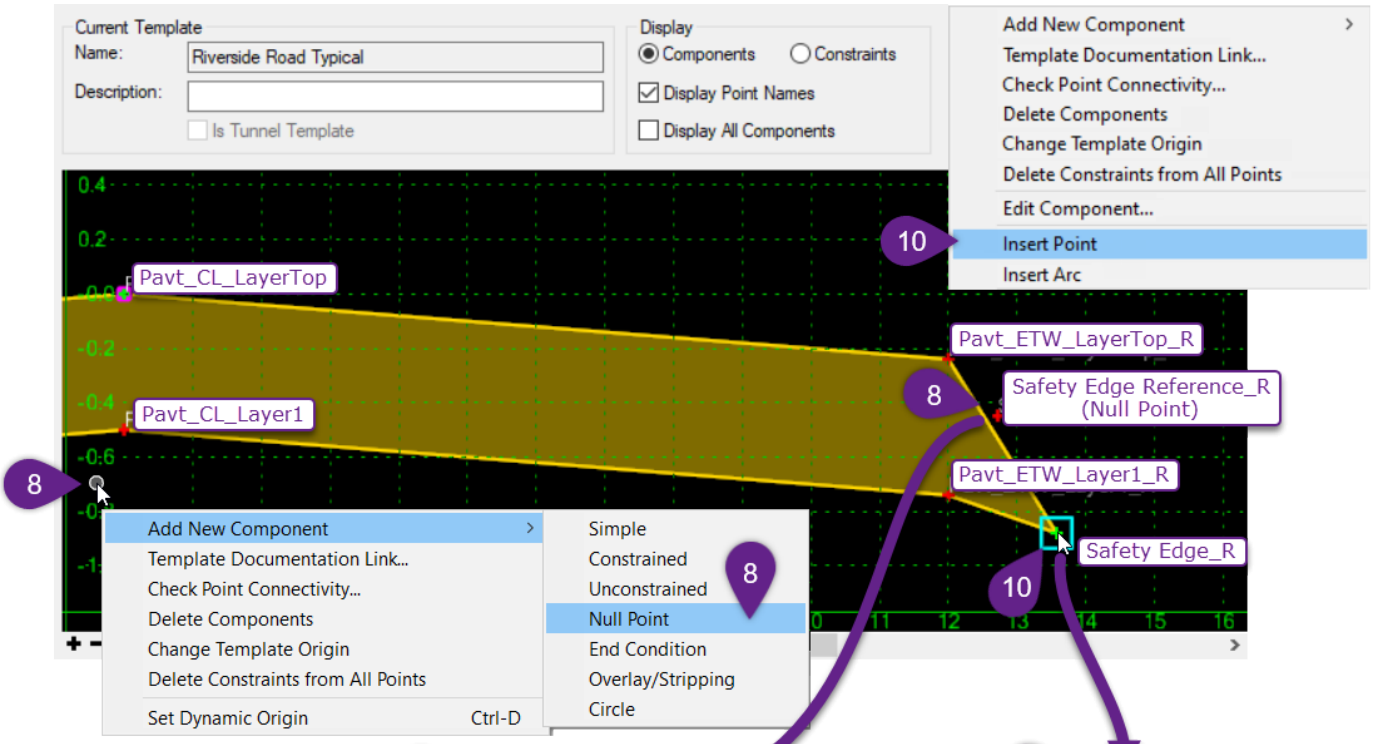


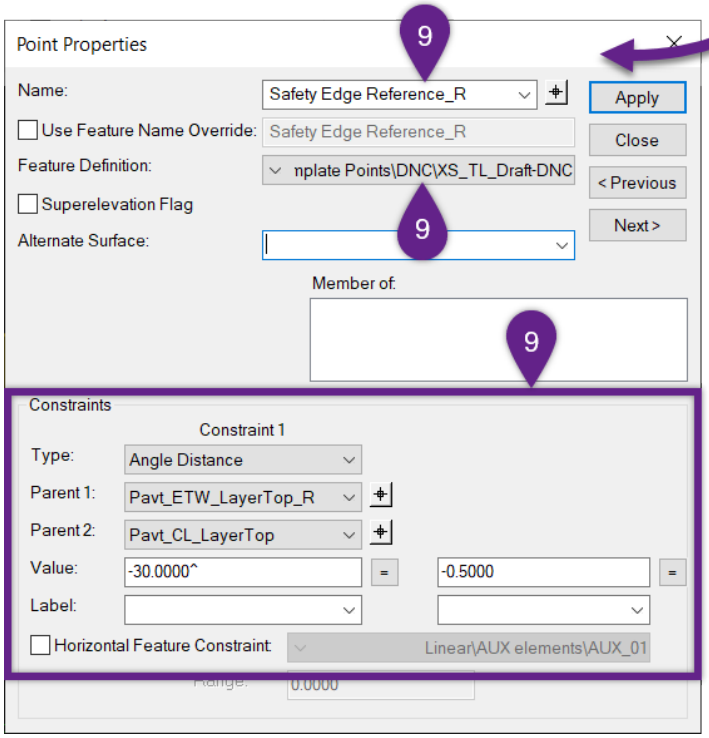
1	Create a *New Template* in the *Project Templates* folder. See *8B.3 Create a New Template or Edit a Pre-Made Template*.
2	Place the *Pavement Layer 1 Vertical* into the Active Template Editor Screen. When placing the Template Component, Right-Click to set the Placement Options to Mirror. See *8B.4 Place Pre-Made Templates and Components into Active Template*.
3	Using the Move Point tool, take the origin point of one of the *Pavement Layer 1 Vertical* components and move it on top of the other. Right-Click directly on an *unconstrained* point to access the *Move* tool. See *8E.5 Move Template Points Graphically*.

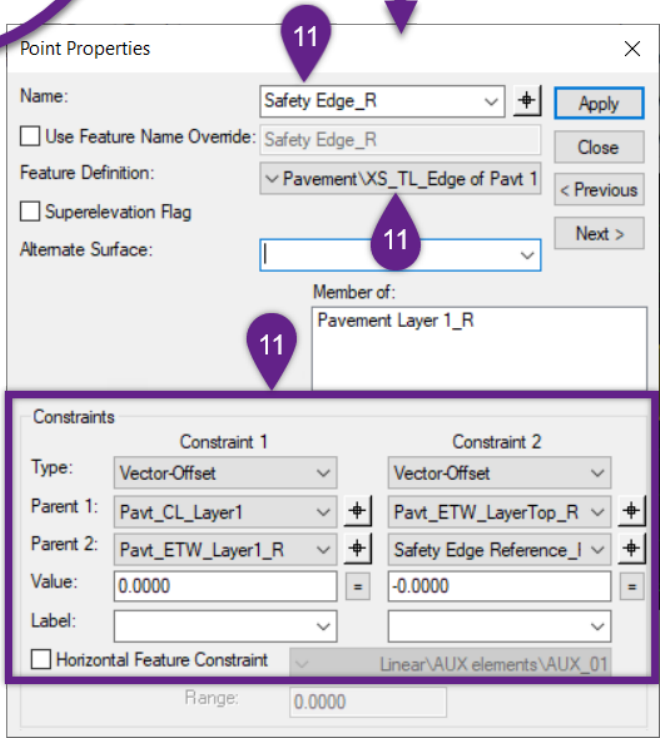


4	Using the *Merge Points* tool, merge overlapping Template Points on the shared border between Components. Right-Click directly on the Points to access the *Merge Points* tool. See *8E.2 Merge Template Points*.
5	Using the *Merging Components* tool, merge the two *Pavement Layer 1 Vertical* components into a single component. Right-Click directly on the shared border to access the *Merge Components* tool. See *8E.10 Merge Template Components*.
6	Using the *Change Template Origin* tool, set the Template Origin Point to the pavement crown point. Right-Click directly on the pavement crown point to access the *Change Template Origin* tool. See *8E.6 Change Template Origin Point*.
7	In the Point Properties for two top edge points, check the Superelevation Flag.


Point Properties
Name:
Pavt_ETW_LayerTop_L
Use Feature Name Override:
Pavt_ETW_LayerTop_L
Feature Definition:
s\Pavement\XS_TL_Edge of Pavt
Superelevation Flag
7
Alternate Surface:
Member of:
Pavement Layer 1_R
XS_TC_Pavement Layer 4
Apply
Close
< Previous
Next >


Current Template
Name:
Riverside Road Typical
Description:
Is Tunnel Template
Display
Components
Constraints
Display Point Names
Display All Components
Add New Component
Template Documentation Link...
Check Point Connectivity...
Delete Components
Change Template Origin
Delete Constraints from All Points
Edit Component...
10
Insert Point
Insert Arc
Pavt_CL_LayerTop
Pavt_CL_Layer1
Pavt_ETW_LayerTop_R
Safety Edge Reference_R
(Null Point)
Pavt_ETW_Layer1_R
Safety Edge_R
8
Add New Component
Template Documentation Link...
Check Point Connectivity...
Delete Components
Change Template Origin
Delete Constraints from All Points
Set Dynamic Origin
Ctrl-D
Simple
Constrained
Unconstrained
Null Point
End Condition
Overlay/Stripping
Circle


Point Properties
9
Name:
Safety Edge Reference_R
Use Feature Name Override:
Safety Edge Reference_R
Feature Definition:
nplate Points\DNC\XS_TL_Draft-DNC
Superelevation Flag
Alternate Surface:
Member of:
Apply
Close
< Previous
Next >
Constraints
Constraint 1
Type:
Angle Distance
Parent 1:
Pavt_ETW_LayerTop_R
Parent 2:
Pavt_CL_LayerTop
Value:
-30.0000^
-0.5000
Label:
Horizontal Feature Constraint
Linear\AUX elements\AUX_01
Range:
0.0000


Point Properties
11
Name:
Safety Edge_R
Use Feature Name Override:
Safety Edge_R
Feature Definition:
Pavement\XS_TL_Edge of Pavt 1
Superelevation Flag
Alternate Surface:
Member of:
Pavement Layer 1_R
Apply
Close
< Previous
Next >
Constraints
Constraint 1
Constraint 2
Type:
Vector-Offset
Vector-Offset
Parent 1:
Pavt_CL_Layer1
Pavt_ETW_LayerTop_R
Parent 2:
Pavt_ETW_Layer1_R
Safety Edge Reference_I
Value:
0.0000
-0.0000
Label:
Horizontal Feature Constraint
Linear\AUX elements\AUX_01
Range:
0.0000

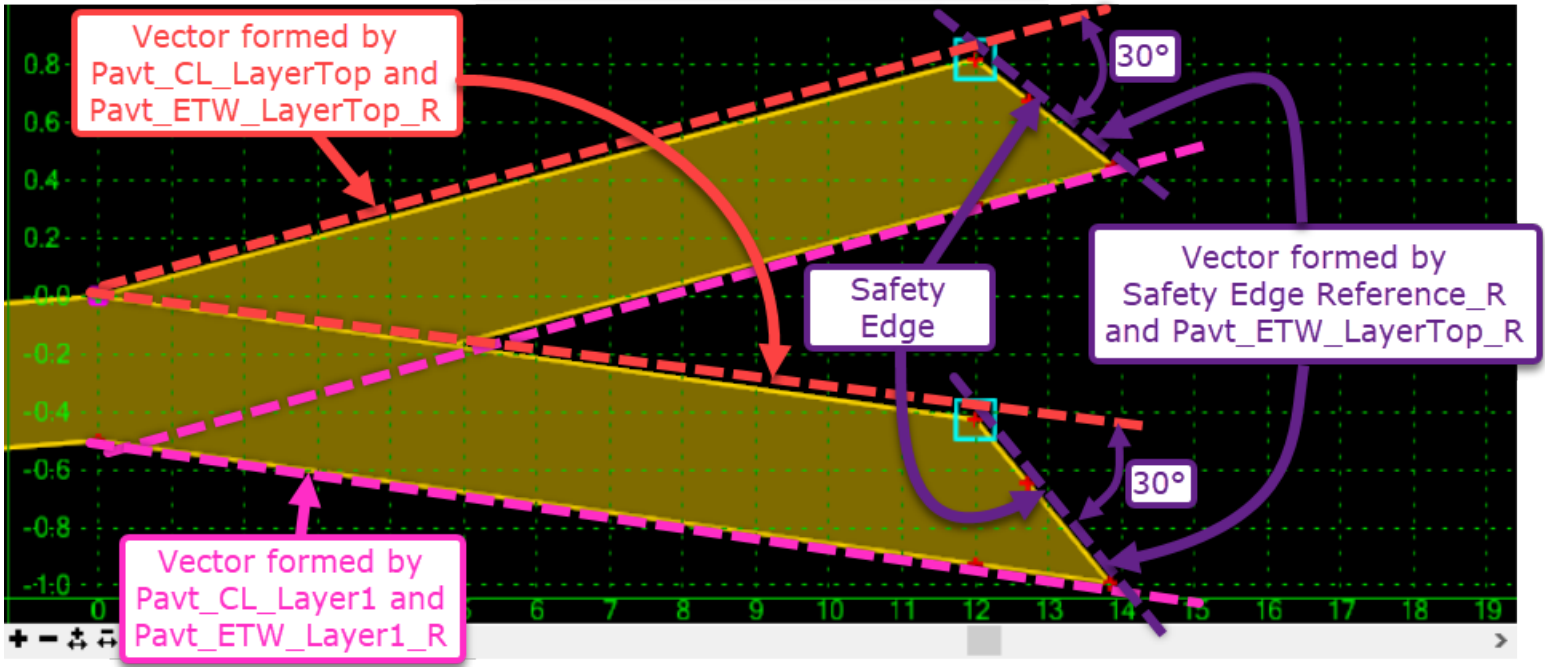
8	Right-Click anywhere in the Active Template Screen Editor and create a Null Point. Place it near the edge of road template. See *8E.7 Create New Template Components*.
9	Access the Point Properties of the newly-created Null Point to edit the Symbology. See *8C.2 Point Feature Definition and Name Properties*. **Name =** Safety Edge Reference_R **Feature Definition =** XS_TL_Draft-DNC ***NOTE:*** This Null Point is used for reference only. The XS_TL_Draft-DNC is a *Do Not Construct* Feature Definition – so this Template Point will not be shown in the 2D Design Model. Assign the Angle-Distance Constraint to the Null Point. See *8C.6.a.xii Angle Distance*. **Constraint 1** **Parent 1** = Pavt_ETW_LayerTop_R **Parent 2** = Pavt_CL_LayerTop **Angle Value** = -30.000° **Distance Value** = -0.5000’
10	Right-Click on the Component boundary in between Pavt_ETW_LayerTop_R and Pavt_ETW_Layer1_R. Insert a Point into the Asphalt Component and place it near the reference Null Point. See *8E.1 Insert a Point into a Template Component*.
11	Access the Point Properties of the newly-created Template Point to edit the Symbology. **Name** = Safety Edge_R **Feature Definition** = XS_TL_Edge of Pavt 1 Assign the Vector-Offset to both Constraints. See *8C.6.a.v Vector Offset*. **Constraint 1** **Type** = Vector Offset **Parent 1 =** Safety Edge Reference_R **Parent 2 =** Pavt_ETW_LayerTop_R **Offset Value =** 0.0000’ **Constraint 2** **Type** = Vector Offset **Parent 1** = Pavt_ETW_Layer1_R **Parent 2** = Pavt_CL_Layer1_R **Offset Value** = 0.0000’

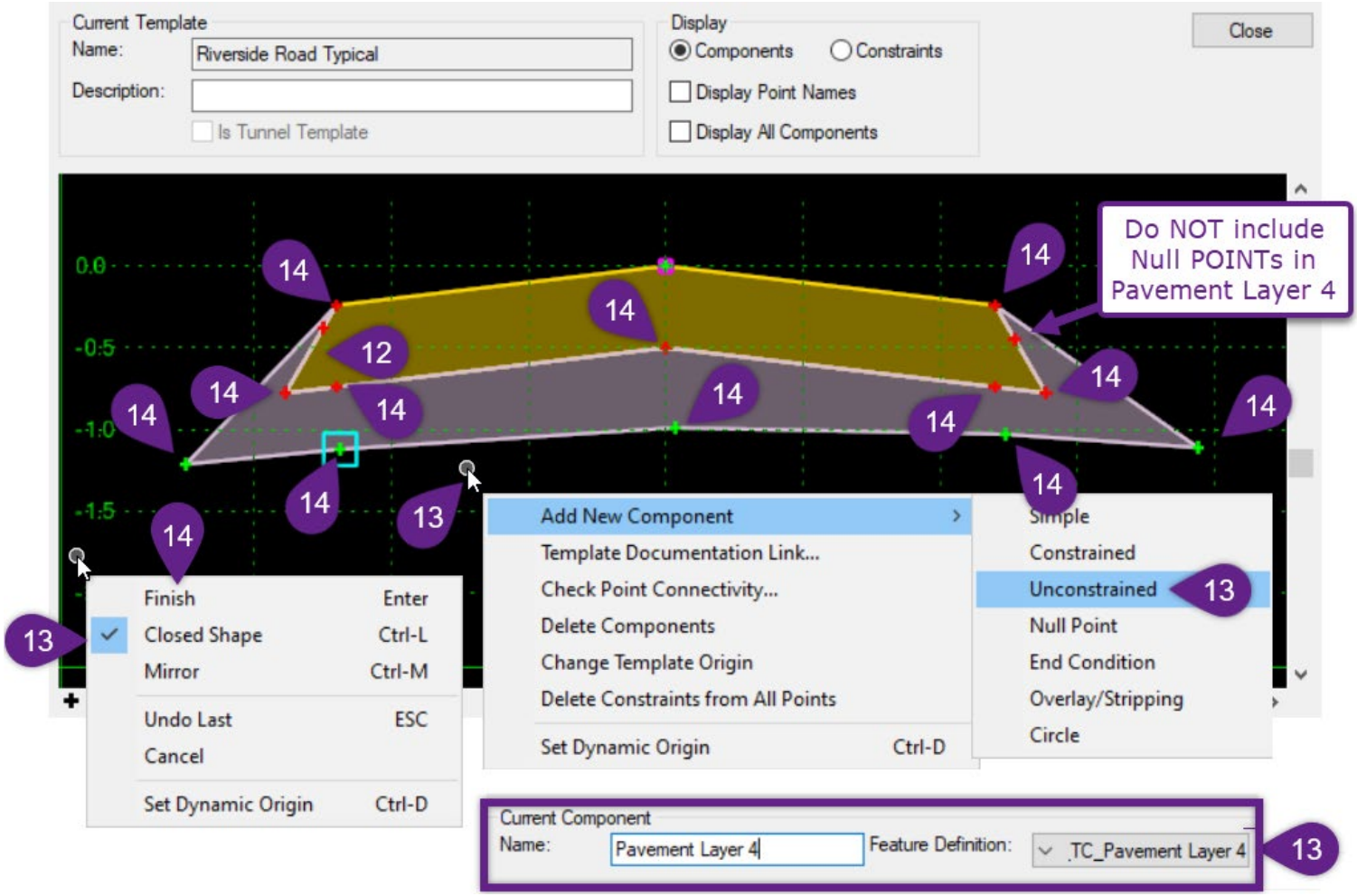
Null Point Reference Explanation: The Null Point is necessary to maintain a 30° Safety Edge slope – even when the pavement cross-slope is superelevated. The pavement cross-slope is defined by the vector between Pavt_CL_LayerTop and Pavt_ETW_LayerTop_R. The Angle Distance constraint type for the Null Point uses this vector as a reference to ensure the Null Point is placed RELATIVE to the pavement cross slope. The Distance Value for the Null Point is unimportant and can be ANY value other than zero.

The Safety Edge_R point will be placed at the intersection of the two vectors specified in its constraints – the vector formed by the bottom of the pavement (Pavt_CL_Layer1_R and Pavt_ETW_Layer1_R) and a 30° vector formed by the Null Point and top edge of pavement (Safety Edge Reference_R and Pavt_ETW_LayerTop_R). This configuration ensures that the Safety Edge maintains proper geometry IF the pavement slope is varied due to superelevation operations.

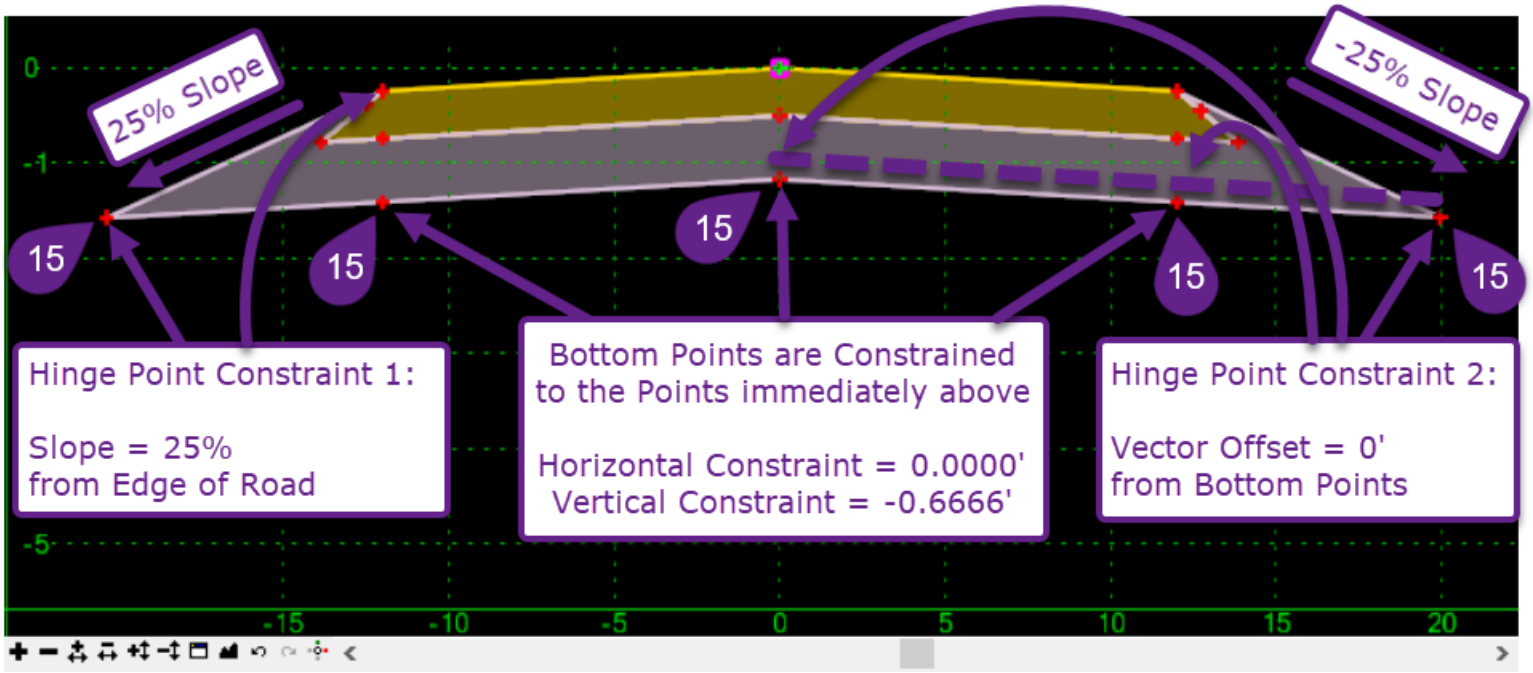
To test Safety Edge behavior under superelevation conditions, see *8C.3 Superelevation Flag*.

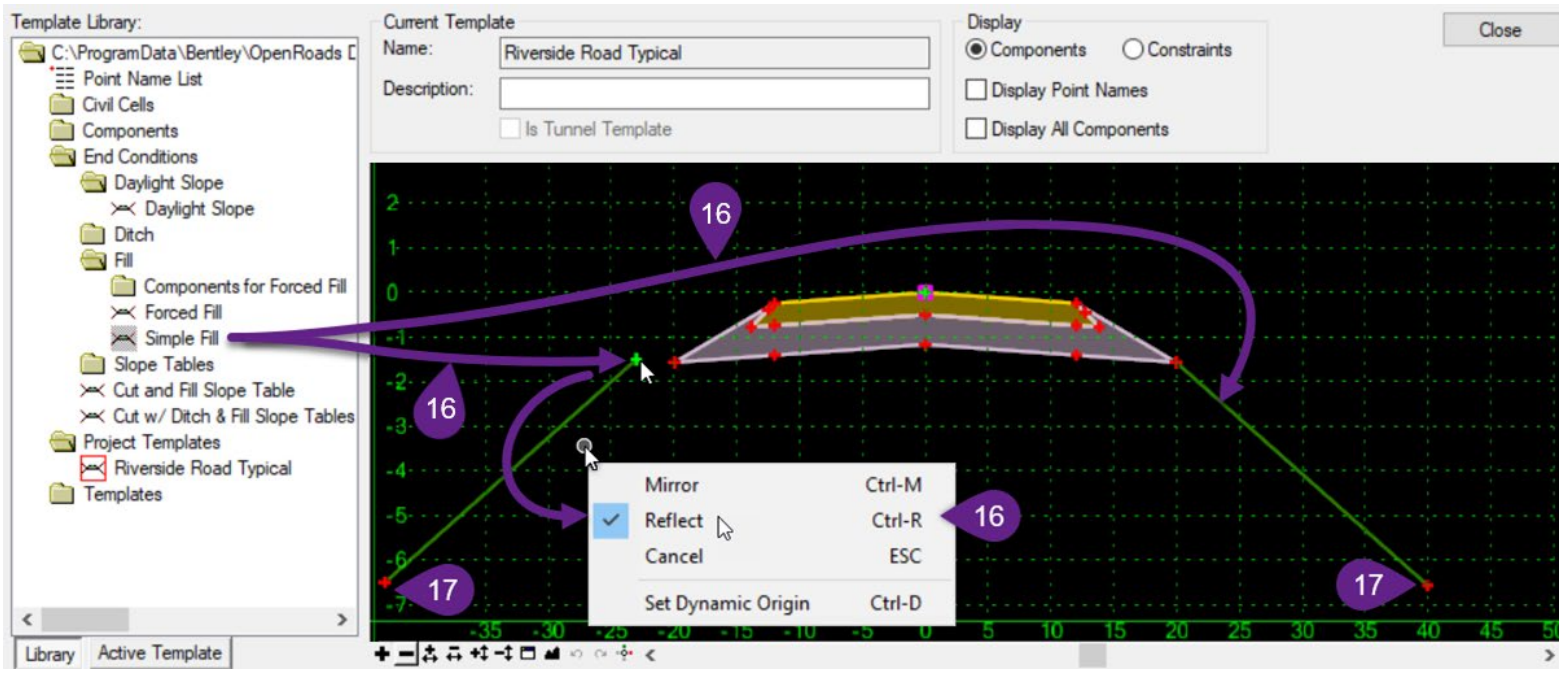
Safety Edge in Superelevation Scenario



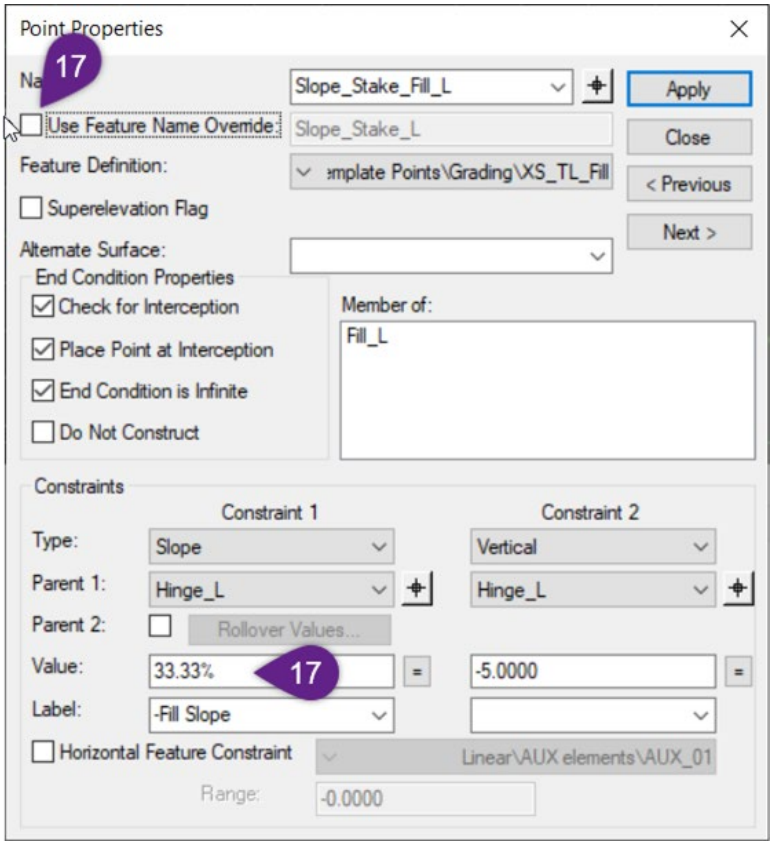


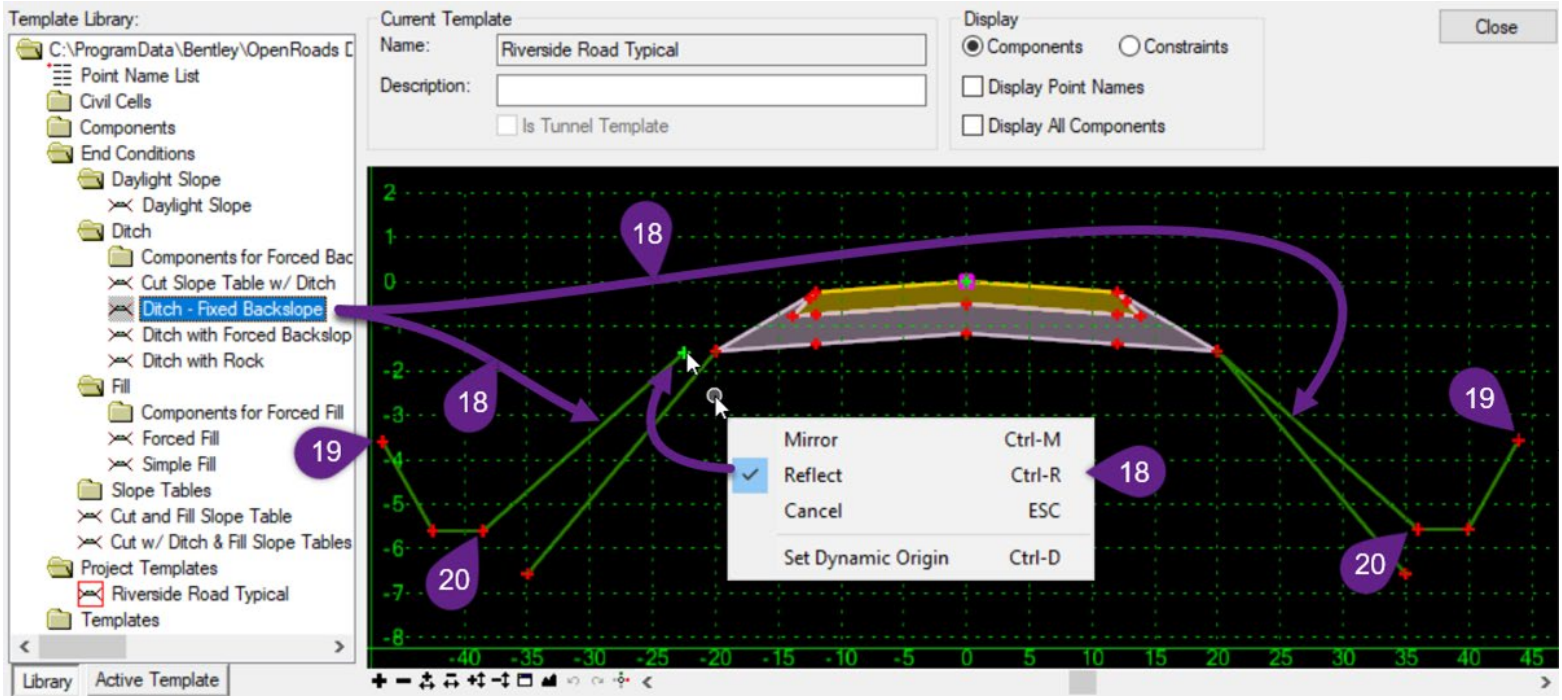
12	Create the Safety-Edge for the left side of the Template. Repeat Steps 7–10.
13	Right-Click in the Active Template Editor Screen and create a new *Unconstrained* Component. Below the Active Template Editor Screen, give the new component an appropriate Name and Feature Definition. Ensure that *Closed Shape* is enabled. See *8E.7 Create New Template Components*.
14	Roughly delineate the new Component's shape by clicking in the locations shown above. When satisfied with the shape, Right-Click in the Active Template Editor Screen and select *Finish*.
15	Assign the 5 unconstrained points appropriate Names, Feature Definitions, and Constraints; as shown below.



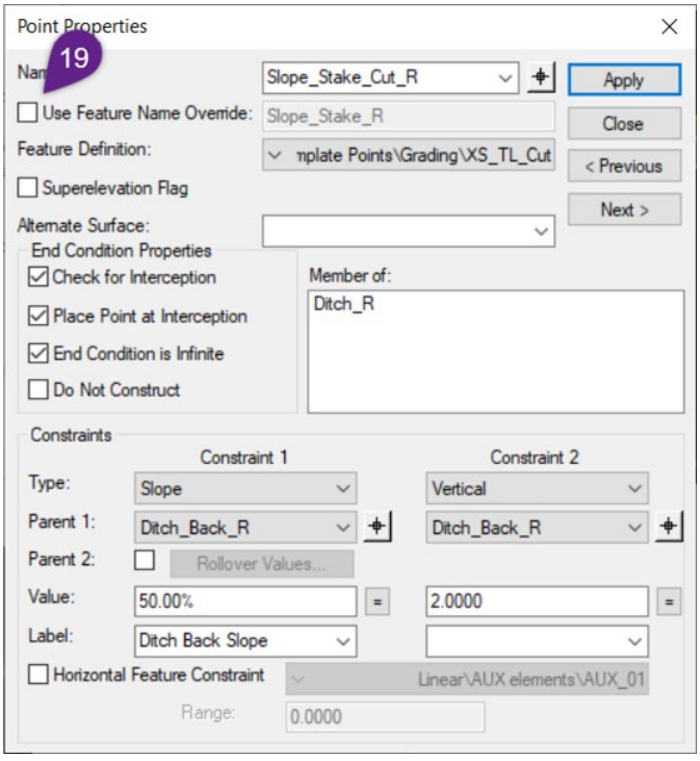


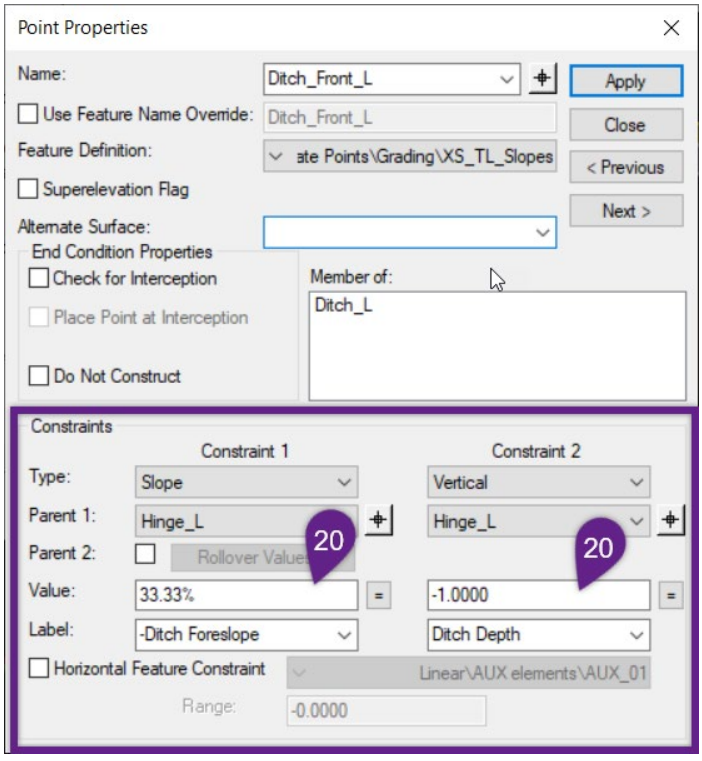
16	Place the *Simple Fill* Template Component directly on the Hinge Point for each side. When placing the Template Component on the left side, Right-Click and select *Reflect.* See *8B.4 Place Pre-Made Templates and Components into Active Template*
17	Access the Point Properties for the *Slope_Stake_Fill* points and change the Slope Constraint from 25% to 33.33% (1V:3H). Also, uncheck the Use Feature Name Override box. *See 8C.7.b Feature Name Override Convention for End Condition Points*.

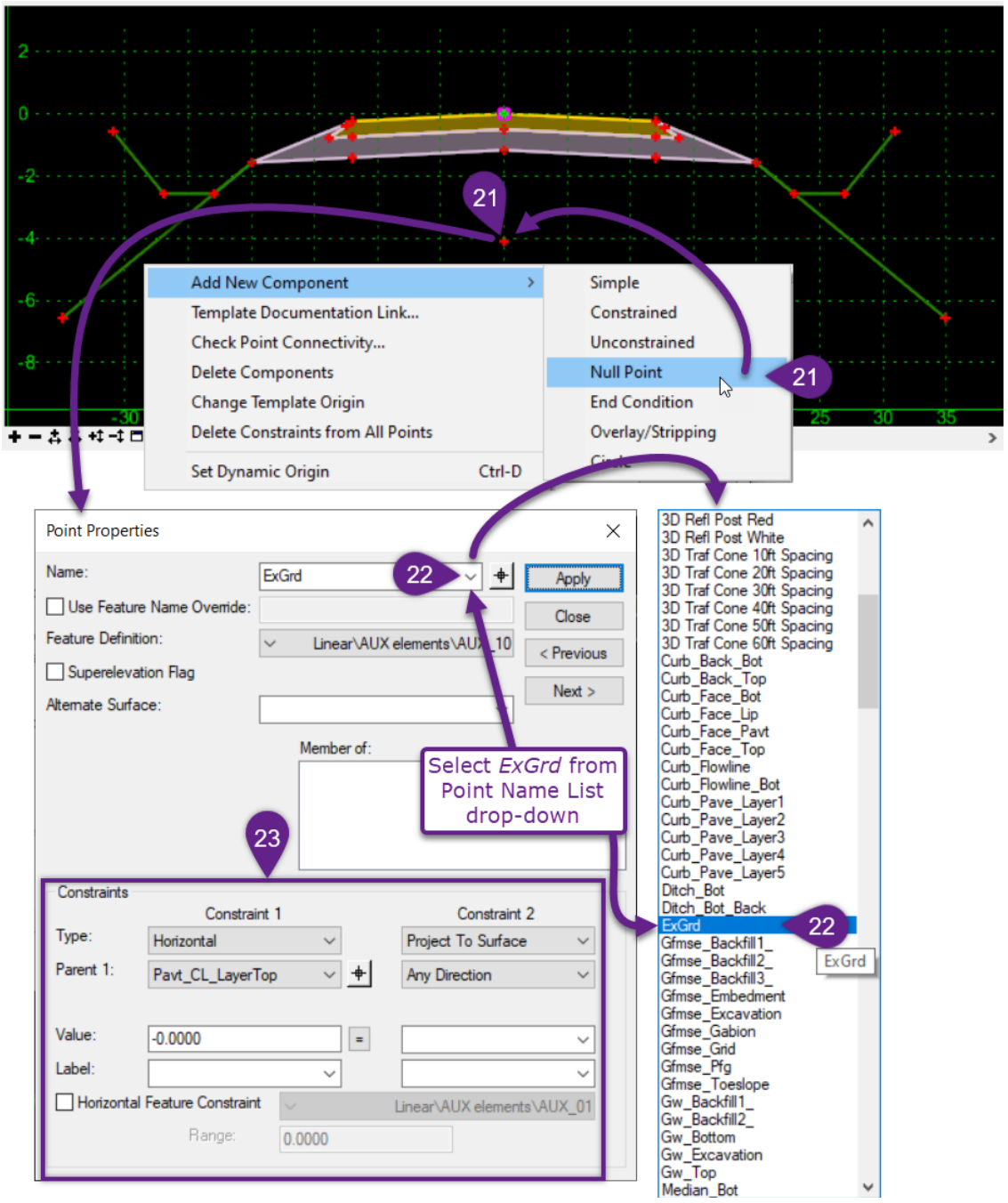




18	Place the *Ditch – Forced Backslope* in the same manner as Step 16.
19	Access the Point Properties for the *Slope_Stake_Cut* points and uncheck the *Use Feature Name Override* box. Also, uncheck the Use Feature Name Override box. See *8C.7.b Feature Name Override Convention for End Condition Points*.
20	Access the Point Properties for the *Ditch_Front* points and change the *Slope Constraint* from 25% to 33.33%. Change the *Vertical Constraint* from -4.0000' to -1.0000'. This creates a Ditch that has the same foreslope as the Fill End Condition with a bottom that is 1' below the hinge point.





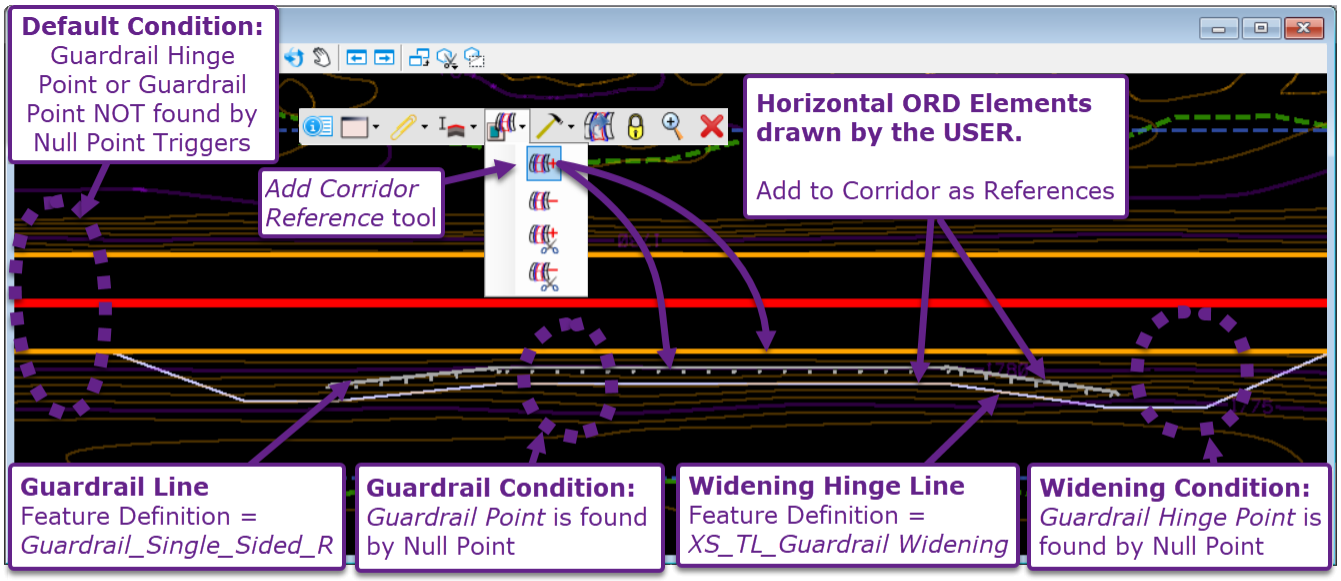


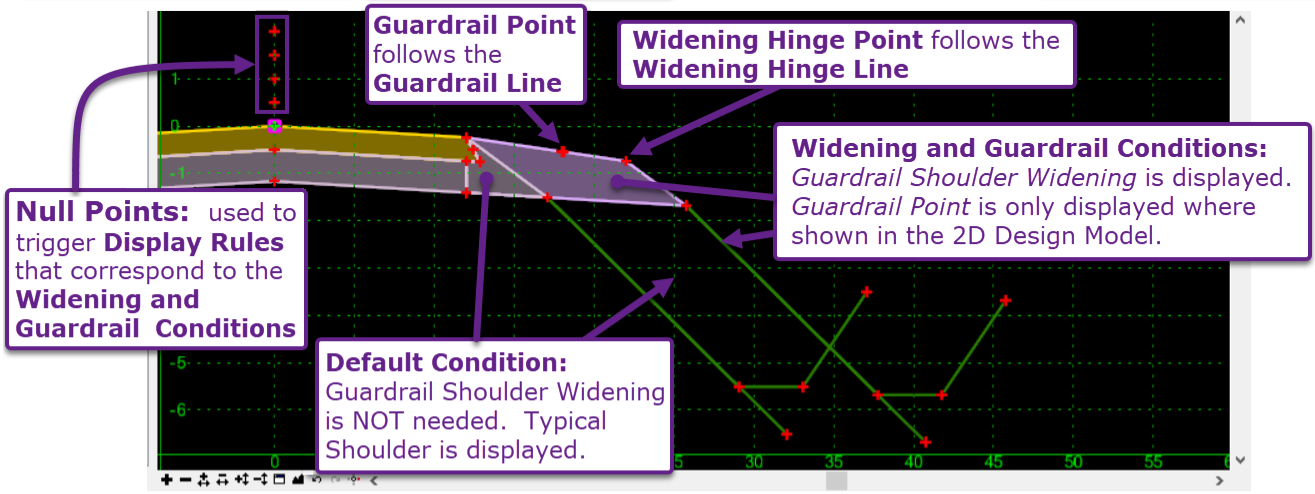
21	Right-Click anywhere in the Active Template Screen Editor and create a Null Point. Place it underneath the aggregate component.
22	Access the Point Properties for the newly-created Null Point and assign a Name and Feature Definition by selecting *ExGrd* from the Point Name List drop-down. ***NOTE:*** This ExGrd Null Point is necessary for labeling the existing ground elevation in Cross Section Production.
23	In the Point Properties, assign the ExGrd null point Constraints. See *8C.6.a.vi Project To Surface*. **Constraint 1** **Type** = Horizontal **Parent 1 =** Pavt_CL_LayerTop **Value =** 0.0000' **Constraint 2** **Type =** Project To Surface **Parent 1** = Any Direction

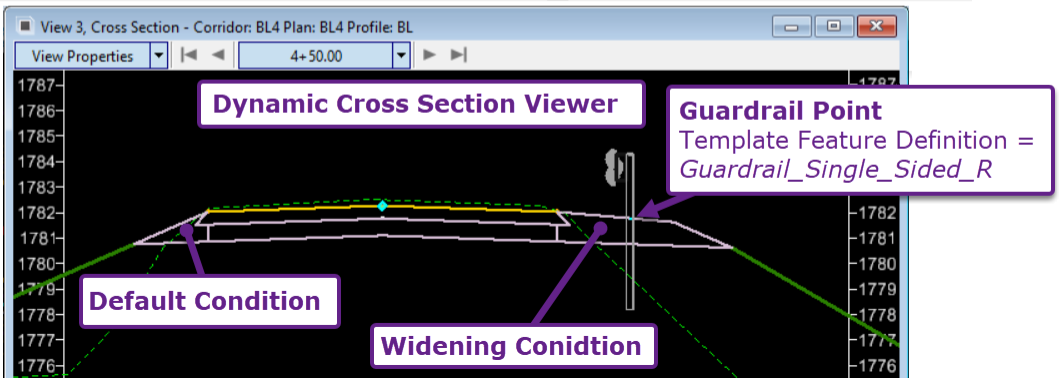
8F.3 Advanced Road Template with Guardrail and Display Rules

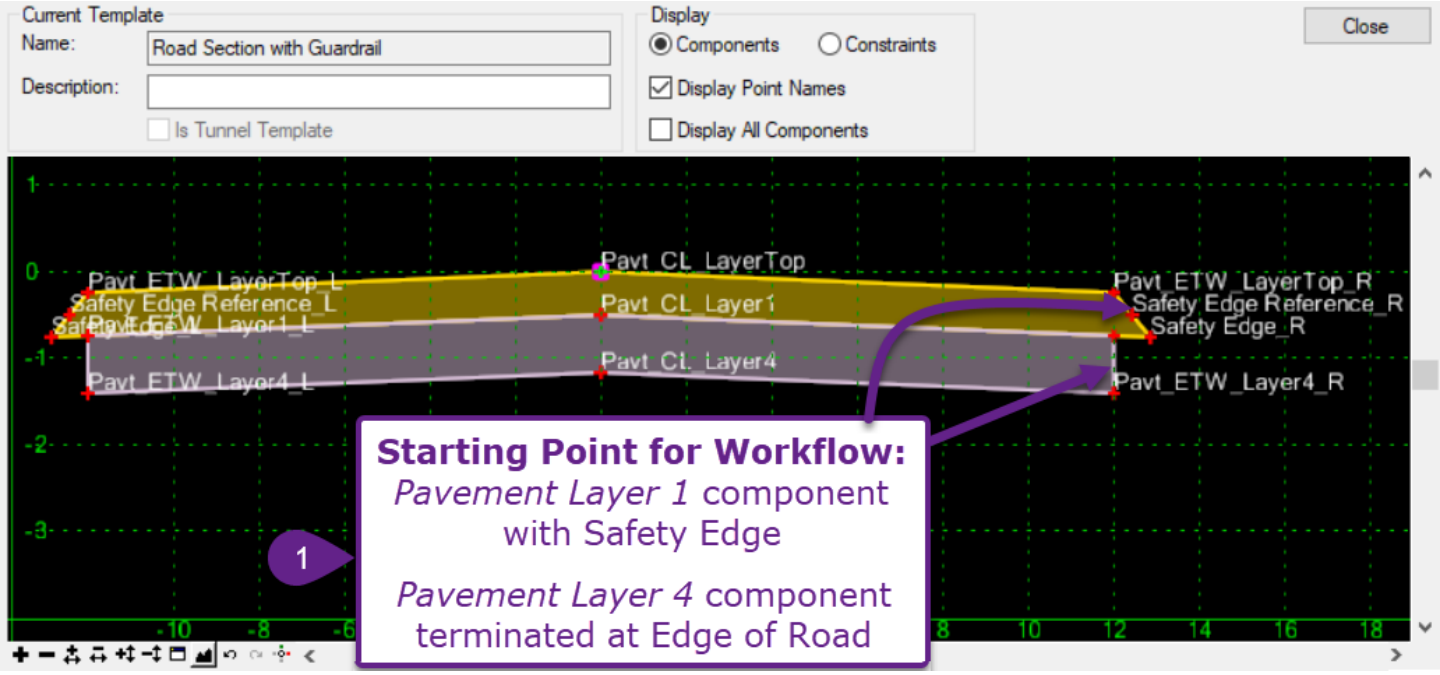
The following workflow demonstrates how to build a template that accommodates both the *typical* road condition AND a guardrail shoulder widening condition. Prior to this workflow, the **Widening Hinge Line** and **Guardrail Line** must be manually drawn using Horizontal ORD Elements with appropriate Feature Definition. The Horizontal ORD Elements must be added to the Corridor as a Reference. See *9G.1 WARNING – Creating Circular References (Recursive Solutions)*. ***NOTE:*** This workflow is done for the Right-Side of the Template only but is applicable to the Left-Side as well.

The **Default Condition** for this template will represent a typical road section.
The **Widening Condition** and **Guardrail Condition** use Display Rules, Null Points, and Horizontal Feature Constraints to adjust to the Horizontal ORD Elements drawn by the User in the 2D Design Model.





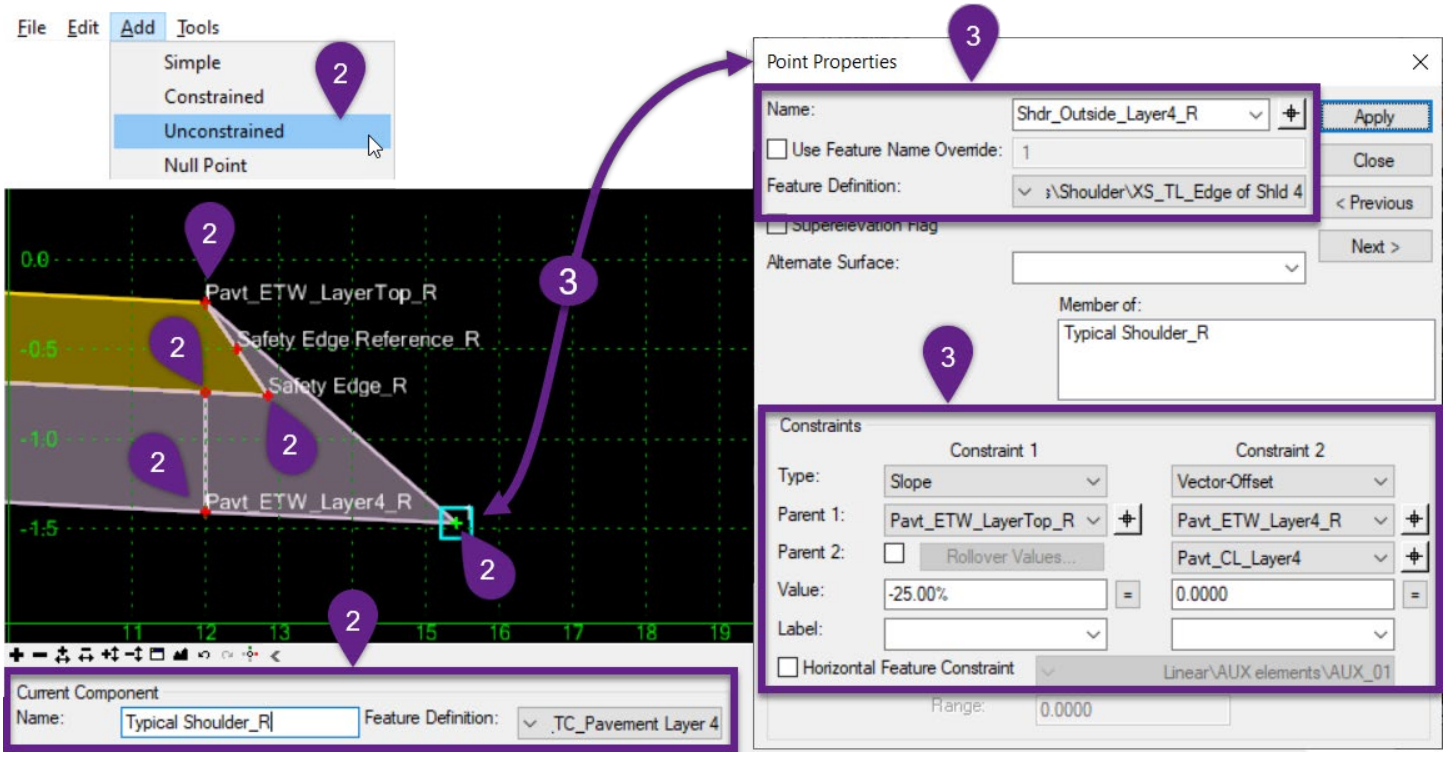


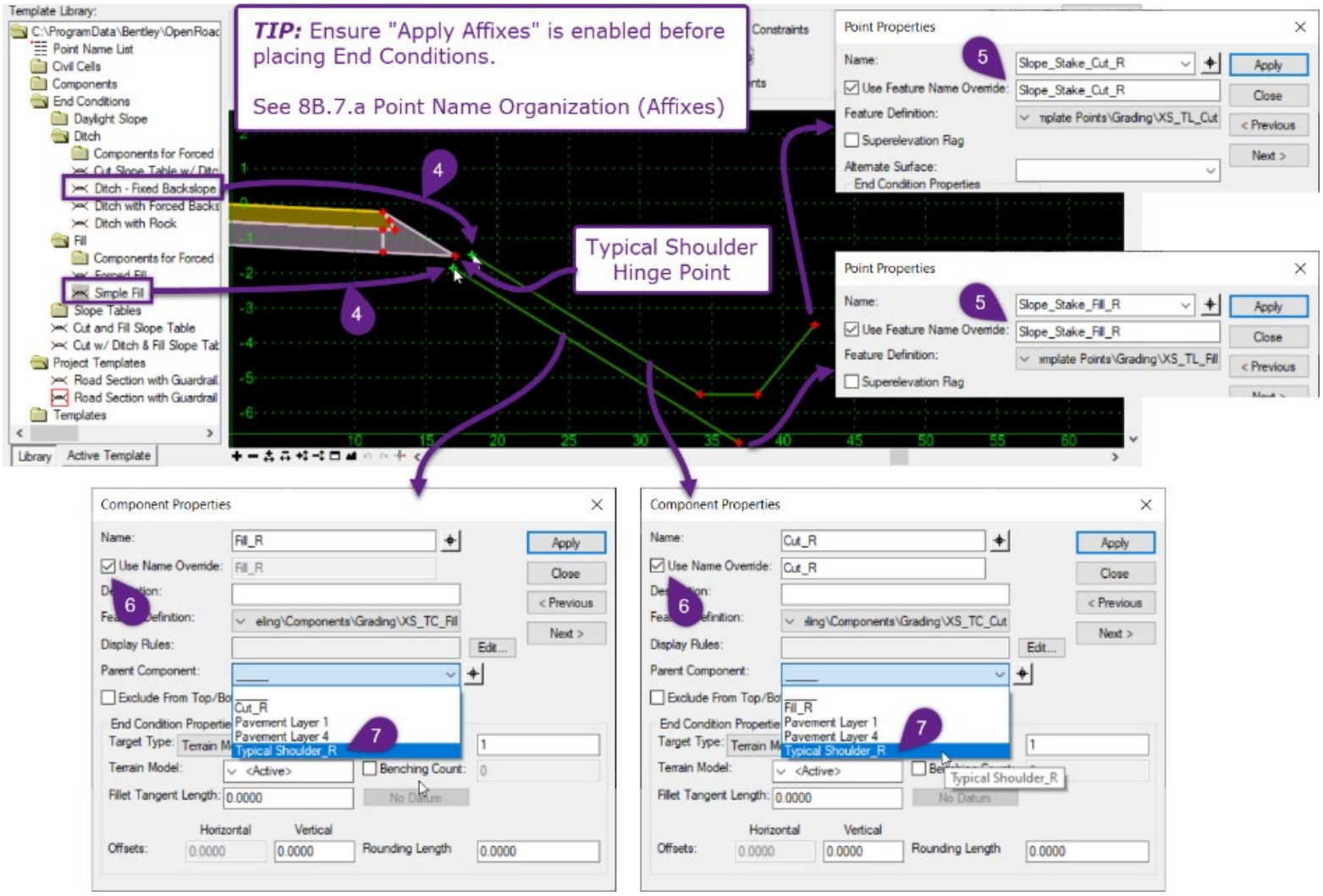


1	Prior to this workflow, Template Components were made for an asphalt component (*Pavement Layer 1*) and an aggregate component (*Pavement Layer 4*).

8F.3.a Create Template Components for the *Default Condition*

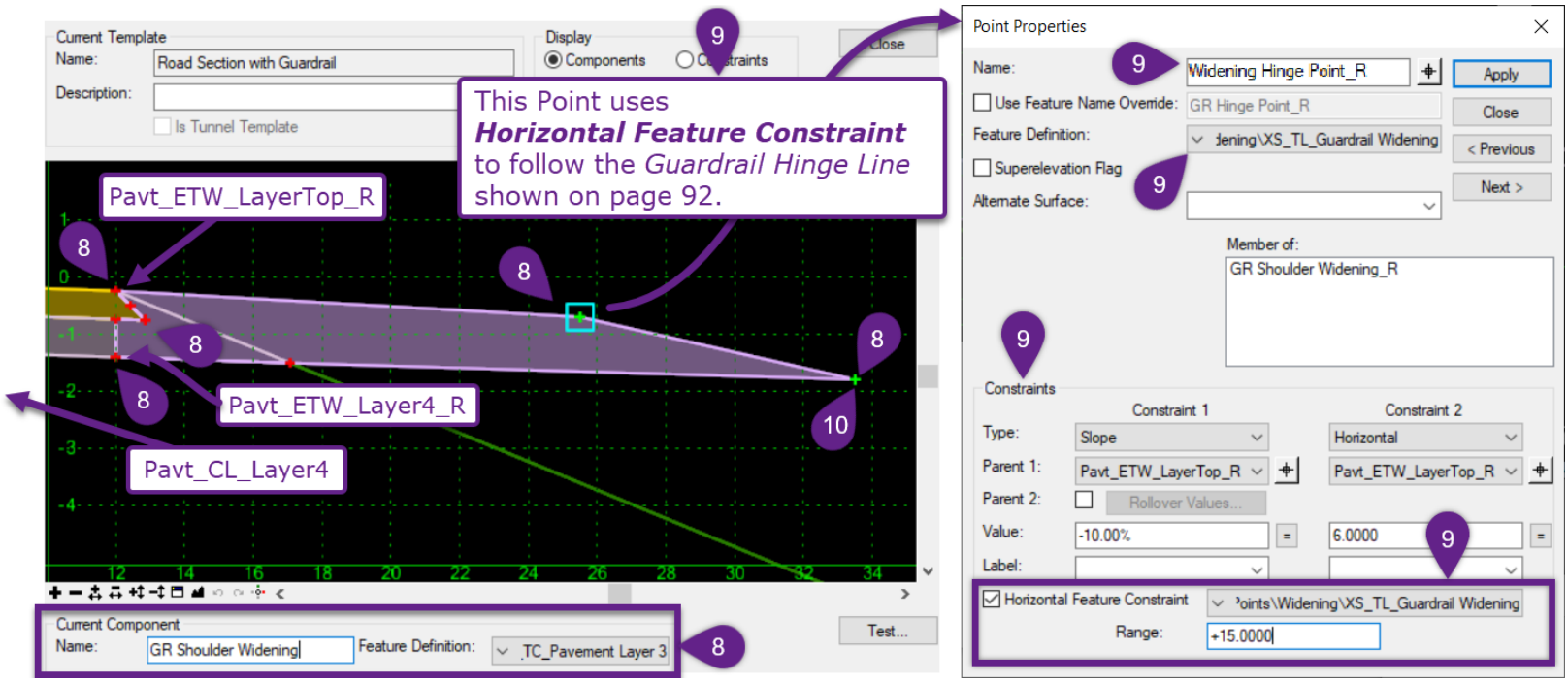
2	Create the *Typical Shoulder_R* component. Delineate the component shape by clicking in the locations shown below. Assign the component a Name and Feature Definition. The **Name** used in this example is *Typical Shoulder_R*. When satisfied with shape, Right-Click and select Finish. See *8E.8 Create a New Component - Workflow*.
3	Assign a Name, Feature Definition, and Constraints to the typical shoulder hinge point (Shldr_Outside_Layer4_R).





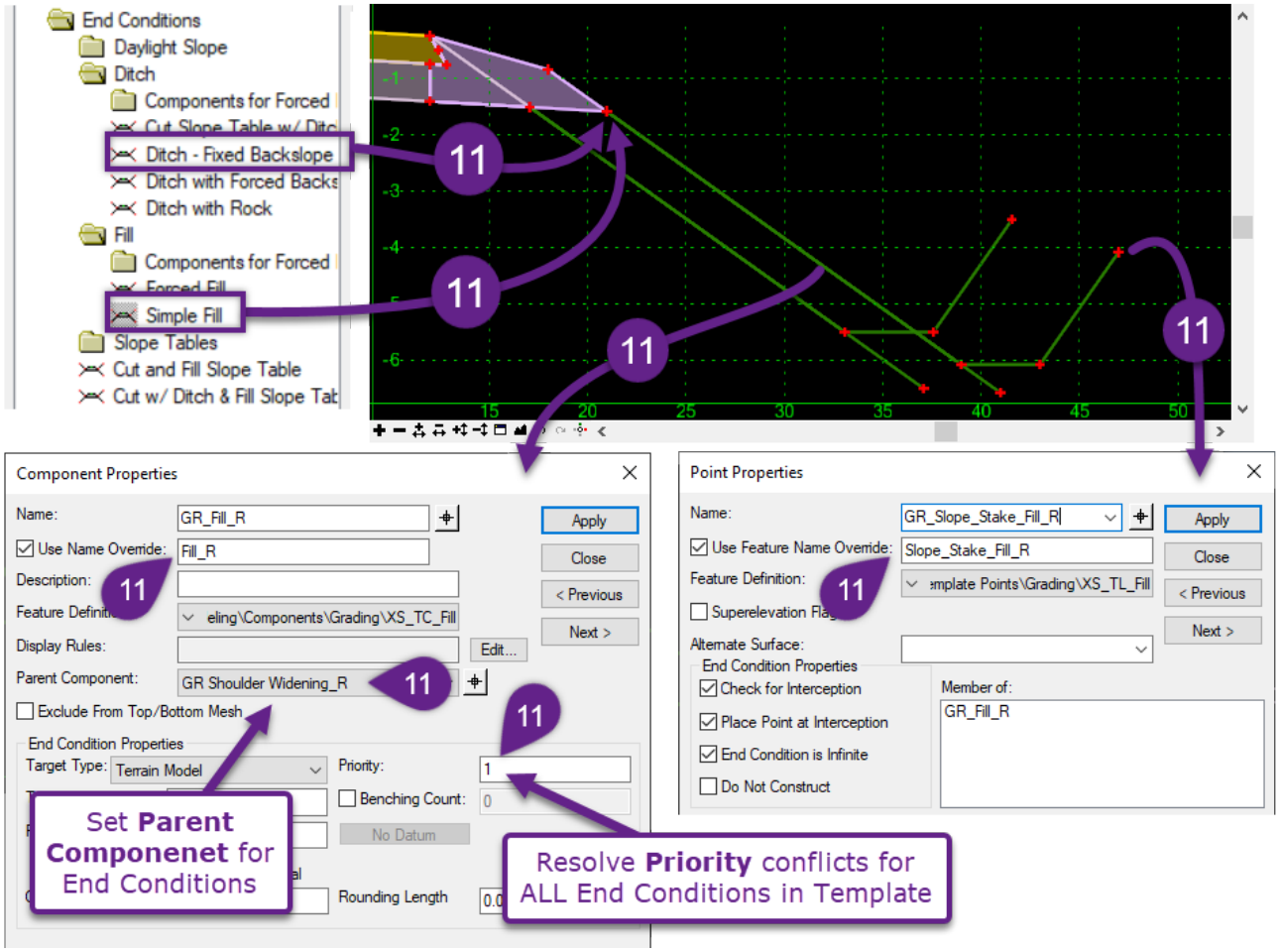
4	From the FLH Standard Template Library, place the Cut and Fill End Condition directly on top of the Typical Shoulder Hinge Point. **Component Location in the FLH Standard Template Library:** Cut: *End Conditions → Ditch → Ditch – Fixed Backslope* Fill: *End Conditions → Fill → Simple Fill*
5	Access the *Point* Properties for both End Condition Template Points and change the *Feature Name Override* to "*Slope_Stake_**Cut/Fill**_R"*. Ensure the box is checked. See *8C.7.b Feature Name Override Convention for End Condition Points*.
6	Access the *Component* Properties for both End Condition Components and check the *Use Name Override* box.
7	Access the Component Properties for both End Conditions Template Components and change the *Parent Component* to *Typical Shoulder_R*. See *8D.3 Parent Components*. The purpose of this step is to make Cut/Fill End Condition components dependent to the *Typical Shoulder_R* component.

8F.3.b Create Template Components for the *Widening Condition*



8	Create the *GR_Shoulder Widening_R* component. Delineate the component shape by clicking in the locations shown above. Assign the component an appropriate Name and Feature Definition. The **Name** used in this example is GR_Shoulder *Widening_R*. When satisfied with shape, Right-Click and select Finish. See *8E.8 Create a New Component - Workflow*.
9	Access the Point Properties for Guardrail Widening Hinge Point: Assign the Point a **Name:** *Widening Hinge Point_R* is used in this example. Assign the Point a **Feature Definition:** *XS_TL_Edge of Pavt 4* is used this example. Assign the **Point Constraints** as shown. The Slope Value is set to -10% relative to the *Pavt_ETW_LayerTop_R* (Parent 1) point. The *Horizontal* constraint value is inconsequential in this example - because the *Horizontal Feature Constraint* is used. Check the **Horizontal Feature Constraint** box and assign the *XS_TL_Guardrail Widening* Feature Definition from the drop-down. This is the same Feature Definition applied to the Horizontal ORD Element shown on page 8-101, called *Guardrail Hinge Line.* The horizontal position of this Template Point will follow this ORD Element, similar to a *Horizontal Point Control*. Set the **Range** Value for the *Horizontal Feature Constraint*. In this example, Range is set to **+**15.0000. ***WARNING:*** To ensure the Horizontal Feature Constraint is searching in the intended direction, a **POSITIVE VALUE** is used on the **RIGHT SIDE** of the Template. Use a **NEGATIVE VALUE** if this step is performed on the **LEFT SIDE.**
10	Access the Point Properties for the last unconstrained point: **Name:** *GR_Outside_Layer4_R* is used in this example. **Feature Definition:** *XS_TL_Edge of Pavt 4* is used in this example. **Point Constraints** as follows: The **Slope Value** is set to -25% relative to the *GR_Outside_Layer4_R* (Parent 1) point. The **Vector-Offset** is used with *Pavt_ETW_Layer4_R* (Parent 1) point and *Pavt_CL_4* (Parent 2).

11	**Create End Conditions** for the *GR_Shoulder Widening* component by repeating **steps 4–7**. Ensure End Condition components are dependent on the *GR_Shoulder Widening* component with the **Parent Component** function in the Component Properties. Ensure Point and Component **Name Overrides** match the *Overrides* shown in **steps 5–6**. In the Component Properties, resolve conflicting **Priorities** between ALL End Conditions in the Template. See *8C.7.e Multiple End Conditions and End Condition Priorities*.



8F.3.c Create Template Component for the *Guardrail Condition*

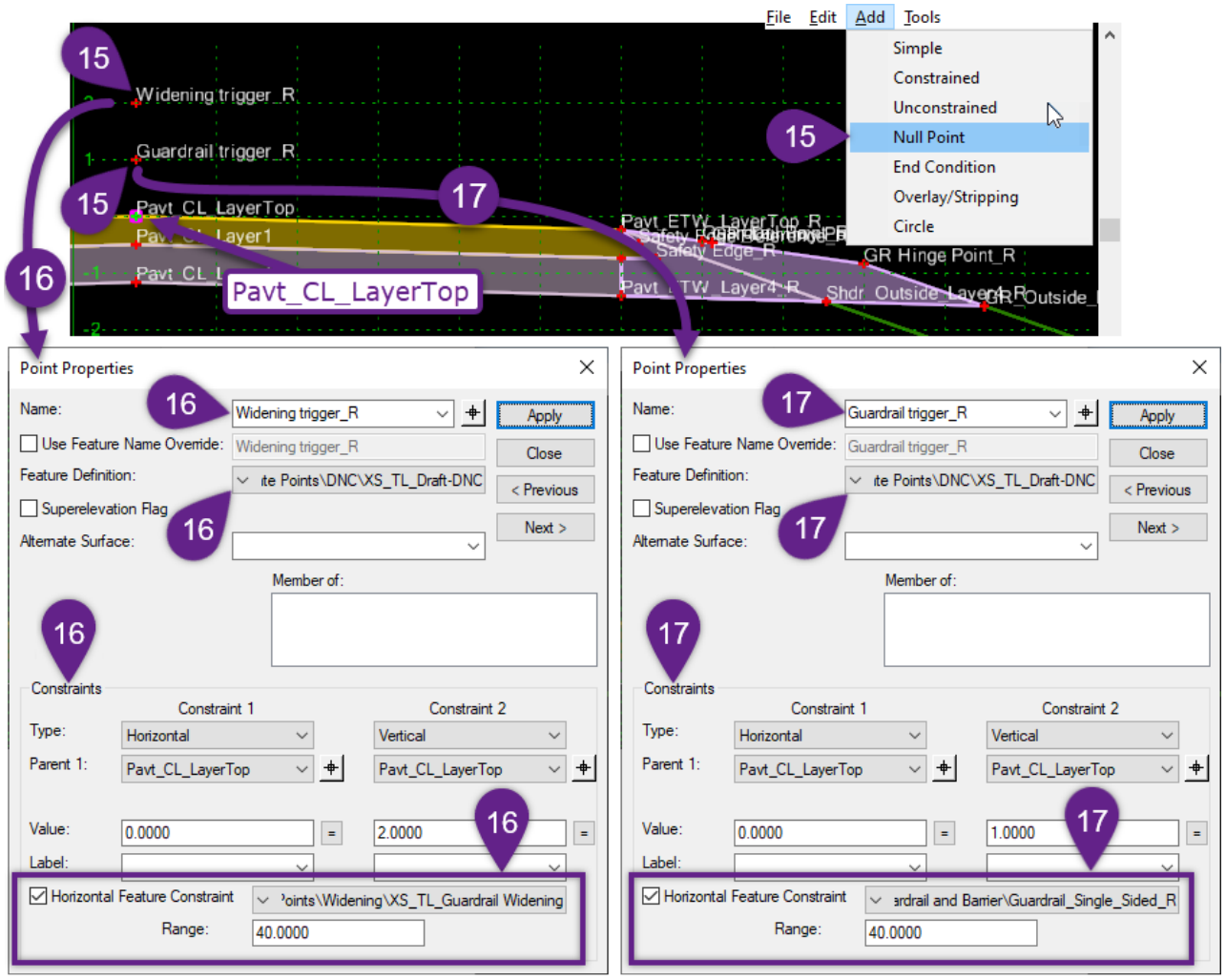
In order to display a Guardrail graphic in the Dynamic Cross Section Viewer and in Cross Section Production, a Point with a Feature Definition under *Guardrail_Single_Sided* (or similar) must be present in the Template. This Point could be inserted in the *GR_Shoulder Widening* component but would not be appropriate because the Widening and Guardrail Conditions need to be triggered separately.

There are some instances in the 2D Design Model (shown on page 8-101) when the *Widening Condition* is necessary, but the *Guardrail Condition* is NOT. To accommodate this condition, a dummy Component is made with a single Guardrail Point to correspond to the Guardrail Line. This dummy Component is conditionally displayed if the Guardrail Line drawn in the 2D Design Model is found (*Guardrail Condition*). This configuration requires a Component as opposed to a single Null Point, because a Null Point can NOT have Display Rules applied to it.

12	Create the *Guardrail Dummy_R* component. Delineate the component shape by clicking in the locations shown above. Only two Points are necessary to define this Dummy Component. The Feature Definition given to this component is *XS_TC_Draft-DNC* – because it is a *Do Not Construct* Feature.
13	Assign the Point a **Name:** *Guardrail Point_R* is used in this example. Assign the Point a **Feature Definition:** *Guardrail_Single_Sided_R* is used this example. Assign the **Point Constraints** as shown. The Vector Offset constraint is used between *Pavt_ETW_LayerTop_R* and *GR Hinge Point_R*. The *Horizontal* constraint value is inconsequential in this example - because the *Horizontal Feature Constraint* is used. Check the **Horizontal Feature Constraint** box and assign the *Guardrail_Single_Sided_R* Feature Definition. This is the same Feature Definition applied to the Horizontal ORD Element shown on page 8-101, called *Guardrail Line.* The horizontal position of this Template Point will follow the ORD Element. The **Range** value for the *Horizontal Feature Constraint* is left at 0.0000. This means the Point will search for the Horizontal Feature (ORD Element) in both directions, until the Feature is found.
14	This is a dummy point and is only necessary because a minimum of 2 Template Points are required to create a Component. The **Name** is set to *GR_Dummy Point_R*. The **Feature Definition** is set to *XS_TL_Draft-DNC* to make it a *Do Not Construct* point. The **Constraints** are inconsequential, but are set to closely follow the *Guardrail Point_R*.

8F.3.d Create Null Points used to trigger Display Rules

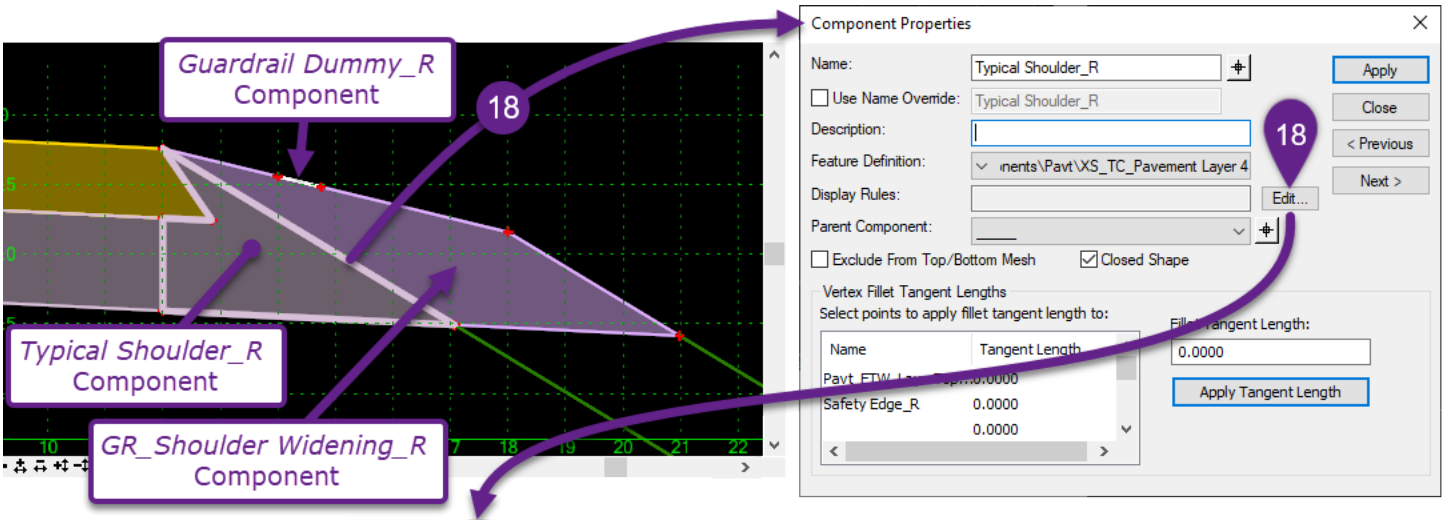
It is required to create two Null Points per side of the Template: one null Point to trigger the *Widening Condition* and another Null Point to trigger the *Guardrail Condition*.



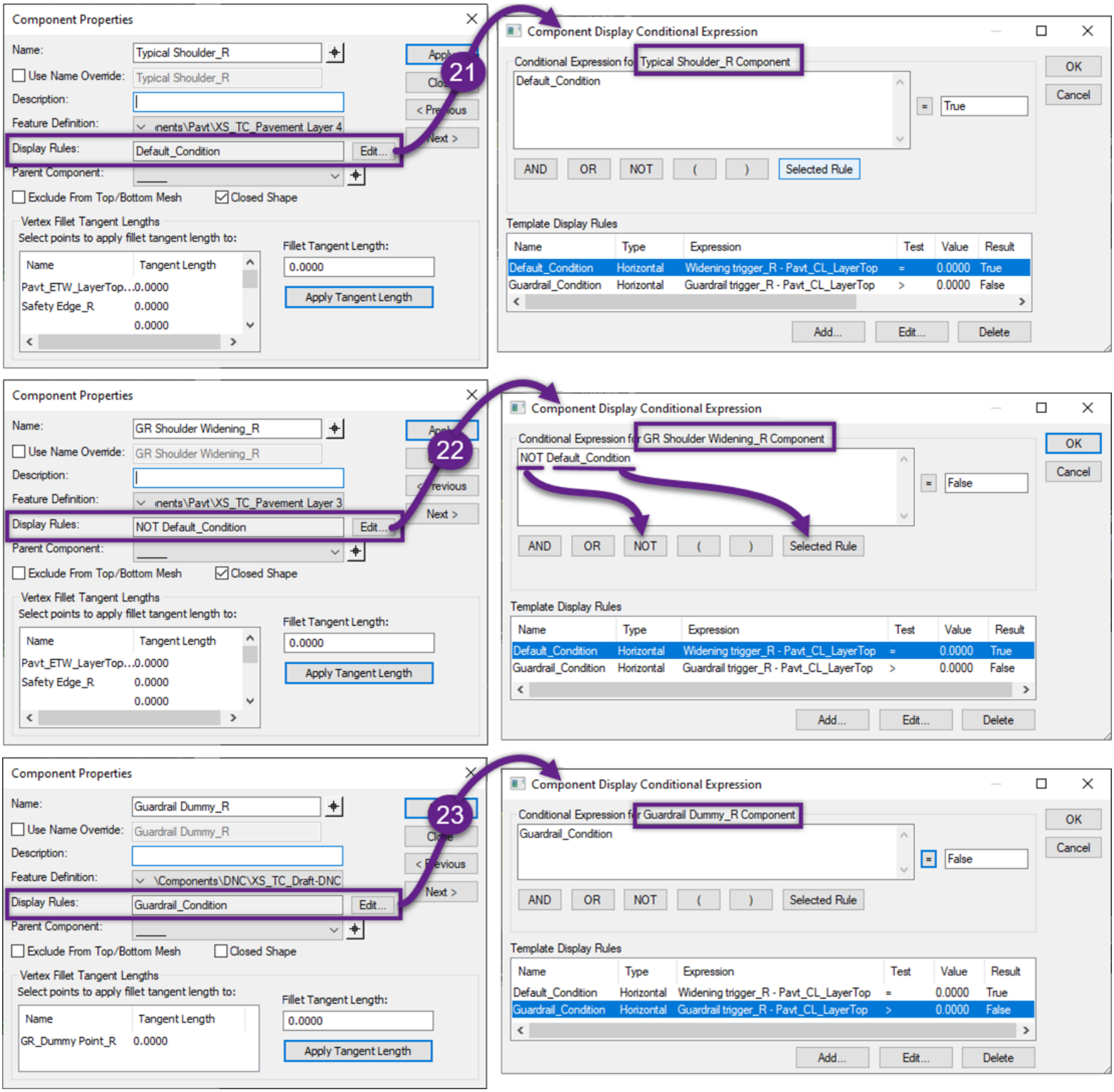
15	Create a Null Point for both the *Widening* and *Guardrail Conditions*.
16	Assign the *Widening Condition* Null Point a **Name:** *Widening trigger_R* is used in this example. **Feature Definition:** *XS_TL_Draft-DNC* is used because this is intended as a *Do Not Construct* point. **Point Constraints**: The Null Point must be positioned *horizontally* in line with the *Pavt_CL_LayerTop* (Parent 1) point. The Horizontal position of the Null Point is what triggers the Display Rules to be set up in the next step. The Vertical constraint is inconsequential. Check the **Horizontal Feature Constraint** box and assign the *XS_TL_Guardrail Widening_R* Feature Definition to correspond with *Widening Hinge Line* shown on page 8-101. The **Range** Value is set to +40.000 to ensure the Horizontal Feature Constraint is searching in the correct direction. When the *Widening Hinge Line* is found in the 2D Model, the Null Point will be moved from the default position (directly over Pavt_CL_LayerTop) which will trigger *Display Rules* set up in the next step.
17	The *Guardrail Condition* Null Point is set up in the same manner as Step 16, but with changes to: **Name:** *Guardrail Trigger_R* **Horizontal Feature Constraint:** Guardrail_Single_Sided_R

8F.3.e Create Display Rules for Default and Guardrail Conditions

18	Via the Component Properties for the *Typical Shoulder_R* component, access the *Display Rules*.
19	Create the *Display Rule* for the *Default Condition* as shown below. The *Default Condition* will be TRUE if the *Widening trigger_R* Null Point remains in its default position, horizontally in line with the *Pavt_CL_LayerTop* point. The *Default Condition* will be FALSE, if the *Widening trigger_R* Null Point is forced out of its default position by the Horizontal Feature Constraint looking for the *Widening Hinge Line* as shown on page 8-101.
20	Create the *Display Rule* for the *Guardrail Condition* as shown below. The *Guardrail Condition* will only be TRUE if the *Guardrail trigger_R* Null Point is forced out of its default position by the Horizontal Feature Constraint looking for the *Guardrail Line* as shown on page 8-101.



21	Apply the *Default Condition* Display Rule to the *Typical Shoulder_R* component. See *8D.2.d Apply Display Rules to a Component*.
22	Apply the NOT *Default Condition* Display Rule to the *GR_Shoulder Widening_R* component. Left-Click on the "NOT" button before Left-Clicking on the "Selected Rule" button. This component will be displayed only if the *Default Condition* is NOT true.
23	Apply the *Guardrail Condition* Display Rule to the *Guardrail Dummy_R* component.



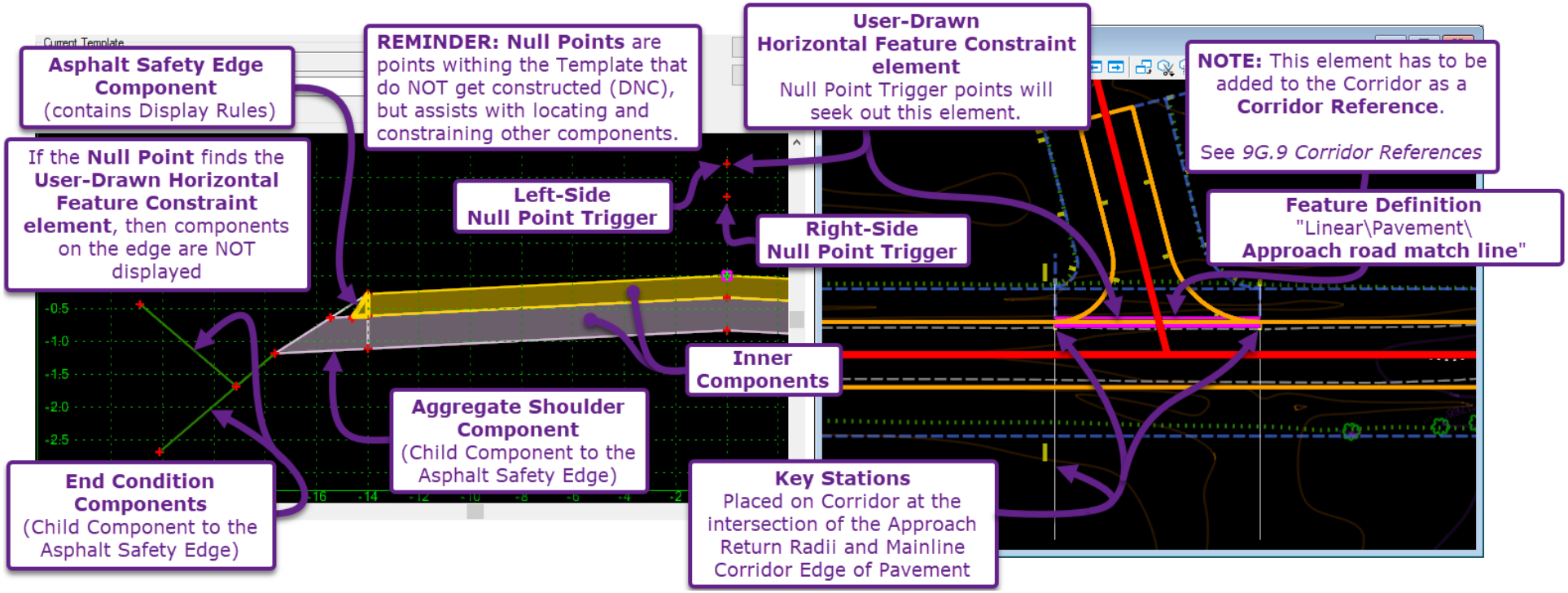
8F.4 Mainline Road Template with Display Rules for Managing Approach Roads and Driveways

Overlap between the Corridor and Intersection, Approaches, and Driveway models can be eliminated by building a Corridor Template with *Display Rules, Horizontal Feature Constraints, Parent Components*, and *Null Points.* In the vicinity of the approach road, the shoulder and end condition components can be removed with *Display Rules*. This method requires advance Template creation techniques but results in a less cluttered model. With this method, the User will NOT need to clip, add additional Template Sections, or create End Condition Exceptions to the Corridor. This method does NOT affect Corridor processing times nearly as much as the aforementioned methods.

Display Rules, *Parent Components, Horizontal Feature Constraints*, and *Null Points* are discussed in detail in the following sections:

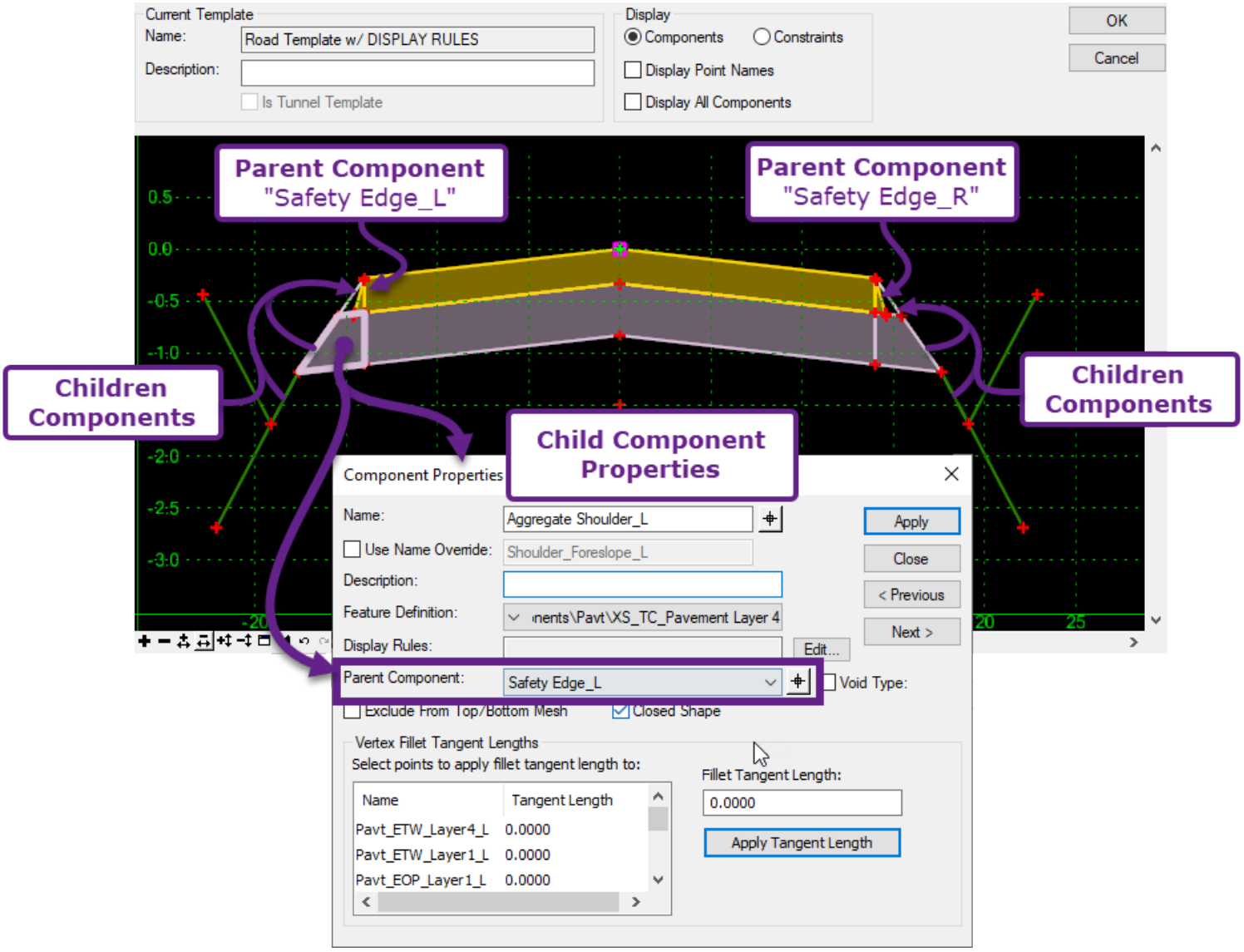
- *8D.1 Component Properties*
- *8C.6.a.xiv Horizontal Feature Constraint*
- *8D.2 Display Rules*
- *8D.3 Parent Components*
- *8F.3 Advanced Road Template with Guardrail and Display Rules*

With this method, the Asphalt Safety Edge and Base Aggregate components must be unattached (but directly adjacent) to the inner asphalt and aggregate components. The Asphalt Safety Edge components at the edges are subjected to Display Rules. The Asphalt Safety Edge components are NOT displayed if the Null Point finds a User-drawn Horizontal Feature Constraint element in the *2D Design Model* . The Aggregate Shoulder components (on the edge) and End Condition (cut/fill) components are Children Components to the Asphalt Safety Edge components (Parent). The Children Components are only displayed if the parent component (Asphalt Safety Edge) is displayed.



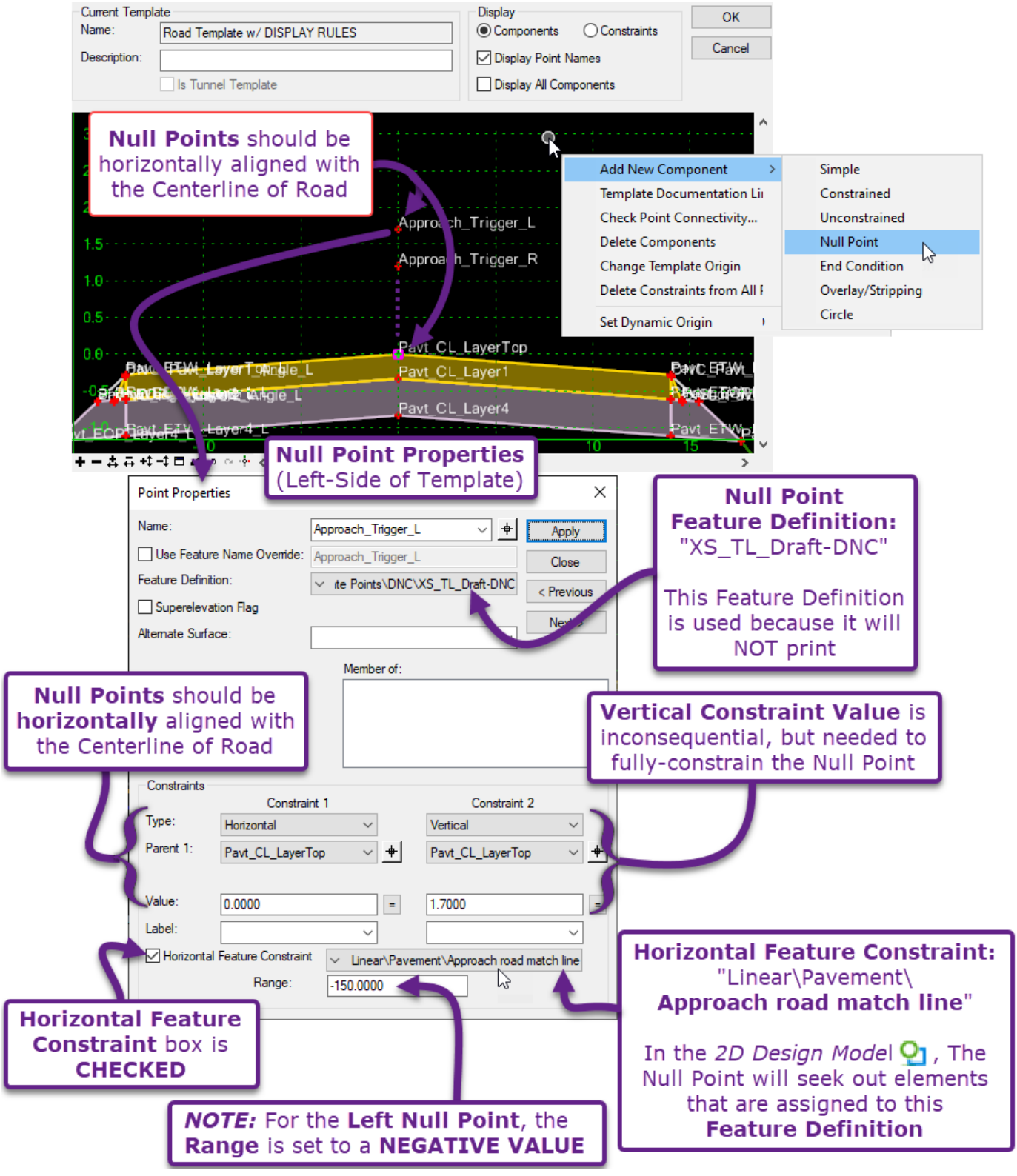
8F.4.a Build and Assemble the Template Components

Using techniques shown in *8E – Creating and Manipulating Points and Components*, create the Template as shown below. The components on the edge must be separate from the inner components. The End Condition and Aggregate Shoulder Components must be *Children* to the Asphalt Safety Edge Component. See *8D.3 Parent Components*.



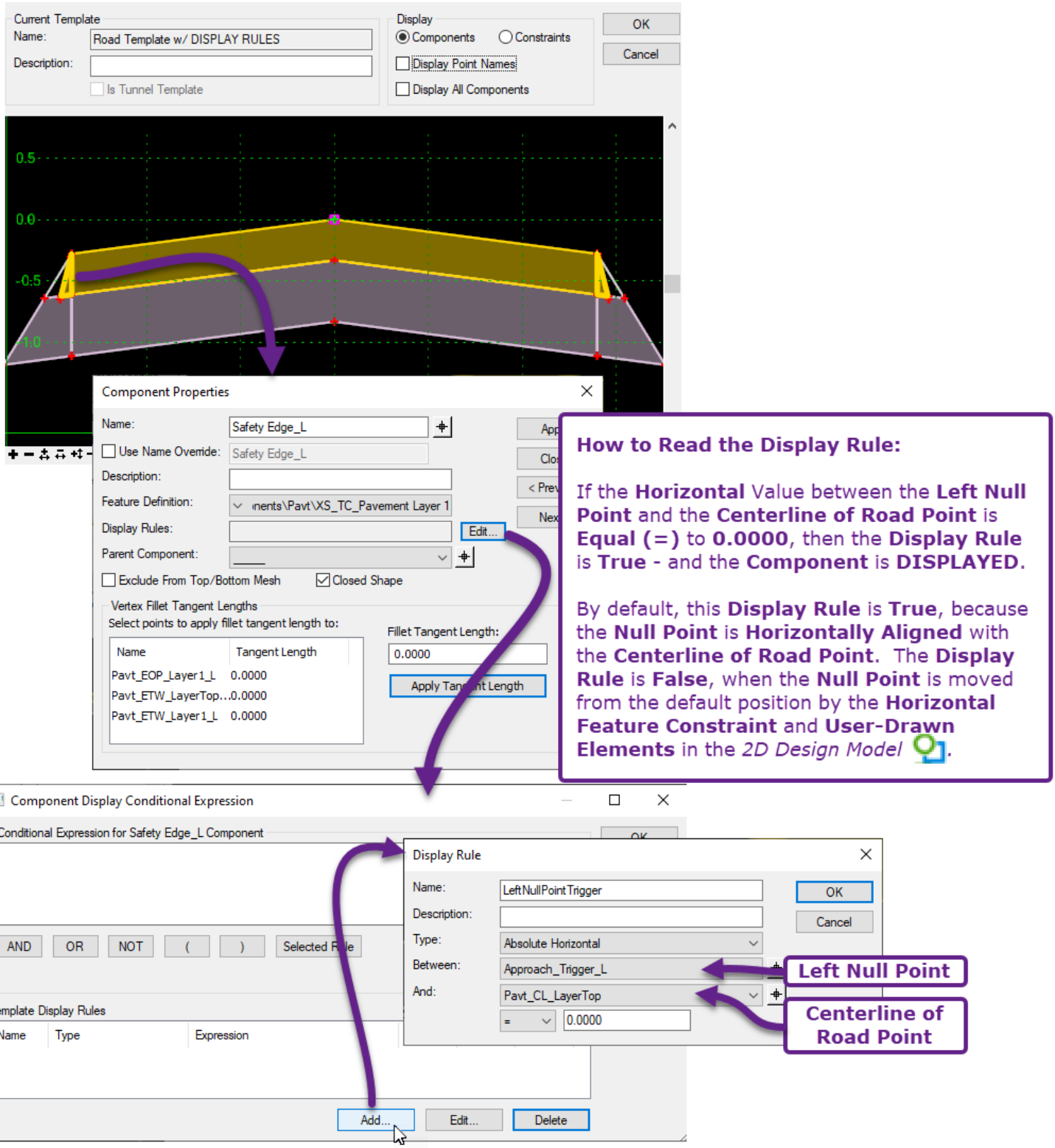
8F.4.b Create the Null Points that contain Horizontal Feature Constraints

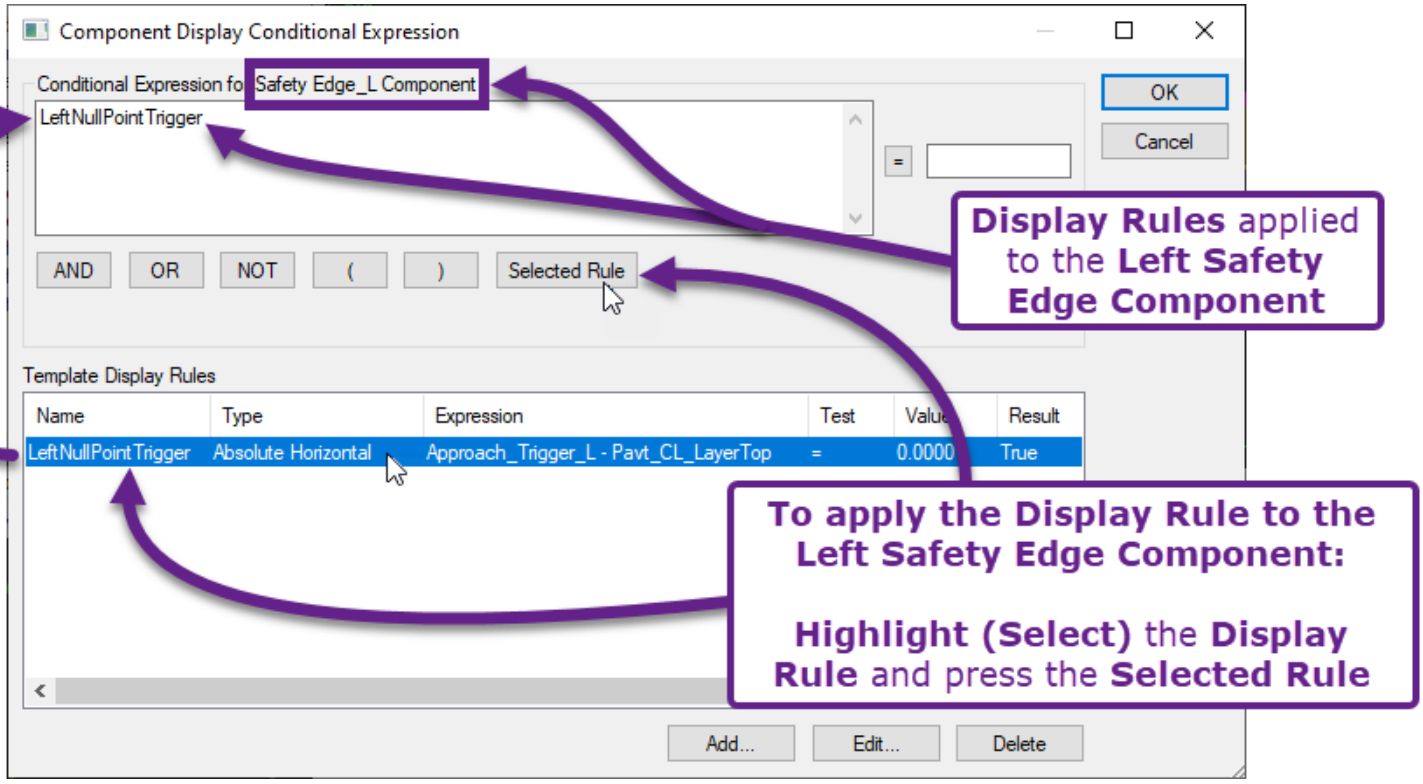
In this step, Null Points are created. A Null Point should be created for each side of the Template. The Null Points must contain Horizontal Feature Constraints – which will trigger the Null Point out of the default position when a User-drawn element is found in the *2D Design Model*. Horizontal Feature Constraints are discussed in detail in *8C.6.a.xiv Horizontal Feature Constraint*. Place the Null Points in vertical alignment with the Centerline point (i.e., Horizontal Constraint = 0).

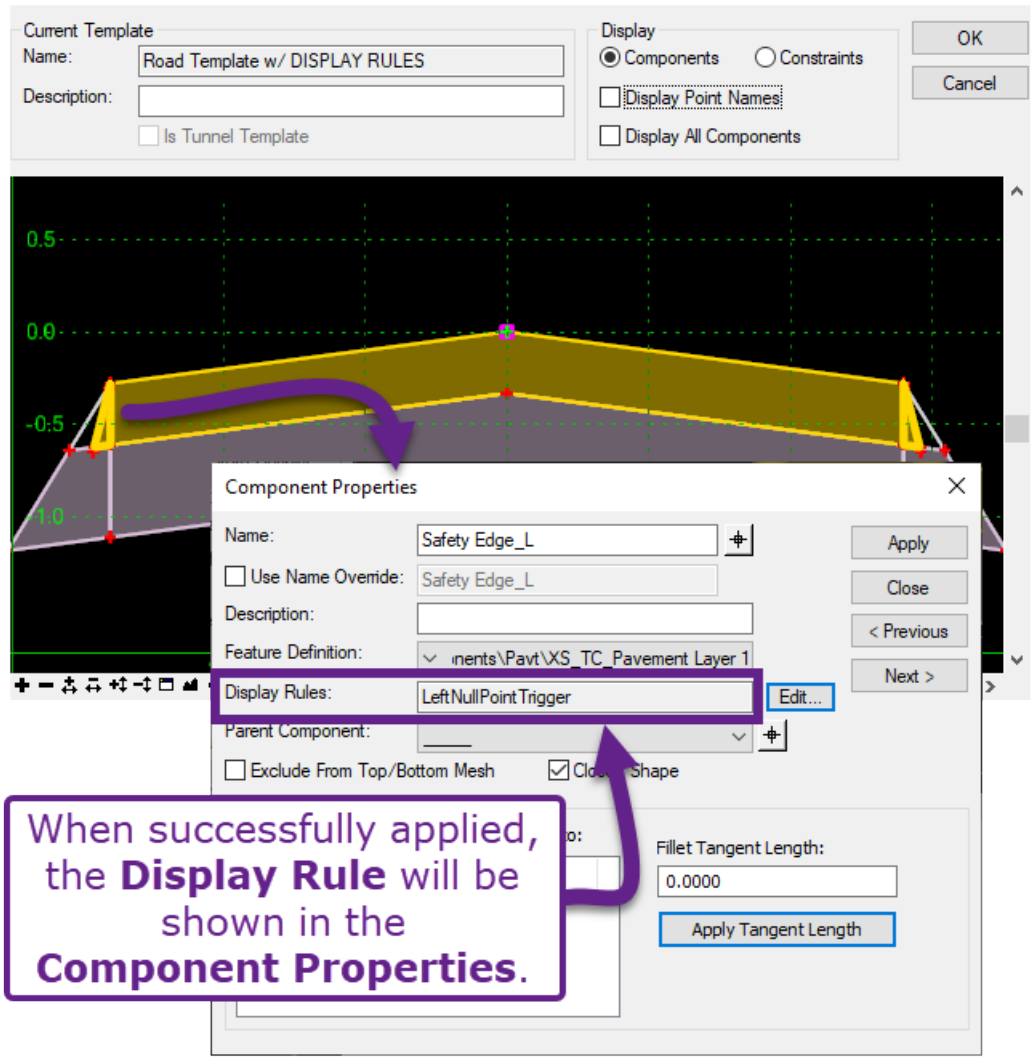


8F.4.c Create Display Rules for the Parent Components

Display Rules must be created and assigned to each of the Parent Components. In this case, the Asphalt Safety Edge Components ("Safety Edge_R" and "Safety Edge_L") will serve as the Parent Components to which Display Rules will be applied. Display Rules are discussed in detail in *8D.2 Display Rules*.




Component Display Conditional Expression
Conditional Expression fo
Safety Edge_L Component
LeftNullPointTrigger
OK
Cancel
=
AND
OR
NOT
(
)
Selected Rule
Display Rules applied to the Left Safety Edge Component
Template Display Rules
Name
Type
Expression
Test
Value
Result
LeftNullPointTrigger
Absolute Horizontal
Approach_Trigger_L - Pavt_CL_LayerTop
=
0.0000
True
To apply the Display Rule to the Left Safety Edge Component:
Highlight (Select) the Display Rule and press the Selected Rule
Add...
Edit...
Delete


Current Template
Name:
Road Template w/ DISPLAY RULES
Description:
Is Tunnel Template
Display
Components
Constraints
Display Point Names
Display All Components
OK
Cancel
0.5
0.0
-0.5
-1.0
Component Properties
Name:
Safety Edge_L
Use Name Override:
Safety Edge_L
Description:
Feature Definition:
nents\Pavt\XS_TC_Pavement Layer 1
Display Rules:
LeftNullPointTrigger
Edit...
Parent Component:
Exclude From Top/Bottom Mesh
Apply
Close
< Previous
Next >
Fillet Tangent Length:
0.0000
Apply Tangent Length
When successfully applied, the Display Rule will be shown in the Component Properties.

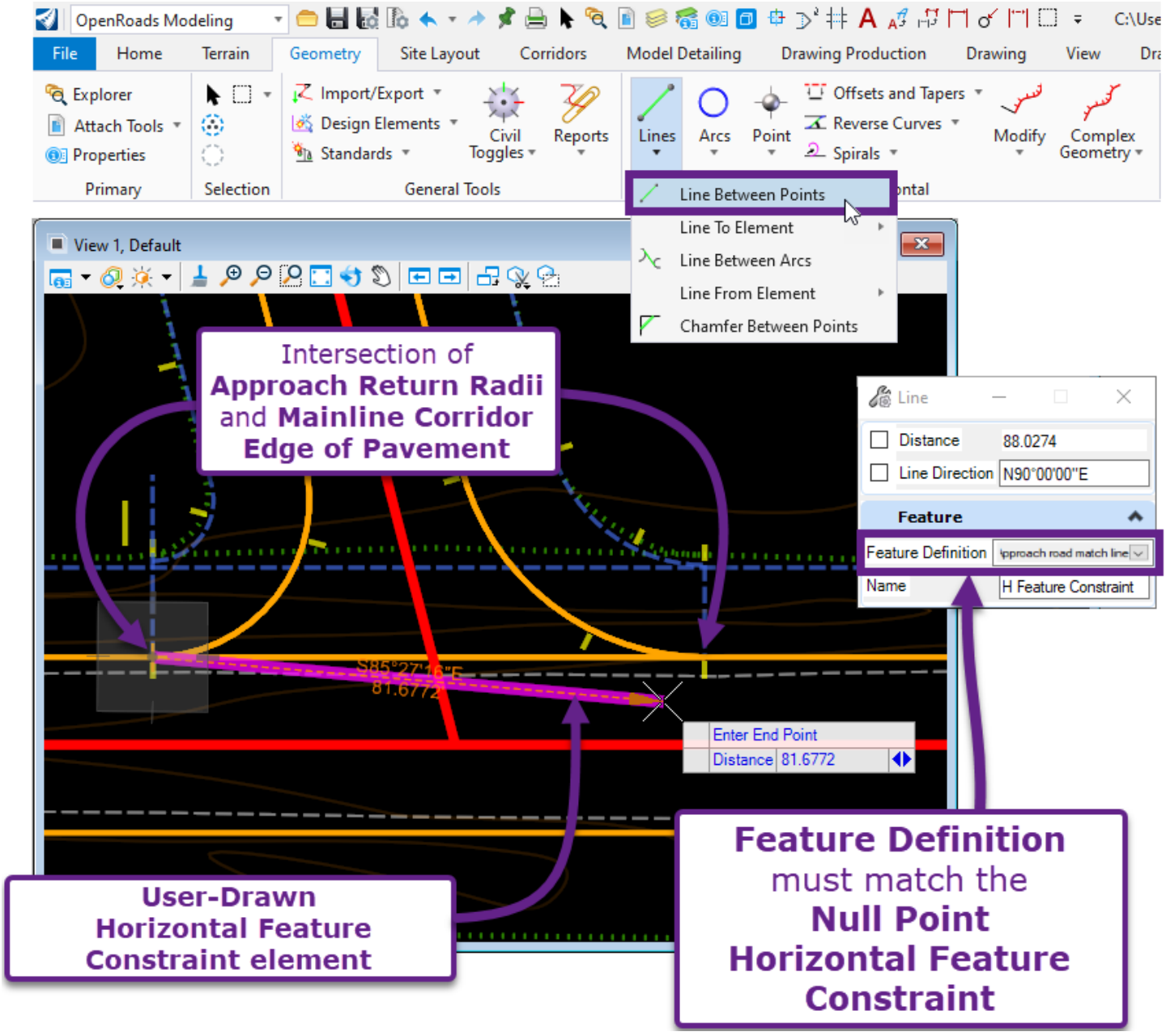
8F.4.d Draw the Horizontal Feature Constraint element in the 2D Design Model

The User-Drawn Horizontal Feature Constraint element represents the location where the Corridor Model and the Approach Site-Model overlap. Typically, the model overlap begins and ends at the point locations where the Approach Return Radii intersect with the Mainline Corridor Edge of Road.

This Horizontal Feature Constraint element displaces the Null Point from its default position, which makes the Display Rule **FALSE** for the overlap range. When the Display Rule is **FALSE**, the *Asphalt Safety Edge* and all Children Components are NOT displayed.

WARNING: The Horizontal Feature Constraint element must be assigned to the SAME Feature Definition that was specified in *8F.4.b Create the Null Points that contain Horizontal Feature Constraints*. In this case, the Feature Definition is "Linear\Pavement**Approach road match line**".

NOTE: Create the Horizontal Feature Constraint element with a Horizontal ORD Line.

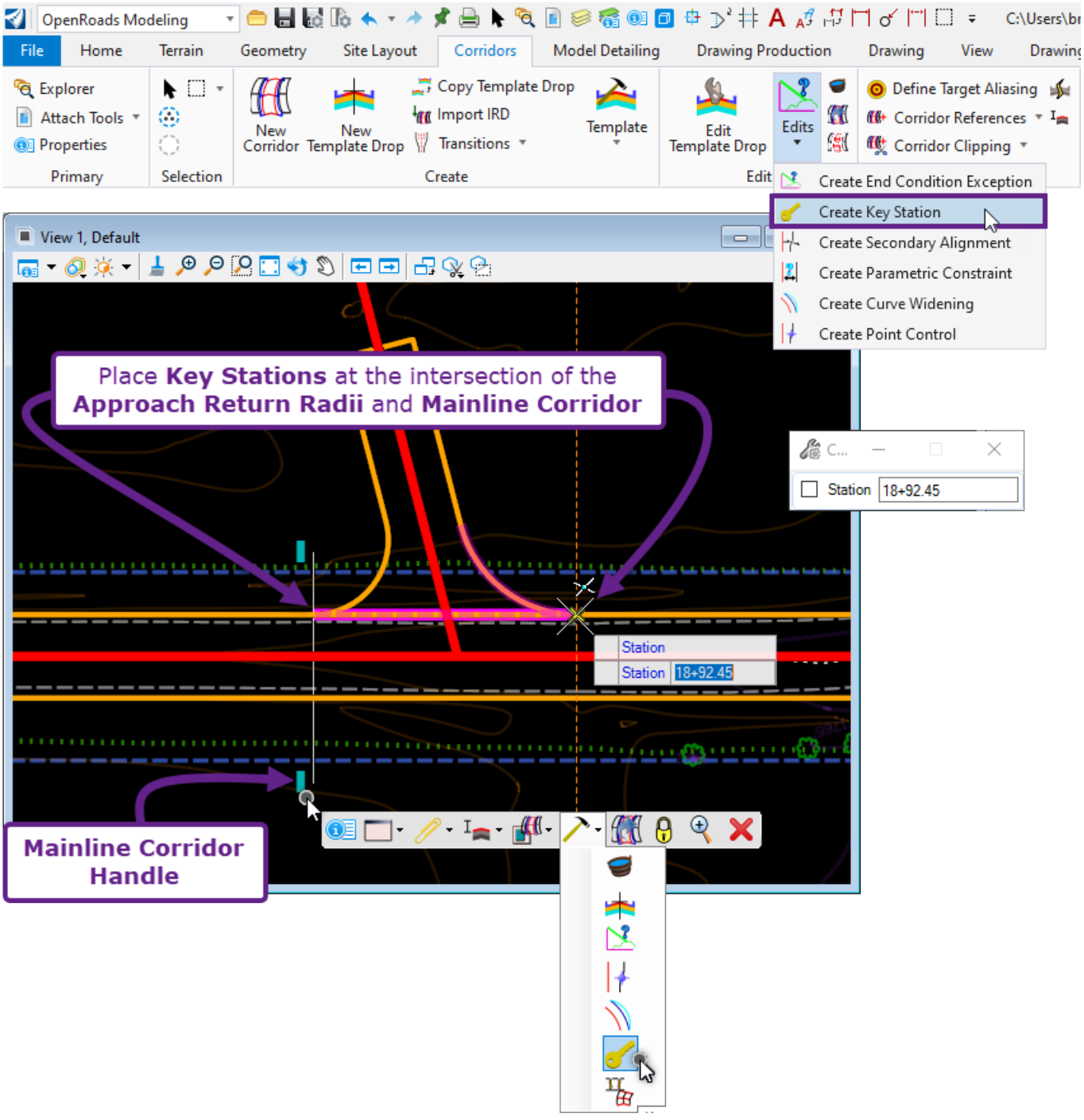


8F.4.e Place Key Stations at the Approach Return Radii

It is necessary to place *Key Stations* on the Mainline Corridor at the intersection with the Approach Return Radii. *Key Stations* will cause the Mainline Corridor to process (place a Template Drop) at these locations. If Key Stations are NOT placed, then the range where the Display Rule is FALSE may overshoot or undershoot the Approach limits.

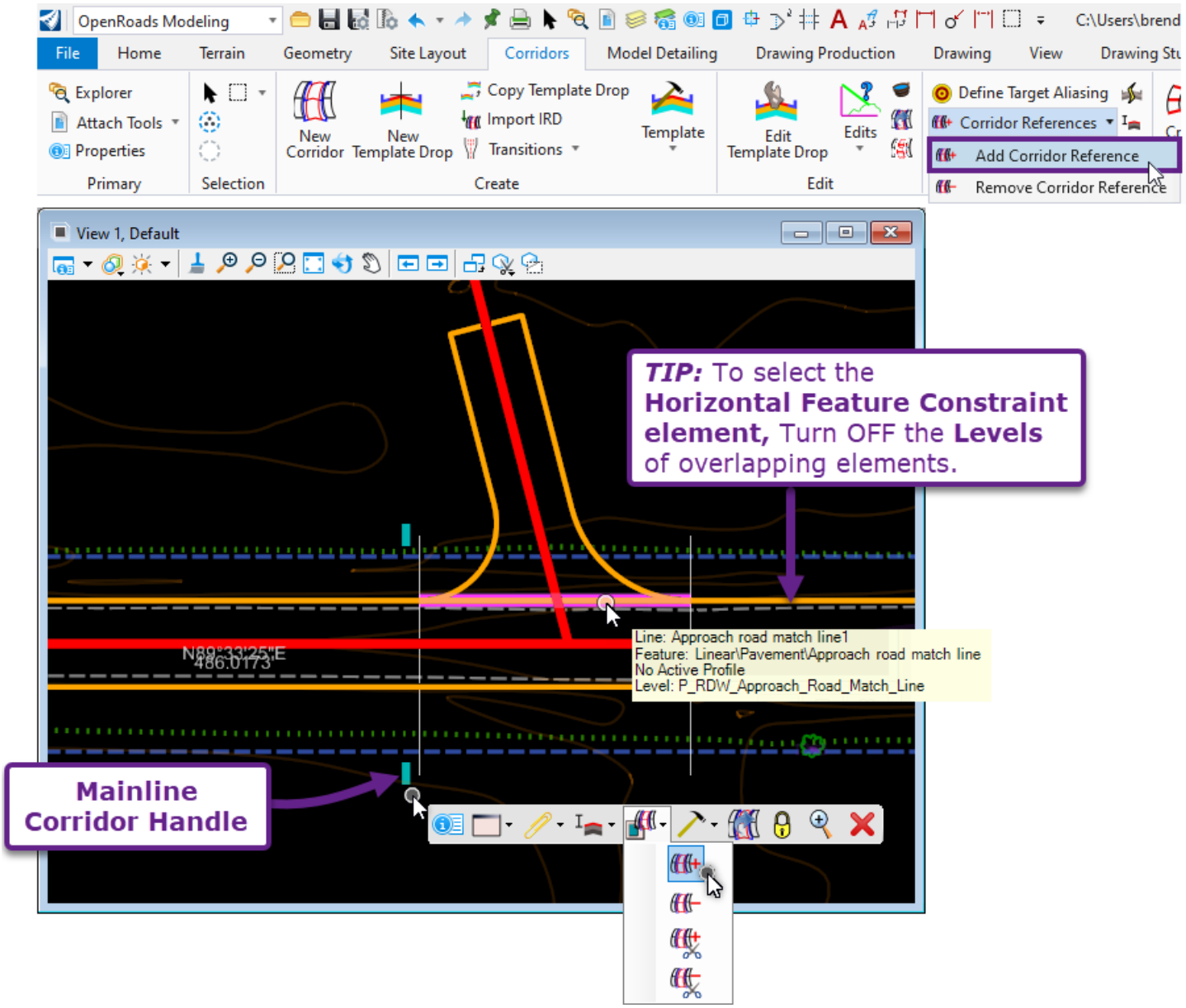
The placement of Key Stations is shown in *9G.3 Key Station*.

TIP: If after several attempts, Key Stations are NOT created at the desired locations, then turn OFF the **Persist Snap** toggle. The Persist Snap toggle is located in the Feature Definition Toolbar. See *7B.3 Feature Definition Toolbar*.



8F.4.f Add the Horizontal Feature Constraint element as a Corridor Reference

For the User-drawn Horizontal Feature Constraint element to be recognized by the Corridor, it must be added as a *Corridor Reference*. The *Add Corridor Reference* tool is discussed in *9G.9 Corridor References*.



8G – SURFACE TEMPLATES

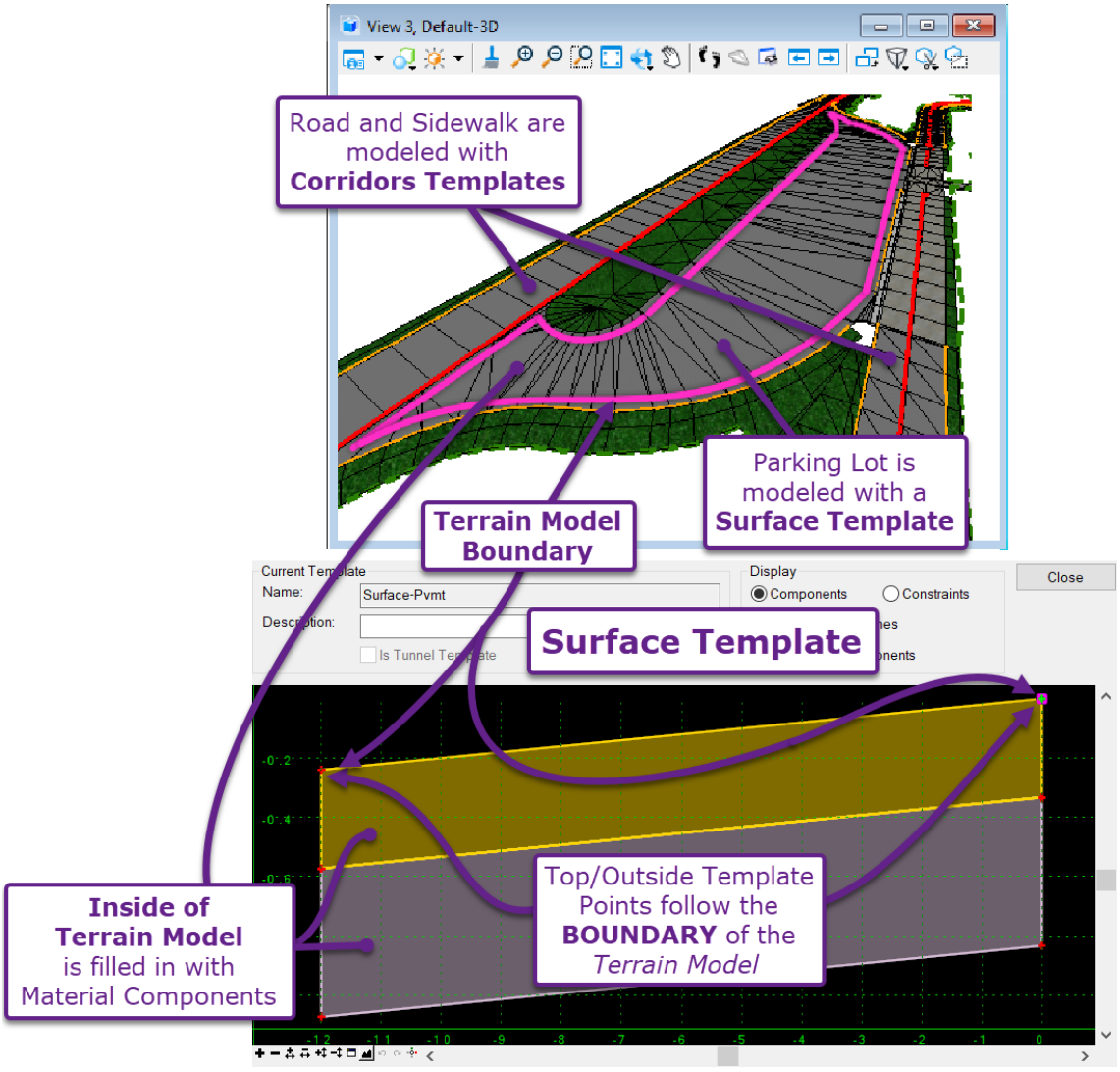
8G.1 Corridor Modeling vs Surface Template Modeling

There are two distinct types modeling in ORD, each requiring a different type of Template:

Corridor Modeling: Corridors use an Alignment, Profile, and Template to create a 3-dimensional model. Corridor Models are used to model features that are generally orientated in a linear path, such as roadways, retaining walls, paved ditches, and culverts. Templates used for Corridor Modeling are akin to a Typical Road Section.

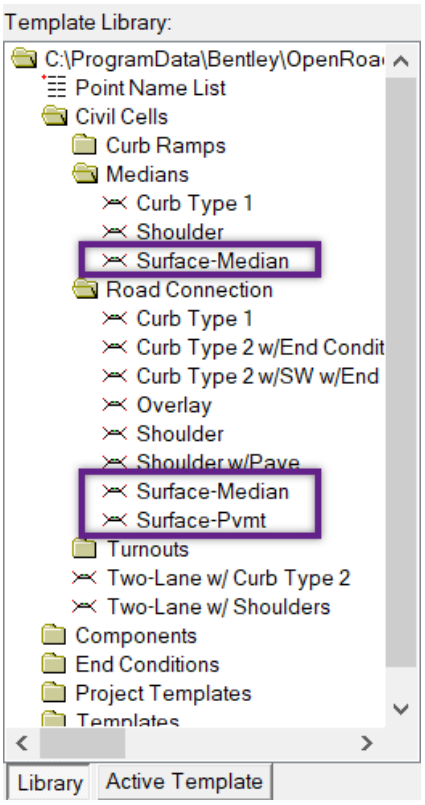
There are two types of Corridor Models: traditional *Corridors* and *Linear Templates*. In general, *Corridor* and *Linear Templates* operate similarly. However, *Linear Templates* are intended for ancillary site features that are linear in nature, such as a curb/sidewalk templates that abuts against a parking lot. *Corridors* are intended for major features, such as roads. For more information, see *9A.2 Civil Models: Corridors vs Linear Templates vs Surface Templates*.

Surface Modeling: Surface Templates are generally used to represent non-linear features – such as a parking lot. Surface Templates are applied to a Terrain Model to model the material underneath the surface of the Terrain Model. For example, a proposed Terrain Model may be created to represent the asphalt surface for a parking lot. A Surface Template is then applied to model the material depths under the surface (i.e., 4" of asphalt and 6" of aggregate). Surface Template creation is discussed in *Chapter 11 – Site Layout*.



8G.2 Surface Template Basics

Surface Templates are simple rectangular-shaped Template Components that are stacked in layers.



Location in Template Library: Pre-made Surface Templates are found in the following locations of the FLH Template Library:

Legacy Templates → Civil Cells → Medians

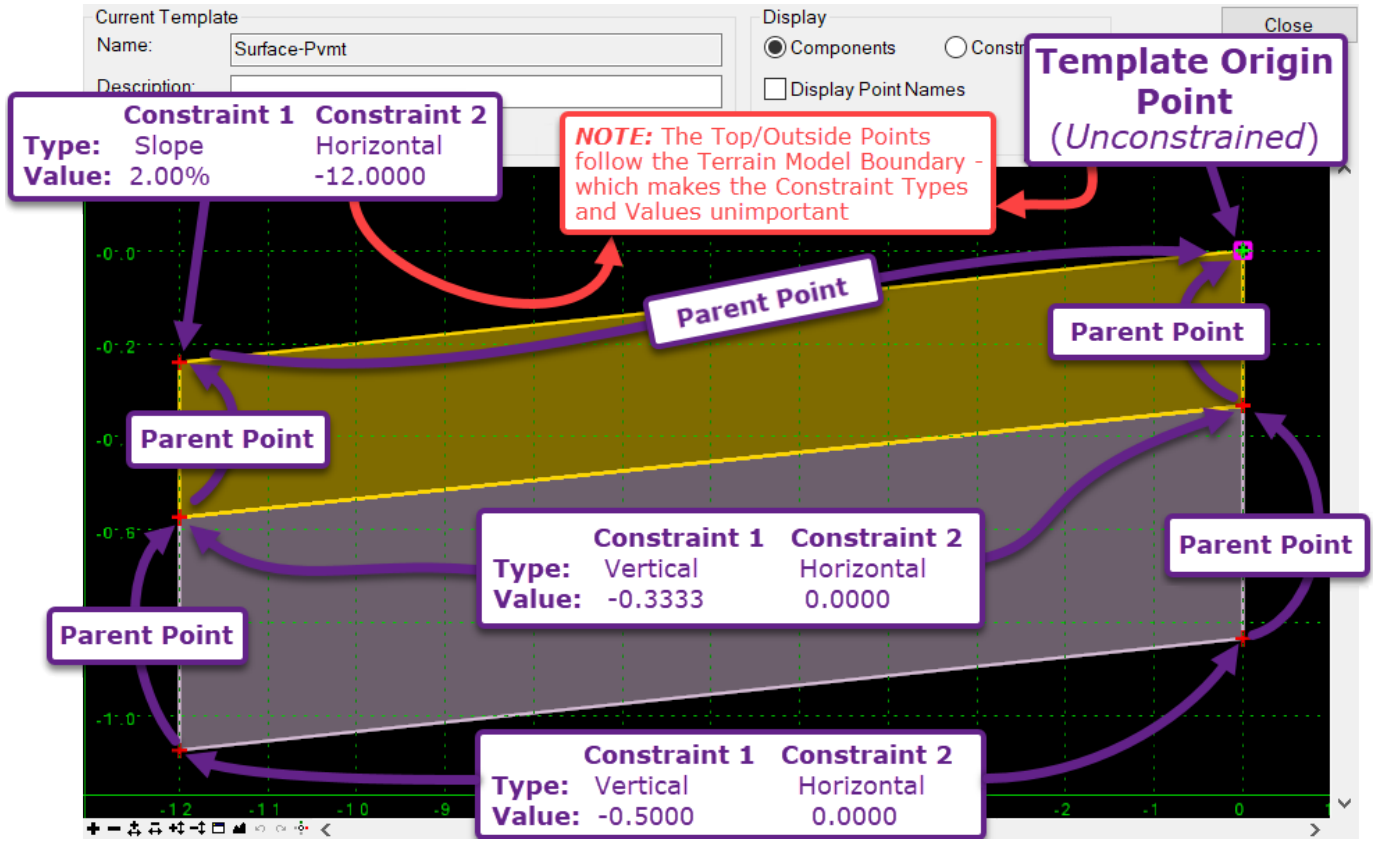
Legacy Templates → Civil Cells → Road Connections

Surface Template Naming: In the FLH Template Library, the *Name* of Surface Templates will have the prefix "Surface –". For example, the Surface Template shown below is named "Surface-Pvmt".

Mechanics of Surface Templates: The Top/Left Point follow the horizontal and vertical position of a Terrain Model Boundary. This point should be UNCONSTRAINED (shown in green). The constraints assigned to the Top/Right point do NOT matter. However, this point should be FULLY CONSTRAINED (shown in red).

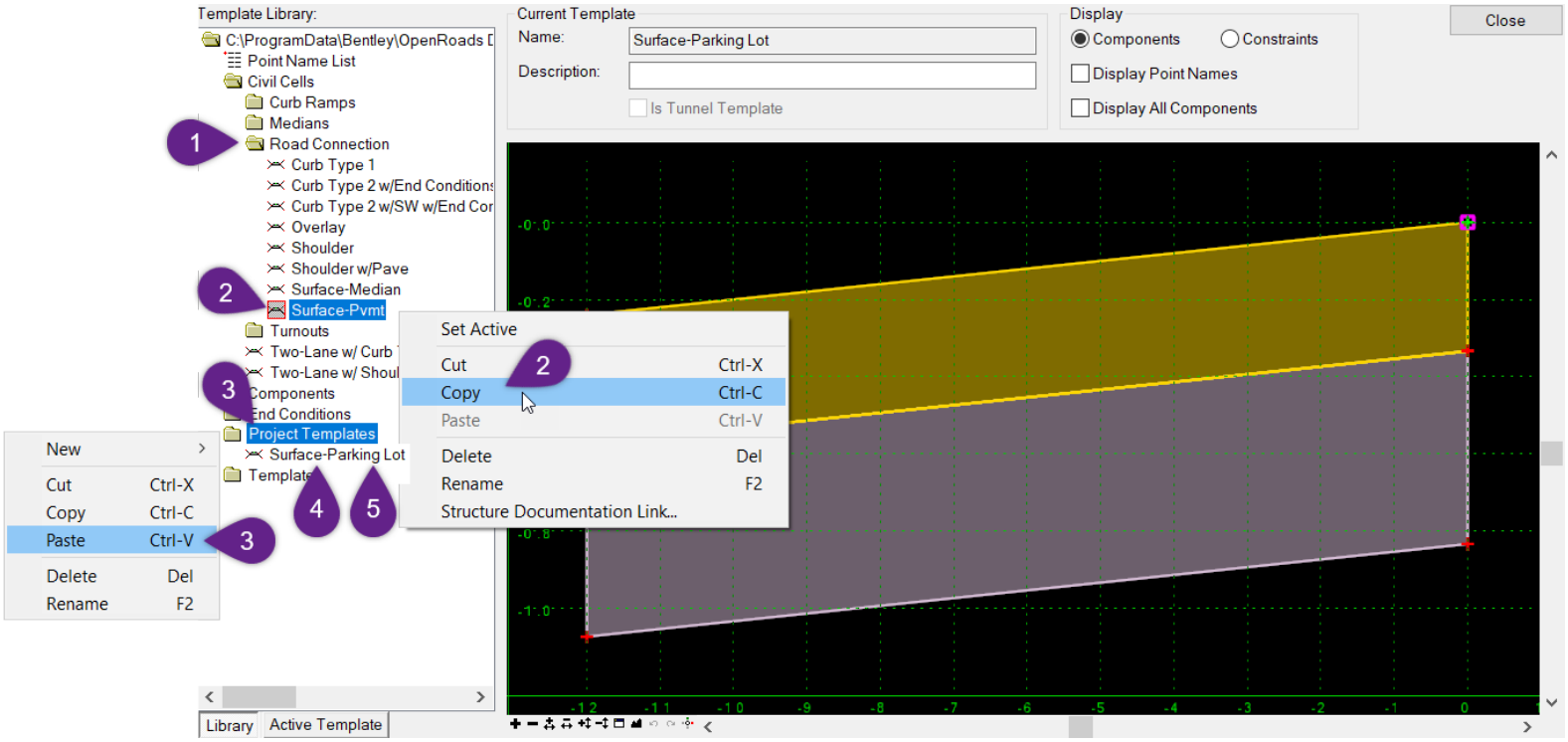
The intermediate and bottom Template Points will need the Horizontal Constraint value set to 0.0000 with the Parent Point being the Template Point immediately above. The Vertical Constraint value is set by the User to specify desired material Component depths.

Creating Surface Templates: Instead of creating from scratch, create a new Surface Template by copying out a pre-made Surface Template and modifying it for project conditions. Be sure to change the Feature Definitions of the Template Points and Components appropriately for the material being modeled.



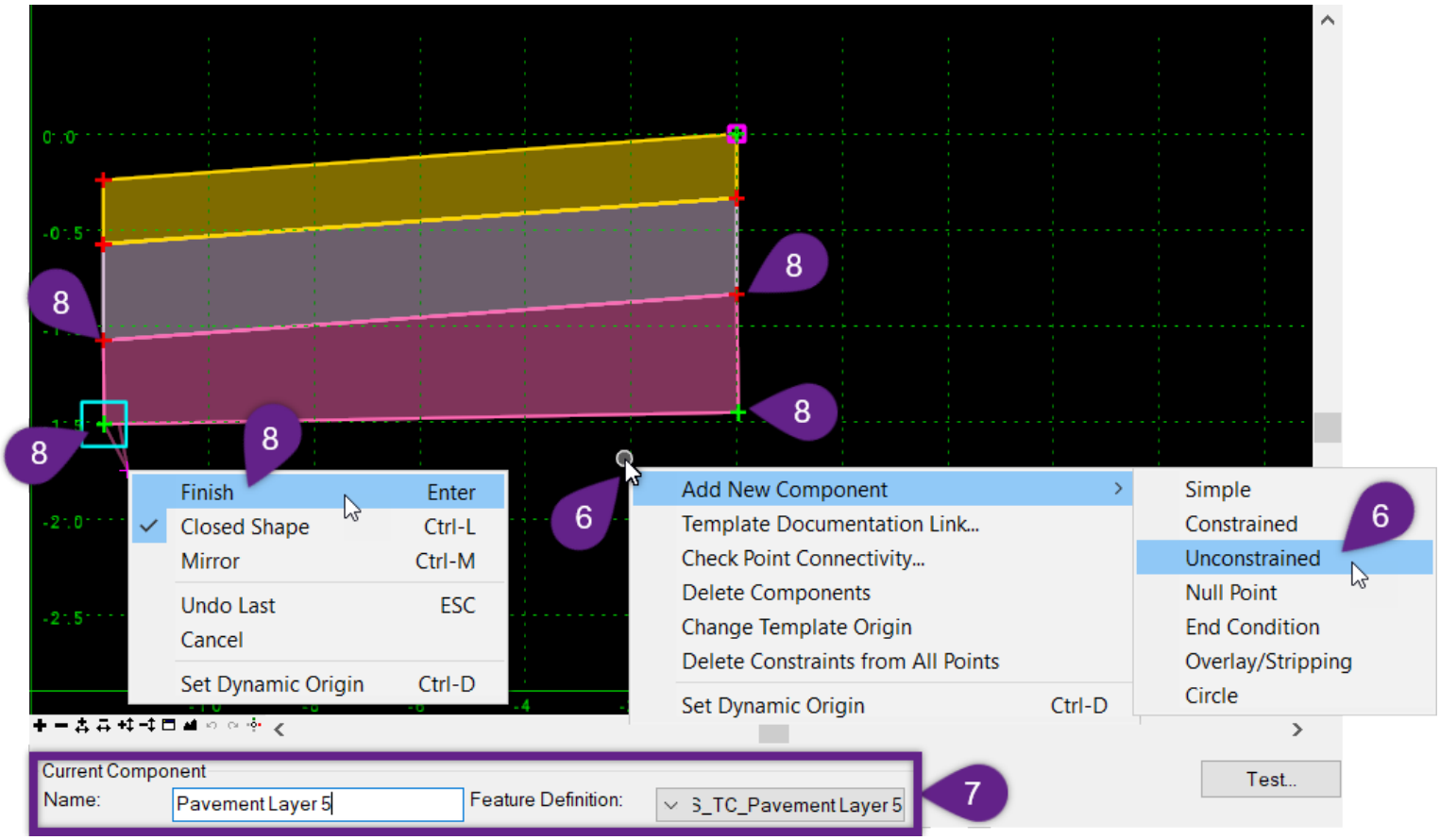
8G.3 Create a Surface Template - Workflow

This workflow demonstrates how to create a new Surface Template by copying and then modifying a pre-made Surface Template from the FLH Template Library. The Surface Template created in this example will represent a parking lot pavement section with three layers to represent a 4" asphalt, 6" base-course aggregate, and 12" sub-base course aggregate.



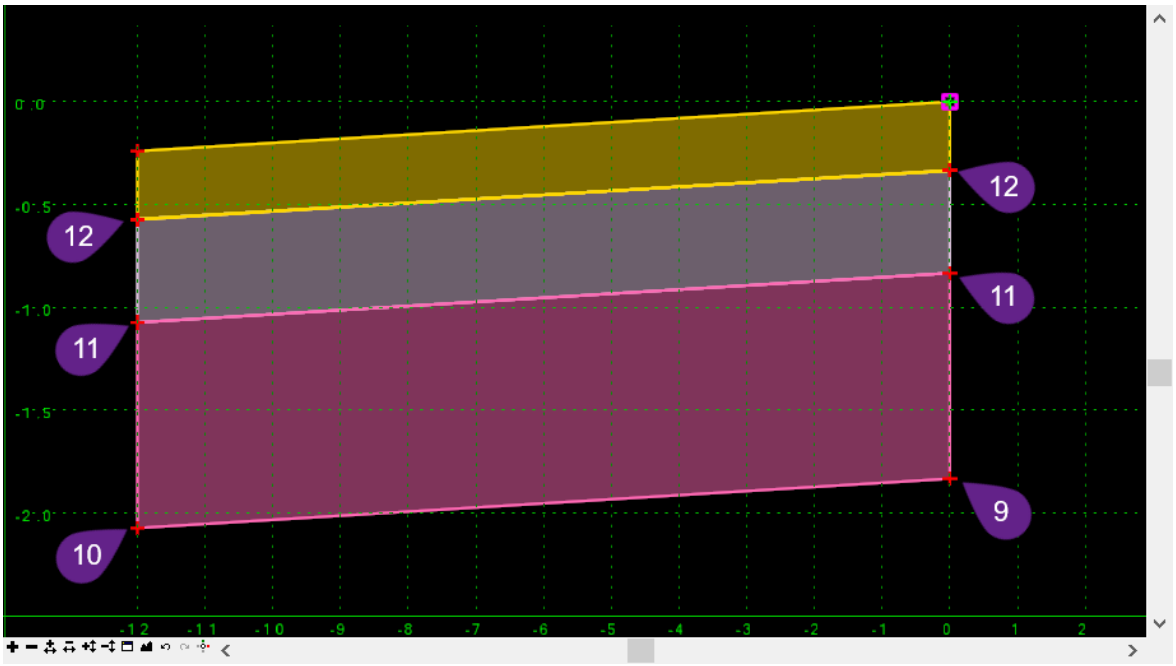
1	With the Project Template Library opened (See *8A.1 Accessing the Template Editor and Template Libraries*), navigate to: *Legacy Templates* → *Civil Cells* → *Medians* OR *Legacy Templates* → *Civil Cells* → *Road Connections*.
2	Locate a pre-made Surface Template that is most similar to the project conditions. In this example, the pre-made Surface Template named "Surface-Pvmt" is used. Right-Click on the pre-made Surface Template and select *Copy*.
3	Right-Click on the *Project Templates* folder and select *Paste*.
4	Right-Click on the pasted Surface Template and select *Rename*. In this example, the Surface Template is renamed to "Surface-Parking Lot".
5	Double-Click on the "Surface-Parking Lot" template to set it as the Active Template.

8G.3.a Create a New Template Component to represent the Sub-base Course:



6	Right-Click in the Active Template Editor Screen and select *Add New Component* → *Unconstrained*.
7	In the boxes below the Active Template Editor Screen, assign the New Component an appropriate *Name* and *Feature Definition*. In this example, the *Name* is set to "Pavement Layer 5" and the *Feature Definition* is set to "XS_TL_Pavement Layer 5".
8	Left-Click on the four locations shown above to place the new Template Component. After placing the last point, Right-Click and select *Finish*.

8G.3.b Set Symbology Properties and Constraints for all Points:



9	Double-Click on the Template Point shown above to access the Point Properties. Set the following: **Name** = Pavt_CL_Layer5 **Feature Definition** = XS_TL_Subgrade **Constraint 1** **Type** = Vertical **Parent 1 =** Pavt_CL_Layer5 **Offset Value =** 1.0000' **Constraint 2** **Type** = Horizontal **Parent 1** = Pavt_CL_Layer5 **Offset Value** = 0.0000'
10	Double-Click on the Template Point shown above to access the Point Properties. Set the following: **Name** = Pavt_ETW_Layer5 **Feature Definition** = XS_TL_Subgrade **Constraint 1** **Type** = Vertical **Parent 1 =** Pavt_ETW_Layer4 **Offset Value =** 1.0000' **Constraint 2** **Type** = Horizontal **Parent 1** = Pavt_ETW_Layer4 **Offset Value** = 0.0000'
11	Double-Click on the TWO Template Points shown above to access the Point Properties. Change the Vertical Constraint Value to 0.5000' Also, change the *Feature Definition* from *XS_TL_Subgrade* to *XS_TL_Edge of Pavt 4*. This is necessary because these points no longer correspond with the subgrade datum of the pavement section.
12	Double-Click on the TWO Template Points shown above to access the Point Properties. Change the Vertical Constraint Value to 0.5000'