AUTODESK UNIVERSITY

CS21034

Using Reusable Subassemblies

Eric Cylwik
Sundt Construction, Inc.

Learning Objectives

- Understand subassembly parameter passing
- Understand custom codes and styles
- Learn how to create dynamic assemblies
- Learn how to generate quantity reports from corridor assemblies

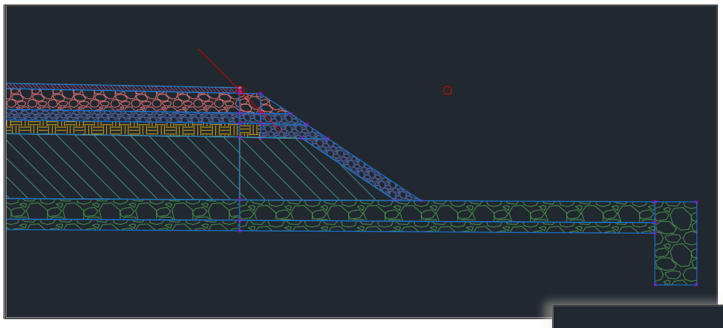
Description

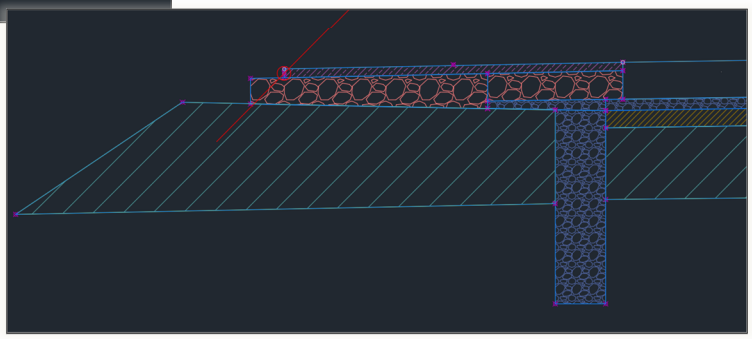
The AutoCAD Civil 3D software corridor is one of the most powerful tools for modeling civil infrastructure projects. While often times it is also viewed as the most complex tool, utilizing reusable subassemblies enables users to harness the power, standardize workflows, and reduce complexity of modeling tasks. This class will demonstrate how to use reusable subassemblies (See CI11634 from Autodesk University 2015) to create common roadway components and utilize them on different projects with different requirements. After creating the assemblies and corridors, we'll explore how to create quantity reports and 3D surfaces for use in other software packages. This session features AutoCAD Civil 3D.

Your AU Expert

Eric Cylwik is the modeling engineer for Sundt Construction's Transportation and Infrastructure division. Before working exclusively for the Heavy Civil division, Cylwik focused on adapting Building Information Modeling (BIM) models from the office to the field for Sundt's Concrete Group. He now focuses on creating virtual construction models that highlight technology’s capability to enhance the way construction is performed in the field for horizontal construction. Focusing on infrastructure, Cylwik has been able to capitalize on parametric modeling to create construction-quality bridge, road, and trench models that are used for survey surfaces, machine control, quantity takeoffs, utility coordination, constructability reviews, and visualizations. Cylwik has helped Sundt procure over $1 billion of alternative delivery method projects. He graduated from Arizona State University with a degree in design studies with an emphasis in digital visualization. He is also a certified professional in several vertical and horizontal BIM and virtual design and construction software programs.

Review of last year's class





Highly parametric shapes

Parameter passing

Parameterize codes for points, links, and shapes

Quadrilateral

Math Operator

Things learned through frustration

Copy/Mirror Functions

- Select logical connections/groups. Using the Assembly Properties Window Construct Tab, all assemblies have to be adjacent. You can also view the logical order in the prospector tab > assemblies > assembly name.
- Mirroring is only enabled when all of the current selection is similarly sided subassemblies (all right, or all left side)
- No-Sided subassemblies cannot be mirrored. Copying is the only way to duplicate no-sided subassemblies.
- After mirroring or copying one needs to check logic and execution of new components. See the recorded class for an example of how the logic and order can change during a copy or mirror operation.
- Update Subassembly Names!

  If there are two subassemblies with the same name, then passing any parameters becomes confusing to Civil 3D. It will select the first subassembly instance with that name and pass that parameter and ignore any after that.

Insert Function

- Before or after options

Before: Civil 3D will insert the new subassembly BEFORE the subassembly you select. There are many instances in which Civil 3D inserts the subassembly not just before that subassembly executes, but before 2 or 3 ahead of it. That functionality makes it difficult to insert math operators at the correct location
After: Civil 3D will insert the new subassembly immediately AFTER the selected subassembly. This seems to execute more consistently as far as placement is concerned, but sometimes messes up conditionality of subsequent subassemblies.

- **Same checks apply – verify logic and execution order**

Updating Subassemblies

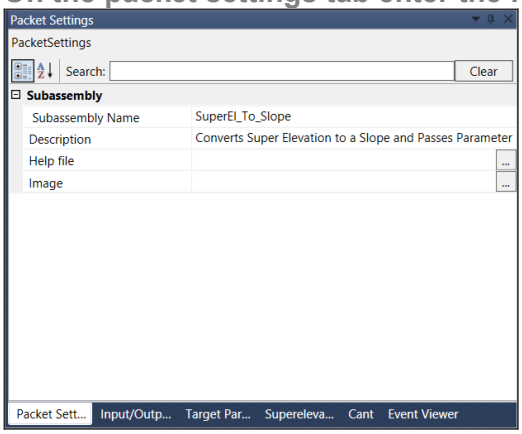
- **No packet name changes**
- **No parameter name changes, removals, or additions**
- **You may change formulas**
- **You may add/change variables and Target Parameters**
- **You may adjust the order of your build**
- **You may adjust defaults**
- **You may add new geometry**
- **Steps to update**
    1. *Open subassembly in Subassembly Composer*
    2. *Make allowable edits *see list above**
    3. *Save file over old .PKT file*
    4. *Switch to Civil 3D*
    5. *Delete old subassembly from tool palette*
    6. *Close civil 3D completely*
    7. *Open Civil 3D*
    8. *In the brand new drawing *drawing1 that is automatically open* import updated subassembly PKT to tool palette*
    9. *Insert subassembly in to current drawing as a detached subassembly*
    10. *Open desired drawing*
    11. *Adjust parameter in updated subassembly/assembly*
    12. *Updated geometry/etc… magically is applied on rebuild*

Super Elevation to Slope subassembly

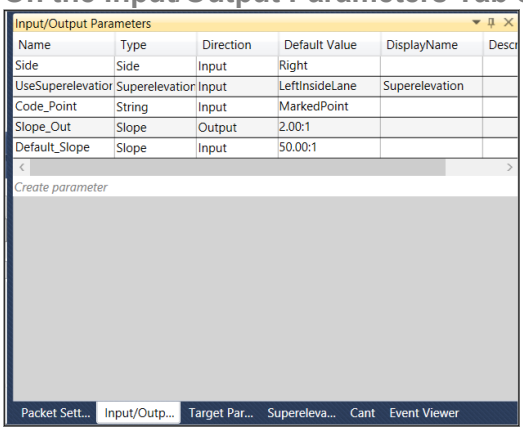
Simplifies super elevation use – common passable parameter. You can select the super elevation once in this subassembly and then pass it to parameters that do not support super elevation.

1. **Open Subassembly Composer**

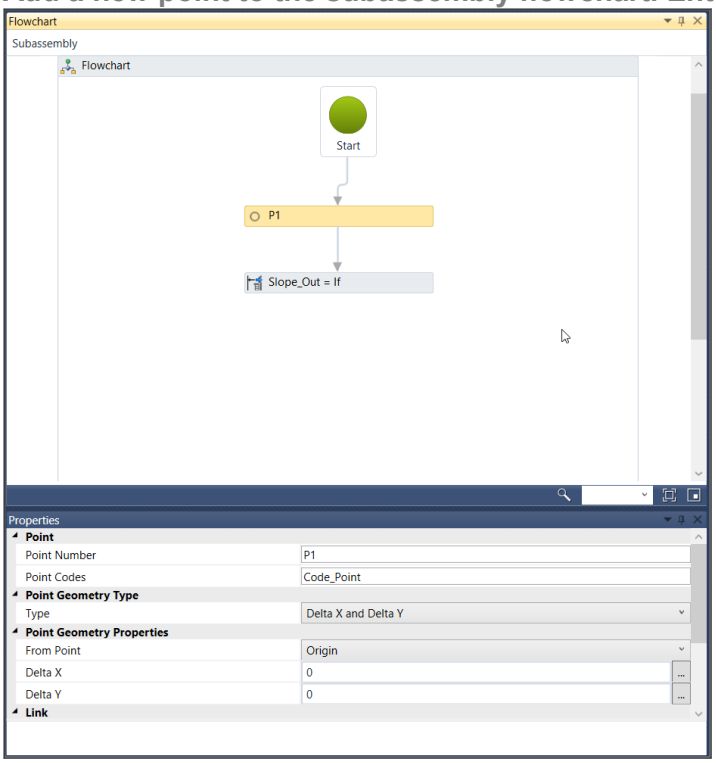
2. **On the packet settings tab enter the following:**



3. **On the Input/Output Parameters Tab enter the following:**



4. **Add a new point to the subassembly flowchart. Enter the following options:**

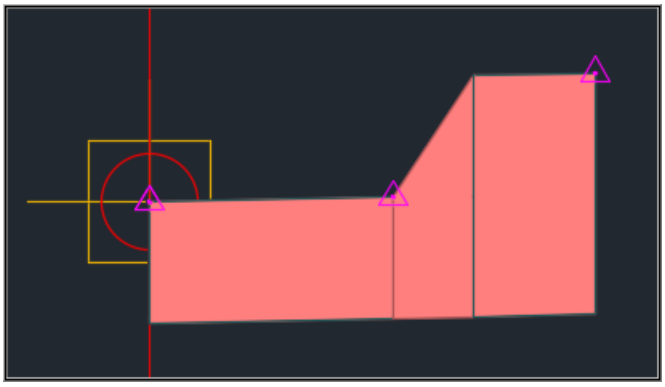


5. **Add the "Set Output Parameters" node, set it to Slope_Out, and then in the formula paste:**

```
If(SA.IsLayout,Convert.ToDouble(Default_Slope),
If(UseSuperelevation=LeftInsideLane,If(SE.HasLeftLI,SE.LeftLI,Convert.ToDouble(Default_Slope)),
If(UseSuperelevation=LeftOutsideLane,If(SE.HasLeftLO,SE.LeftLO,Convert.ToDouble(Default_Slope)),
If(UseSuperelevation=LeftInsideShoulder,If(SE.HasLeftSI,SE.LeftSI,Convert.ToDouble(Default_Slope)),
If(UseSuperelevation=LeftOutsideShoulder,If(SE.HasLeftSO,SE.LeftSO,Convert.ToDouble(Default_Slope)),
If(UseSuperelevation=RightInsideLane,If(SE.HasRightLI,SE.RightLI,Convert.ToDouble(Default_Slope)),
If(UseSuperelevation=RightOutsideLane,If(SE.HasRightLO,SE.RightLO,Convert.ToDouble(Default_Slope)),
If(UseSuperelevation=RightInsideShoulder,If(SE.HasRightSI,SE.RightSI,Convert.ToDouble(Default_Slope)),
If(UseSuperelevation=RightOutsideShoulder,If(SE.HasRightSO,SE.RightSO,Convert.ToDouble(Default_Slope)),Con
vert.ToDouble(Default_Slope)))))))))
```

6. **Save the subassembly as "2016-SuperEltoSlope.pkt"**

Sloping Curb Assembly

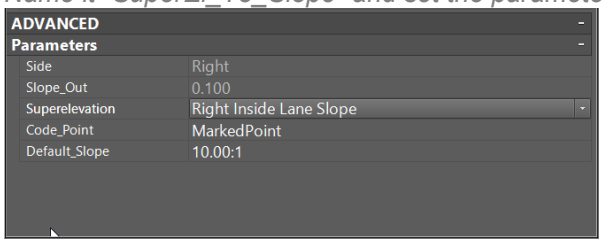


Ingredients:

- (1) Super Elevation to Slope
- (3) Quadrilaterals
- (3) MarkPoints

Steps to create sloping curb assembly:

- Place new assembly
- Place Super Elevation To Slope subassembly on assembly.
  Name it "SuperEl_To_Slope" and set the parameters as below:



- Place 3 quadrilaterals. The first quadrilateral should be placed on the SuperEl_To_Slope subassembly. The second two attaching to top outside point of previous quadrilateral
- Name the recently placed quadrilaterals the following (inside to out) Gutter, Lip, and Top

Gutter settings:

ADVANCED	-
Parameters	-
Side	Right
Out_BottomSlope	0.100
Out_BottomWidth	1.000
Out_InsideHeight	0.500
Out_OutWidth	0.000
Out_OutsideHeight	0.500
Out_TopSlope	0.100
Out_TopWidth	1.000
Top_Slope	10.00:1
PointsCode	_NoDisplay
LinksCode	Pave
Inside_Slope	0.00:1
Outside_Slope	0.00:1
Top_Width	1.00'
Inside_Height	0.50'
Bottom_Slope	10.00:1
ShapeCode	Pave

Lip Settings:

ADVANCED	-
Parameters	-
Side	Right
Out_BottomSlope	0.100
Out_BottomWidth	0.330
Out_InsideHeight	0.500
Out_OutWidth	0.000
Out_OutsideHeight	0.967
Out_TopSlope	1.515
Out_TopWidth	0.330
Top_Slope	0.66:1
PointsCode	_NoDisplay
LinksCode	_NoDisplay
Inside_Slope	0.00:1
Outside_Slope	0.00:1
Top_Width	0.33'
Inside_Height	0.50'
Bottom_Slope	10.00:1
ShapeCode	Pave

Top Settings:

Parameters	-
Side	Right
Out_BottomSlope	0.100
Out_BottomWidth	0.500
Out_InsideHeight	0.967
Out_OutWidth	0.000
Out_OutsideHeight	0.967
Out_TopSlope	0.100
Out_TopWidth	0.500
Top_Slope	10.00:1
PointsCode	_NoDisplay
LinksCode	Pave
Inside_Slope	0.00:1
Outside_Slope	0.00:1
Top_Width	0.50'
Inside_Height	0.97'
Bottom_Slope	10.00:1
ShapeCode	Pave

- **Now that the subassemblies parameters have been set appropriately, set the parameters to pass as shown below:**

Gutter:

Value Name	Default Input Value	Parameter Reference	
		Use	Get Value From
Side	Right	☐	<None>
Top_Slope	10.00:1	☑	SuperEl_To_Slope.Slope_Out
PointsCode	_NoDisplay	☐	<None>
LinksCode	Pave	☐	<None>
Inside_Slope	0.00:1	☐	<None>
Outside_Slope	0.00:1	☐	<None>
Top_Width	1.00'	☐	<None>
Inside_Height	0.50'	☐	<None>
Bottom_Slope	10.00:1	☑	SuperEl_To_Slope.Slope_Out
ShapeCode	Pave	☐	<None>

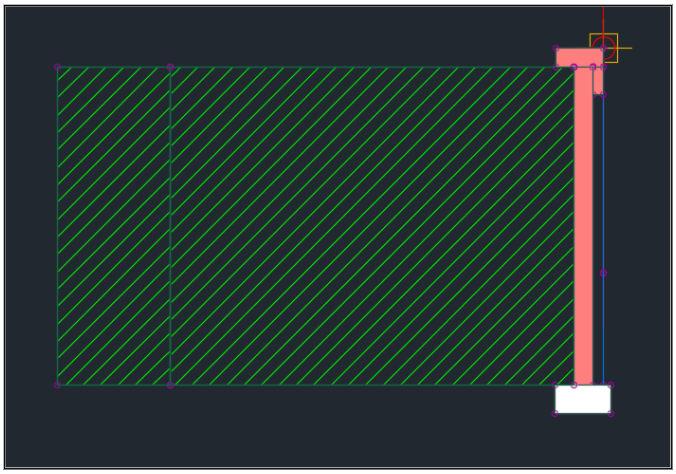
Lip:

Value Name	Default Input Value	Parameter Reference	
		Use	Get Value From
Side	Right	☐	<None>
Top_Slope	0.66:1	☐	<None>
PointsCode	_NoDisplay	☐	<None>
LinksCode	_NoDisplay	☐	<None>
Inside_Slope	0.00:1	☐	<None>
Outside_Slope	0.00:1	☐	<None>
Top_Width	0.33'	☐	<None>
Inside_Height	0.50'	☑	Gutter.Out_OutsideHeight
Bottom_Slope	10.00:1	☑	Gutter.Out_BottomSlope
ShapeCode	Pave	☐	<None>

Top:

Value Name	Default Input Value	Parameter Reference	
		Use	Get Value From
Side	Right	☐	<None>
Top_Slope	10.00:1	☑	Gutter.Out_TopSlope
PointsCode	_NoDisplay	☐	<None>
LinksCode	Pave	☐	<None>
Inside_Slope	0.00:1	☐	<None>
Outside_Slope	0.00:1	☐	<None>
Top_Width	0.50'	☐	<None>
Inside_Height	0.97'	☑	Lip.Out_OutsideHeight
Bottom_Slope	10.00:1	☑	Gutter.Out_BottomSlope
ShapeCode	Pave	☐	<None>

Basic MSE Wall



Ingredients:

- (2) Vertical Deflection
- (6) Quadrilateral
- (3) Math Operators

Place two vertical deflection subassemblies, first attaching to the assembly origin and second to the ending point of the first.

Place quadrilateral for coping cap top on first vertical deflection's point of origin (same location as the assembly origin, but to keep the execution order make sure to click on the vertical deflection point)

Name Coping Cap_Top and enter the parameters below:

ADVANCED	-
Parameters	-
Side	Left
Out_BottomSlope	0.000
Out_BottomWidth	0.850
Out_InsideHeight	0.333
Out_OutWidth	0.000
Out_OutsideHeight	0.333
Out_TopSlope	0.000
Out_TopWidth	0.850
Top_Slope	Horizontal
PointsCode	Basic
LinksCode	Pave
Inside_Slope	0.00:1
Outside_Slope	0.00:1
Top_Width	0.85'
Inside_Height	0.33'
Bottom_Slope	Horizontal
ShapeCode	Pave

Place quadrilateral for coping cap leg on bottom inside point of coping cap top, name it Coping Cap_Leg and enter the parameters below:

ADVANCED	-
Parameters	-
Side	Left
Out_BottomSlope	0.000
Out_BottomWidth	0.200
Out_InsideHeight	0.500
Out_OutWidth	0.000
Out_OutsideHeight	0.500
Out_TopSlope	0.000
Out_TopWidth	0.200
Top_Slope	Horizontal
PointsCode	Basic
LinksCode	Pave
Inside_Slope	0.00:1
Outside_Slope	0.00:1
Top_Width	0.20'
Inside_Height	0.50'
Bottom_Slope	Horizontal
ShapeCode	Pave

Place two math operators on the top outside point of the Coping Cap_Leg

- **Name the first MathOp_FullHeight. In the assembly properties > Construct Tab window set the following:**

Input values:

Value Name	Default Input Value	Parameter Reference	
		Use	Get Value From
Side	Right	☐	<None>
A	8.00'	☑	Vert_WallHeight.DeltaY
B	2.00'	☑	Vert_Embedment.DeltaY
MathOperator	Add	☐	<None>
PointCode	0	☐	<None>

- **Name the second MathOp_Height less CapTop. In the assembly properties > Construct Tab window set the following:**

Input values:

Value Name	Default Input Value	Parameter Reference	
		Use	Get Value From
Side	Right	☐	<None>
A	10.00'	☑	MathOp_TotalHeight.MathResult
B	0.50'	☑	Coping Cap_Top.Out_InsideHeight
MathOperator	Subtract	☐	<None>
PointCode	0	☐	<None>

Place a quadrilateral on the math operator and name it Wall Panels. Enter the parameters below:

Input values:

Value Name	Default Input Value	Parameter Reference	
		Use	Get Value From
Side	Right	☐	<None>
Top_Slope	Horizontal	☐	<None>
PointsCode	Basic	☐	<None>
LinksCode	Pave	☐	<None>
Inside_Slope	0.00:1	☐	<None>
Outside_Slope	0.00:1	☐	<None>
Top_Width	0.33'	☐	<None>
Inside_Height	9.50'	☑	MathOp_TH less CopingCap.MathResult
Bottom_Slope	Horizontal	☐	<None>
ShapeCode	Pave	☐	<None>

Copy coping cap top to bottom inside point of Wall Panels subassembly. Rename to leveling pad, adjust parameter values as below:

Input values:

Value Name	Default Input Value	Parameter Reference	
		Use	Get Value From
Side	Left	☐	<None>
Top_Slope	Horizontal	☐	<None>
PointsCode	Basic	☐	<None>
LinksCode	Pave	☐	<None>
Inside_Slope	0.00:1	☐	<None>
Outside_Slope	0.00:1	☐	<None>
Top_Width	1.00'	☐	<None>
Inside_Height	0.50'	☐	<None>
Bottom_Slope	Horizontal	☐	<None>
ShapeCode	Concrete	☐	<None>

MOVE leveling pad to align the center of the subassembly with the center of the Wall panels subassembly

Place a Math Operator on the top outside point of the “wall panels” subassembly. Rename it to MathOp_StrapLength, set the parameter as below:

Input values:

Value Name	Default Input Value	Parameter Reference	
		Use	Get Value From
Side	Right	☐	<None>
A	10.00'	☑	MathOp_TotalHeight.MathResult
B	1.20'	☐	<None>
MathOperator	Multiply	☐	<None>
PointCode	0	☐	<None>

Place a quadrilateral on the MathOp_StrapLength point, name it EngFill_StrapZone. Set parameters as below:

Input values:

Value Name	Default Input Value	Parameter Reference	
		Use	Get Value From
Side	Right	☐	<None>
Top_Slope	Horizontal	☐	<None>
PointsCode	Basic	☐	<None>
LinksCode	Pave	☐	<None>
Inside_Slope	0.00:1	☐	<None>
Outside_Slope	0.00:1	☐	<None>
Top_Width	12.00'	☑	MathOp_StrapLength.MathResult
Inside_Height	9.50'	☑	MathOp_TH less CopingCap.MathResult
Bottom_Slope	Horizontal	☐	<None>
ShapeCode	EngFill	☐	<None>

Place a quadrilateral on the EngFill_StrapZone's top outside point. Call this EngFill_ExtraZone. Set the top width to two, pass the height as below:

Input values:

Value Name	Default Input Value	Parameter Reference	
		Use	Get Value From
Side	Right	☐	<None>
Top_Slope	Horizontal	☐	<None>
PointsCode	Basic	☐	<None>
LinksCode	Pave	☐	<None>
Inside_Slope	0.00:1	☐	<None>
Outside_Slope	0.00:1	☐	<None>
Top_Width	2.00'	☐	<None>
Inside_Height	9.50'	☑	EngFill_StrapZone.Out_OutsideHeight
Bottom_Slope	Horizontal	☐	<None>
ShapeCode	EngFill	☐	<None>

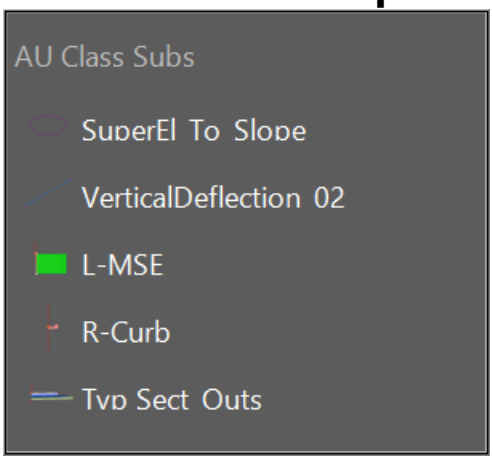
Adding assemblies to the tool palette

Select assembly marker

Click and drag on the red assembly marker, mouse over the tool palette, release the mouse button. The assembly will add to the tool palette with a reference image.

Open provided Civil 3D DWG, "2016-AU-Corridor.dwg"

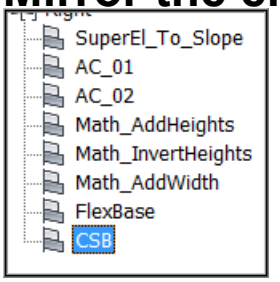
Import the curb and MSE assemblies created earlier in this class. Import the subassembly called "AU_VerticalDeflection_02.pkt" from the class files. Then import the roadway assembly from last year's class. Your tool palette should look like this:



Insert the assembly from last year's class, Typ Sect Outs.

Using the Insert BEFORE command, insert a SE to Slope subassembly before the first layer of the Typ Sect Outs assembly. Pass the SE to Slope subassembly's output to the appropriate slopes on your roadway assembly.

Mirror the entire roadway (SE all the way to the CSB):



Adjust names of the subassemblies by adding a "L-" prefix to each name.

Insert the curb assembly.

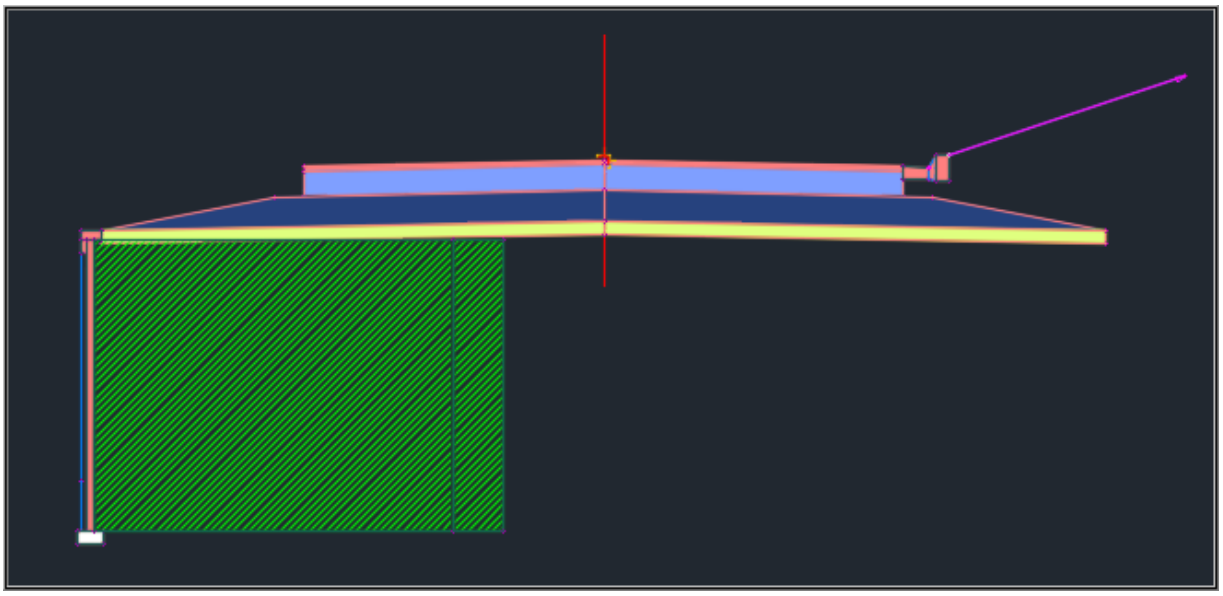
Add curbs by copying to the right side of the roadway, and then mirroring to the left side. Pass the slope parameters from the roadway.

Rename the curbs appropriately.

Use the link slope to surface subassembly to daylight on the right side of the roadway, set the slope to 3:1

Insert the MSE Wall assembly.

Copy the entire MSE wall subassembly to the bottom outside point on the left side of the roadway assembly. It should look like below:



Create a corridor, auto map the surface target to the only surface in the file. Sections of the corridor should look like below: