AUTODESK UNIVERSITY 2014

Tales from the Road - Corridor Modeling Best Practices

David Zavislan, P.E. – Wood Rodgers, Inc.

CV5646-P In this class David shares corridor modeling lessons he has learned over the past seven years from his experiences training and supporting the use of AutoCAD® Civil 3D® in design processes at several engineering firms. Discussion will cover a wide range of projects, including using corridors for site grading, levees, basic roadway models and highway-widening projects. The class will provide tips and guidelines for determining how to create assemblies, using conditional subassemblies, when to use offset assemblies, and the competitive advantage of developing custom subassemblies.

Learning Objectives

At the end of this class, you will be able to:

- Identify a wide range of applications for corridor models
- File management tips for optimal model performance
- Develop assemblies to meet specific design objectives, such as 3D models, sections, or material quantities
- Build smarter corridor models with conditional subassemblies
- Describe the need for and benefits of custom subassemblies

About the Speaker

David is an Integrated Technology Engineer for Wood Rodgers in Sacramento California. Wood Rodgers is a full-service civil engineering firm with offices in Northern California and Nevada. David supports the firm's engineering and document delivery processes - developing CAD tools, training, and standards. David is a licensed civil engineer in California and Colorado. He has spent many years working as a CAD consultant, specializing in civil engineering, surveying, and mapping. Over the span of his career he has held many positions in the engineering industry: CAD manager, structural engineer, staff engineer for a municipal public works department, project engineer and project manager. He also worked several years as a software trainer and consultant for a software reseller.

dzavislan@woodrodgers.com

WOOD RODGERS
DEVELOPING INNOVATIVE DESIGN SOLUTIONS

Introduction

This document contains a summary of best practices for working with corridor models. The information presented is based on over seven years of lessons learned while working on and supporting a wide variety of projects at several engineering firms.

The Projects - Corridor Modeling Applications

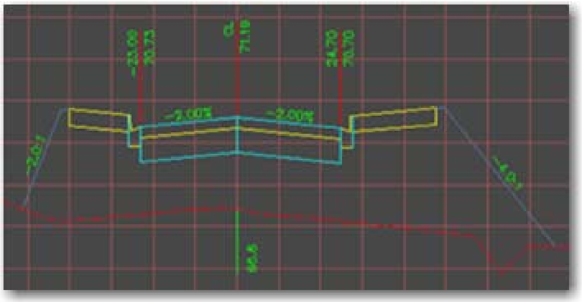
This section contains a sampling of some of the corridor modeling projects that I have been involved with during the past seven years. They are presented as examples of the types of models that can be created with corridors and as introductions to some of the lessons contained in this document.

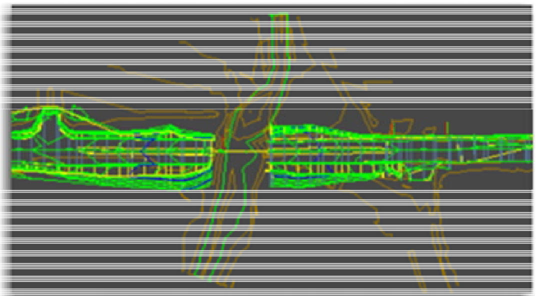
Basic Roadways

The first examples are of some basic roadway projects. These represent the majority of the models that I have worked on in Civil 3D. Some of these were fairly simple models with typical road sections applied to the entire length of the road. However, they quickly become complex as design conditions changed. The addition of cross streets, driveways and grading constraints turned simple corridor projects into very complex ones.

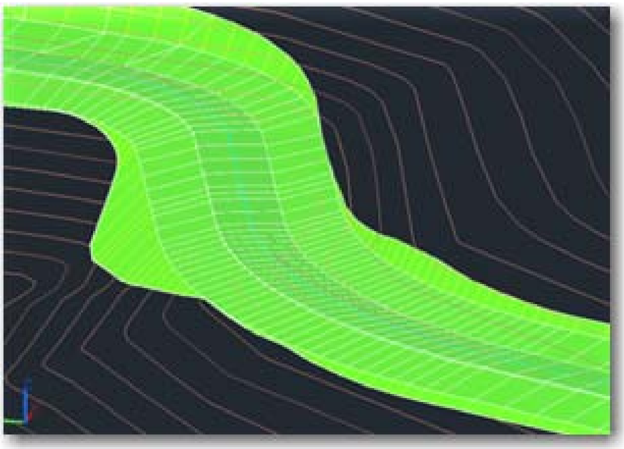
The following images are from a bridge replacement project located in Northern California. The model was comprised of a single alignment and profile with separate corridors for the two bridge approaches. A third corridor model was also created to generate bridge deck contours for the structural engineer.

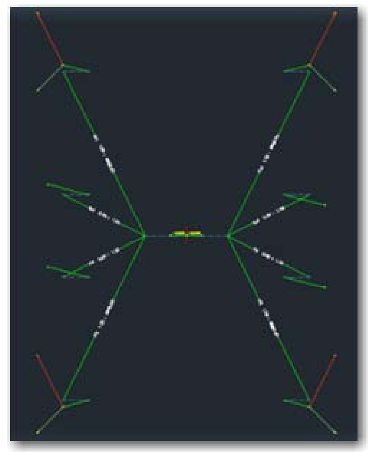
Modeled in the 2007 version of Civil 3D, the corridors for this project required multiple regions to accommodate the varying curb and sidewalk conditions encountered as it crossed side streets and driveways. In order to keep track of the numerous assemblies and target conditions on this model, a logical naming standard for the subassemblies and assemblies was developed and implemented.





Another basic road example is from a project that was modeled in Civil 3D 2010. It was a private access road that was designed for a coastal resort in Mexico. This unique, tropical location presented some design challenges. First, for safety concerns, the shoulder area needed to maintain a maximum cross slope of 20:1 within 5 meters of the roadway. If the model daylighted beyond that distance, it needed to do so as closely and steepley as possible to limit the impact on native vegetation. I learned that the process could be siimplified by incorporating conditional subassemblies, instead of breaking the corridor into multiple regions and assemblies. Conditional cut-fill subassemblies allowed for the creation of a single assembly that met all design objectives, and automatically determined where the different slope conditions needed to be applied.

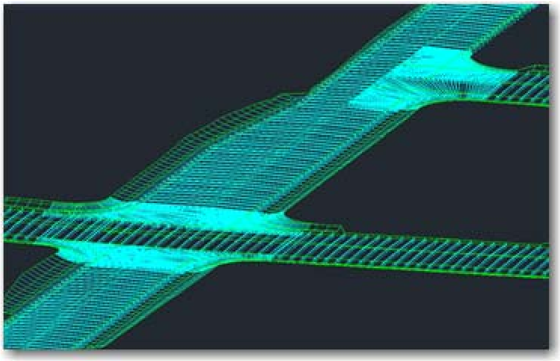


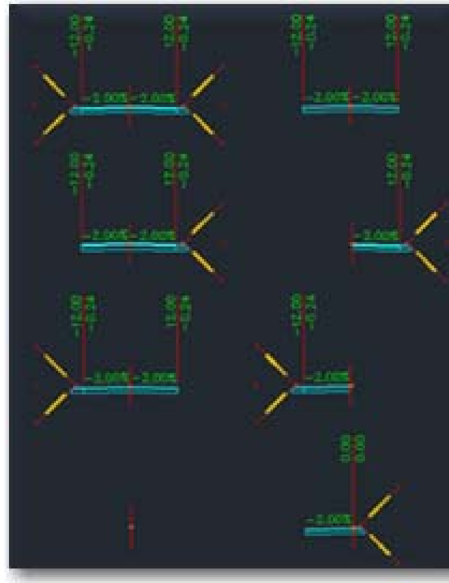


Intersections

The following images show a single corridor that was used to model all of the roads in a small residential subdivision. In the foreground is a four-way intersection where the main road's crown has been extend through the intersection. In the background is a T intersection where the crowns of both roads have been extended through the intersection.

Civil 3D has allowed the creation of models like this since the 2007 release. Because of the amount of work required to build a detailed intersection model in these earlier relesases, it was greatley appreciated when Autodesk incorporated the intersection modeling tools in Civil 3D 2010. Prior to 2010, a designer would have to individually create the alignments and profiles for all of the offsets and curb returns. The elevations and grades of the offsets and curb returns would have to be calculated and manually entered. The Civil 3D program's intersection modeling tools automate this process, reducing the time it takes to model a single intersection from around 45 minutes to less than 15 minutes.





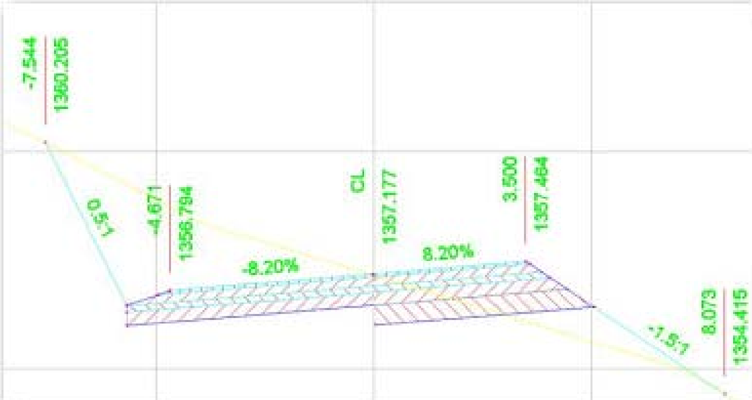
Highways

Highway design modeling is where Civil 3D corridors display their power. The following images show a sample of the corridor and assembly that was used to model 90 kilometers of rural, two-lane highway in Oaxaca Mexico.

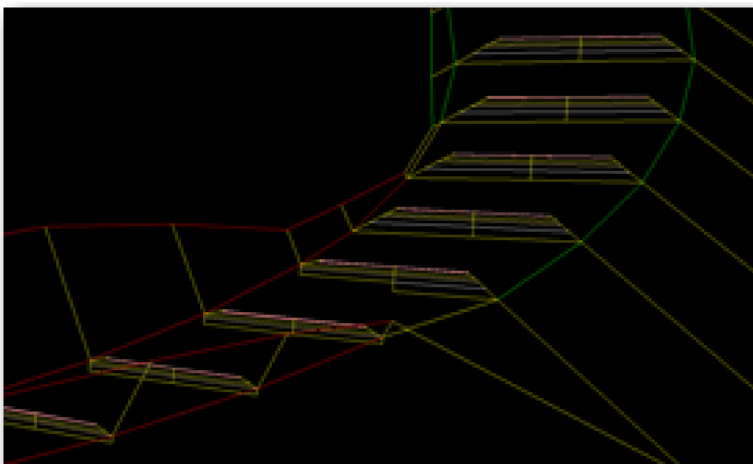
This project was modeled in Civil 3D 2010 and was one of the first that I was able to exploit the powers of custom subassemblies. A custom subassembly was created with the ability to determine if it was in cut or fill. If it was in cut it would apply pavement, subgrade and a ditch hinge point as shown on the left side of the typical section. In fill conditions it would change to a steeper shoulder and add a third subgrade material layer.

In addition to the different sections in cut or fill, this assembly also automatically widened the travel lane on the inside of curves based on the critical superelevation points of the alignment.

This subassembly was created for Civil 3D 2010 usind the VB .NET programming language. One of the biggest drawbacks of custom .NET assemblies is that they require updating for each new release of Civil 3D. Autodesk changed the superelevation programming interface with Civil 3D 2011, forcing the need to recreate this subassembly so that it could be used in the newer version. For 2011, the inside curve widening functionality of the subassembly was removed, since Civil 3D now had that capability with offset alignments and widening criteria.



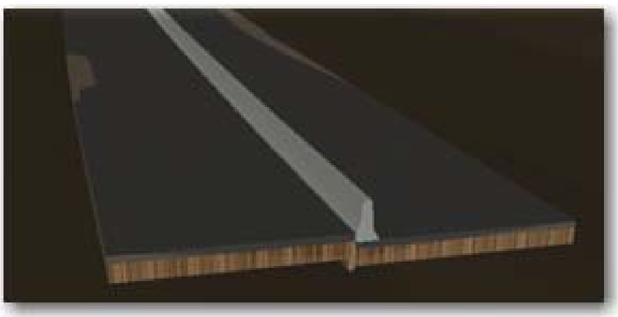
Typical Section showin cut condion on left and fill condition on right.

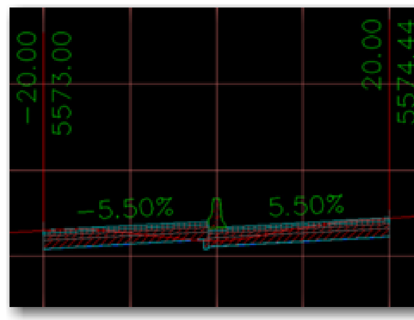


3D view of coridor in transition from cut to fill section.

This next highway project example is from a Northern California highway widening project where the highway median was replaced with full-depth pavement HOV lanes. This project was done in Civil 3D 2007 and the custom subassembly was created with VBA. It was created by adding additional pavement, subgrade, and the sub-drain trench shapes to a copy of the stock OverlayMedianAsymetrical subassembly provided with Civil 3D.

The subassembly extracted the surface elevations at the edge of pavement (sawcut) points on each side and extrapolated them towards the centerline. The lowest pavement section at centerline would get extended to the opposite edge of the median barrier and was used to define the bottom of the barrier. The upper portions of the barrier was then located relative to the pavement section on the high side. Unlike the OverlayMedianAsymetrical subassembly , this subassembly was furthar modified to read cross slope values from the alignment's superelevation data.

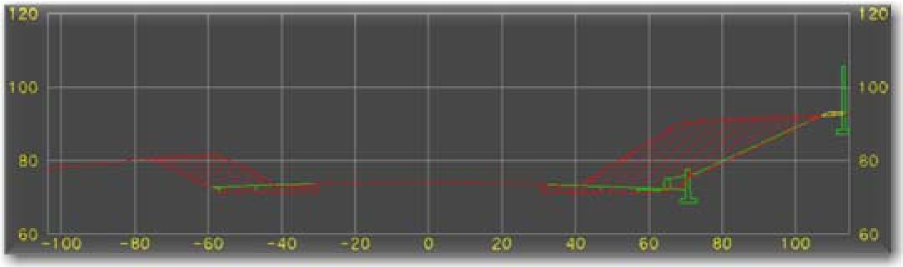


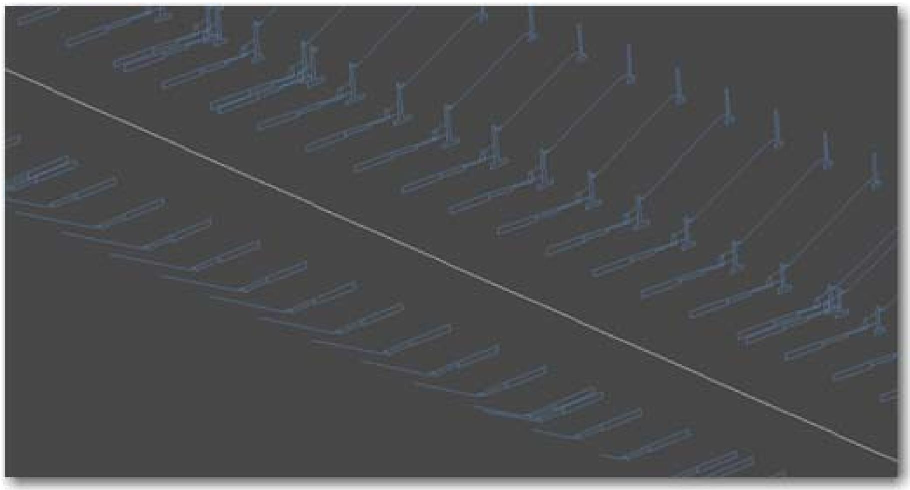


Another highway widening project that provided lots of 'lessons learned' was an auxiliary lane project on California Highway 1. An auxiliary lane is a continuous lane added to the outside edge of the existing highway that continues from an on-ramp to the next off-ramp.

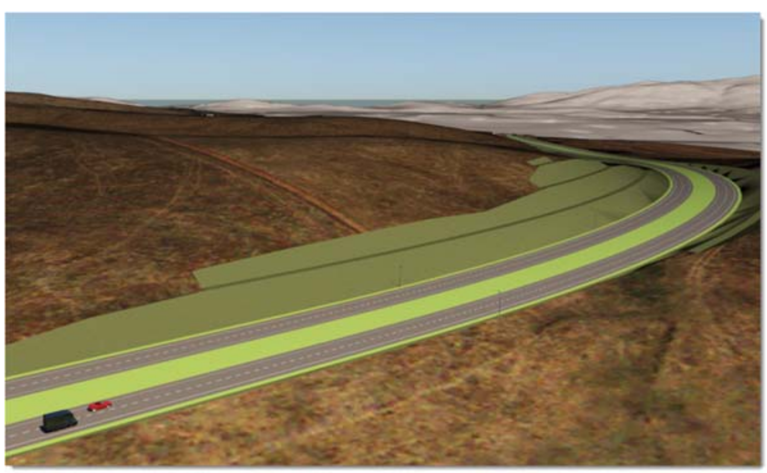
This project was modeled in Civil 3D 2011. The corridor model was created with the highway centerline as the main baseline alignment, using it for all stationing control and cross sections. Separate alignments and offset assemblies were used for the 'saw-cut' lines on each side of the highway. Additional alignments were used to locate retaining walls and sound walls, as they were not always parallel to centerline.

This project exposed some limitations that Civil 3D 2011 had with Assembly Offsets. The model did not calculate materials correctly if the alignment used for the offset did not have station values within the same range as the baseline alginment. Because of these limitations, the offset components were located with generic links. This resulted in a model where all assembly components were created radial to the main alignment, and did not accurately represent the portions of the molel where the target alignments were not parallel.

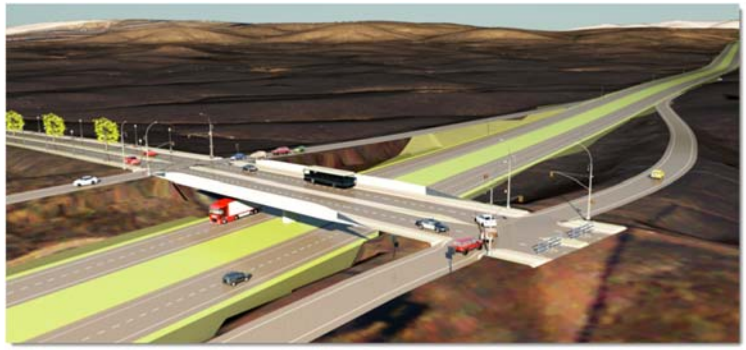




A recent project involved a corridor that was built to analyze several design alternatives for six miles of expressway in the Sacramento region. The model was completed in Civil 3D 2013 and imported as imx data into InfraWorks for renderings and animations. It took less than three days to create the alignment, profiles, assemblies and corridor models for two design alternatives and render them in InfraWorks.



Renedered image of expressway from InfraWorks.



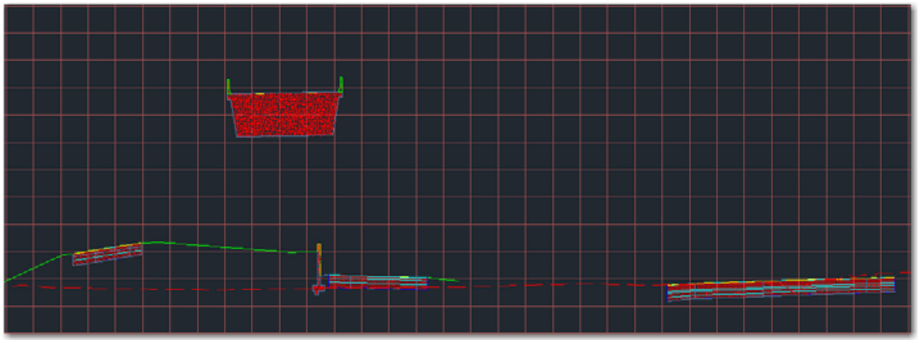
Interchange model rendered in InfraWorks

Interchanges

I have also experienced modeling several highway interchanges with Civil 3D corridors. These have been some of the most challenging projects from a data management standpoint. The main lesson learned was that maintaining strict naming standards for subassemblies, assemblies, alignments, profiles and corridors was extremely valuable in keeping track of multiple corridors and all of the target elements required to control the lane transitions and matching grades.

Interchange corridors and surfaces rendered in Navisworks.

These multi-corridor models also exposed limitations of Civil 3D that existed in earlier versions. Prior to the 2012 version, the software could only sample corridor sections that had the same baseline alignment as the sample line group, and where the corridor frequency coincided with the sample lines. To overcome this limitation, surface models had to be created of all the top, pavement, subgrade and datum layers of each corridor. All of these surfaces could then be sampled and displayed in sectoin views. This was resolved with the 2012 version of Civil 3D and its ability to sample sections anywhere on a corridor, and its ability to sample corridors that have different base lines than the sample line alignment. The following image shows a cross section that was generated by sampling four separate corridors.

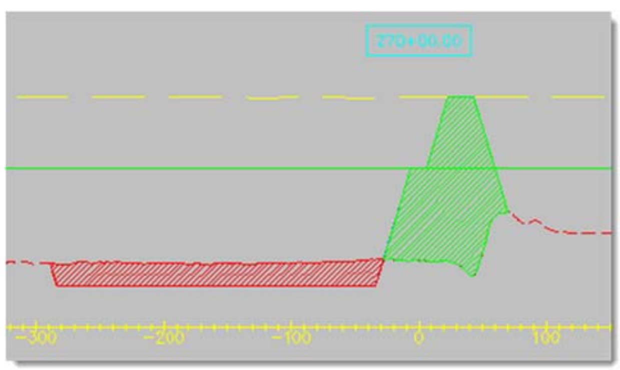


Four separate corridors displayed in Civil 3D 2012 section view.

Levees

Living in the Sacramento River and Delta region, I have also had the pleasure of working on several levee projects. Levee corridors have typcally been used to generate surface models for earthworks calculations. Because of this, assemblies for these models tend to be built mostley with the generic link subassemblies.

Another issue presented by levee projects was that most of the existing terrain data was collected using LiDAR. Because of the large volume of data LiDRA generates, It was not uncomman to have DEM databases that were excess of 1GB in size. In order to work effectively with this data in Civil 3D, workflows were developed to reduce the size of surface models and improve the program's performance.



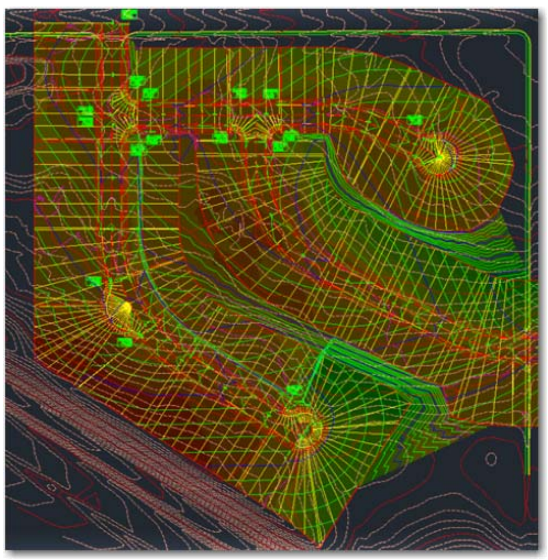
Levee cross section

Site Grading

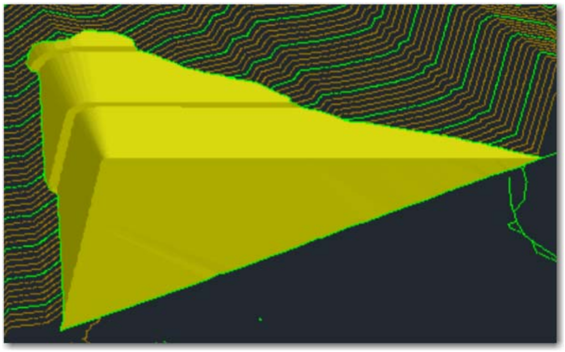
In addition to roadway modeling and levees, I have also incorporated corridors into the development of site grading models.

The following image shows a small subdivision project where a single corridor was used to model a preliminary grading surface. (This model is explored in greater detail in my other Autodesk University 2014 clasee: CV5644 – Advanced Grading Solutions Using AutoCAD Civil 3D Corridors.)

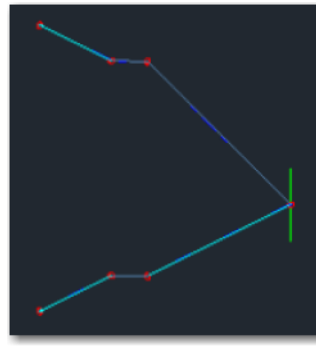
A series of assemblies were created for the various lot grading configurations. These were assigned to regions that could be easily grip edited to locate them at the appropriate lot line locations. Lot depths were controlled with target alignments and polylines.This approach allowed multiple grading scenarios to be analyzed simply by swapping assemblies or by editing profile grades.



Corridors also lend themselves to modeling any linear grading element. The follwoing images show an example of using a one-sided assembly to apply daylight benching around the perimeter of a site. A surface model can be created directly from the corridor, or feature lines can be extracted and used as breaklines in other surfaces.



Benched slope around perimeter of site.



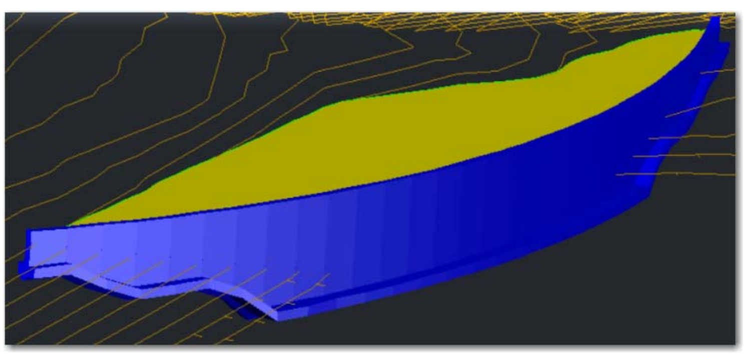
One-sided bench slope assembly.

The final grading example shows a retaining wall model that was created using a couple of custom subassemblies.

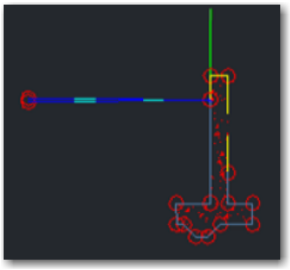
The daylight element on the left side of the assembly allows for negative slope values, which essentially lets the assembly daylight up in fill situations, and daylight in a downwards direction while in cut.

The other custom subassembly creates a wall shape that uses profiles to control the wall height, top of wall and footing elevations. This capability simplified the wall modeling process because the stock wall subassemblies, that came with Civil 3D, could only target surfaces for the location of the top and bottom of the wall.

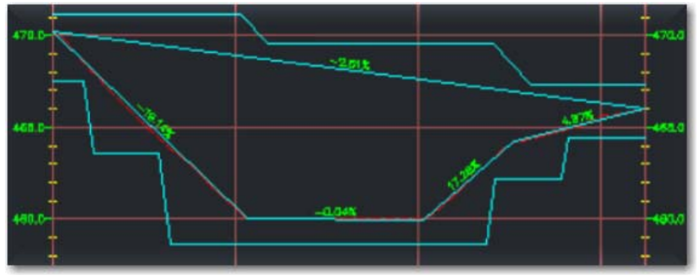
Both of these subassemblies were created with the Subassmbly Composer program that is installed with the latest releases of Civil 3D. With a fairly quick learning curve, the Subassembly Composer allows non-programmers to create custom susbassemblies which can be used to solve unique design conditions.



3D view of wall with backfill.



Retaining wall assembly

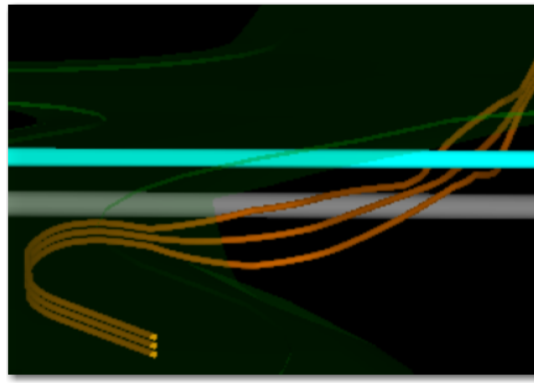


Retaining wall profiles.

More…

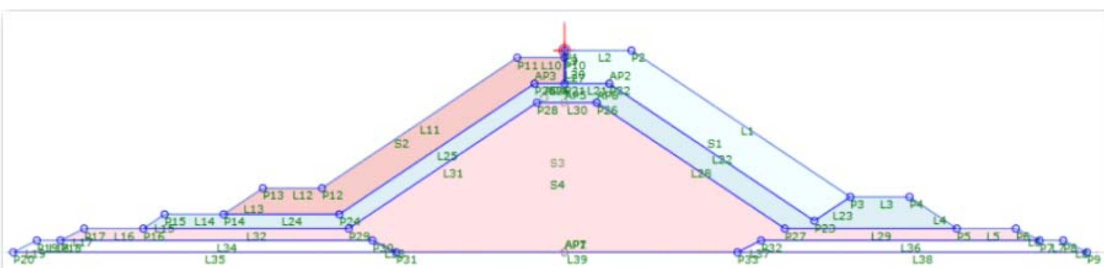
In addition to the preceeding examples, tools like the Subassembly Composer open up many more uses for corridor modeling. Corridors can be used to model any lineal element, such as the grading slopes, and retaining walls examples presented previously.

The image to the right shows a corridor that uses a custom subassembly to model a triple condiut package for undergound electrical transmission lines. The subassemlby uses superelevation parameters to control the distance between the conduits and the rotation angle of the package, allowing them to be easily modified along the length of the alignment.



3 Phase, high-voltage conduit subassembly displayed in Navisworks.

Combining all of the corridor modeling capabilities with tools like the Corridor Solids Extractor and Navisworks will allow for the creation of quick and accurate BIM models of civil infrastructure.



Breakwater subassembly created with Suassembly Composer. (Image courtesy of Dave Simeone, Autodesk.)

The Lessons:

File Management for Corridor Models

Managing Civil 3D data is important because not everyone has access to the latest and greatest computer technology. Most engineering firms are in a never-ending process of upgrading all of their computers to the latest version of Windows, with ever increasing amounts of RAM. Because of this, design models need to be created so that they can be effectively worked with on all workstations. This can be accomplished by applying some 'best practices' to the Civil 3D corridor modeling workflow.

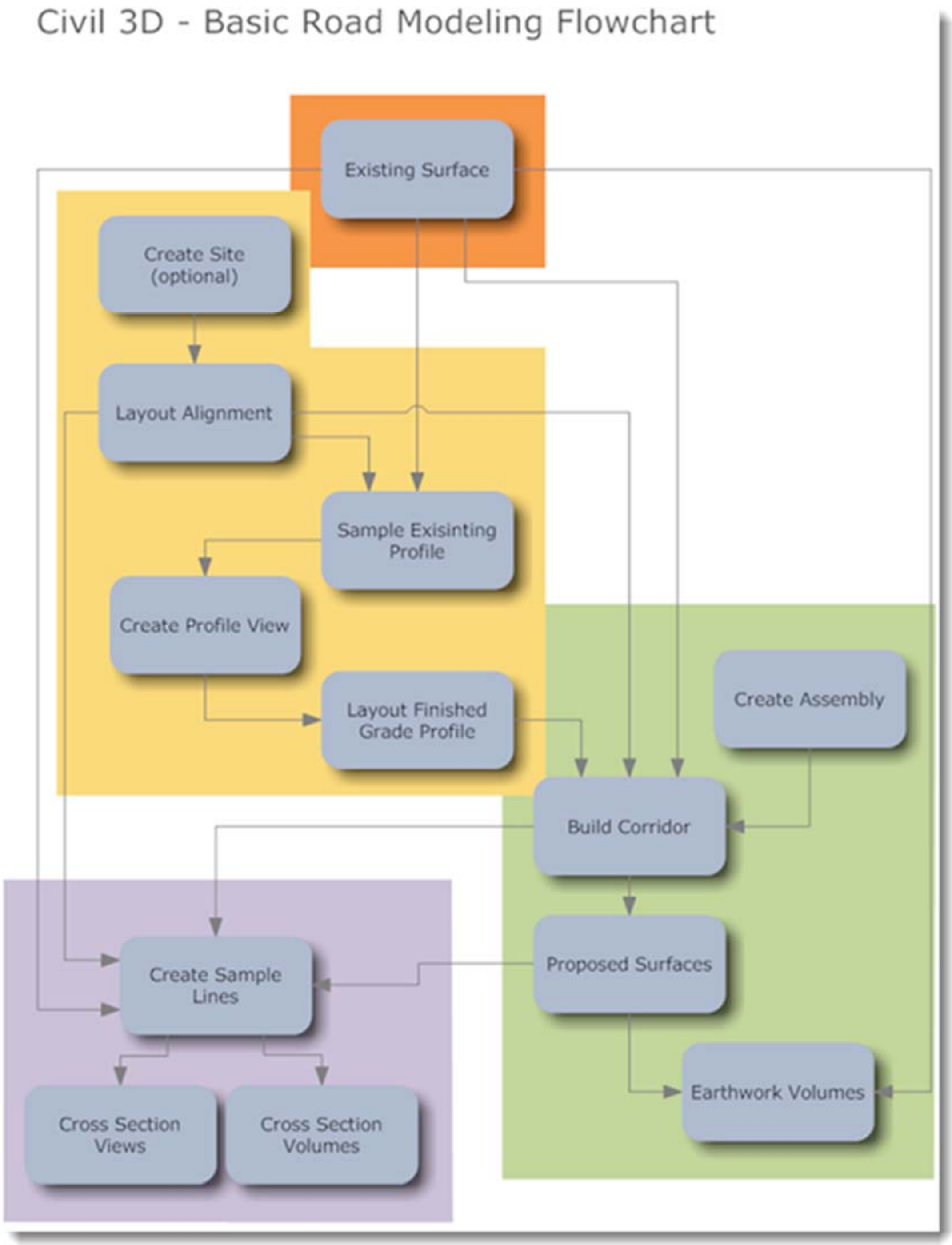
How does your workstation compare? Even if you have 64-bit workstations with more than 16 GB of RAM, your modeling workflows and efficiency will improve by utilizing these file management suggestions.

Some of the tips that are presented in this class were taken from the Civil 3D Best Practices Guide. This document is available from the on-line help in Civil 3D, or you can search for it on the Internet.

Corridor Modeling Workflow

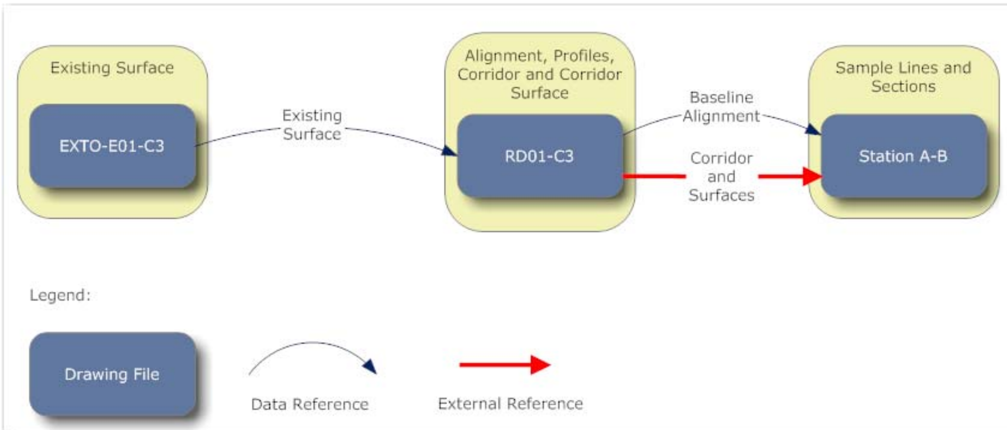
The following flowchart identifies the general steps for modeling corridors and generating cross sections in Civil 3D. The steps are organized into groups based on the drawing files that they should be performed in. The drawings are represented by the four colored regions in the background of the flowchart:

- Orange = Surface source drawing.
- Yellow = Alignment and profile source drawing.
- Green = Corridor and corridor surface drawing.
- Purple = Cross section drawing.

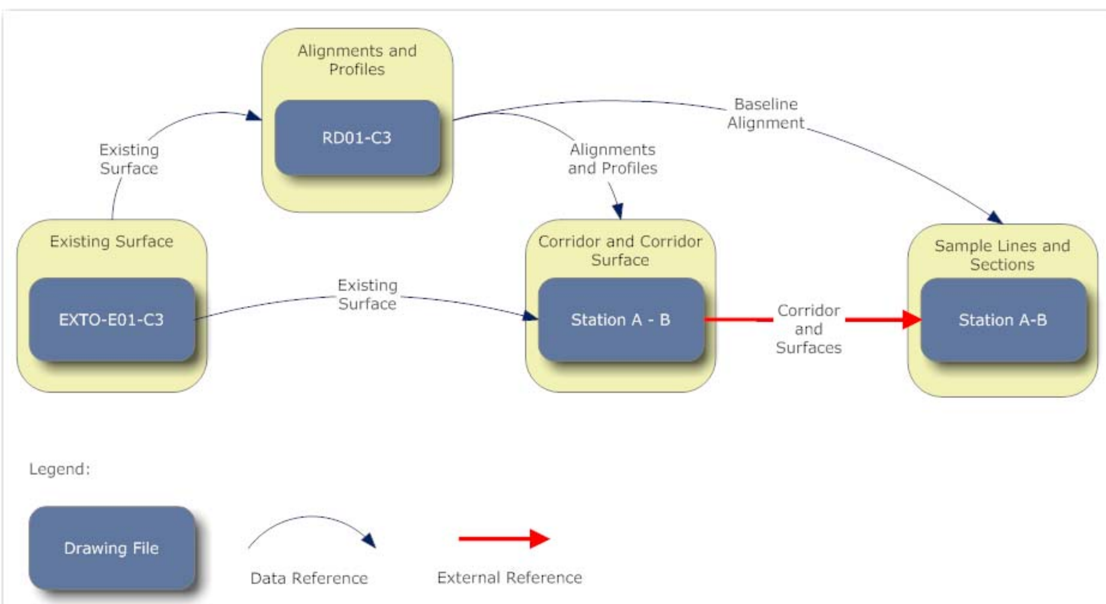

Civil 3D - Basic Road Modeling Flowchart
Existing Surface
Create Site (optional)
Layout Alignment
Sample Exisinting Profile
Create Profile View
Layout Finished Grade Profile
Create Assembly
Build Corridor
Proposed Surfaces
Earthwork Volumes
Create Sample Lines
Cross Section Views
Cross Section Volumes

Drawing Organization

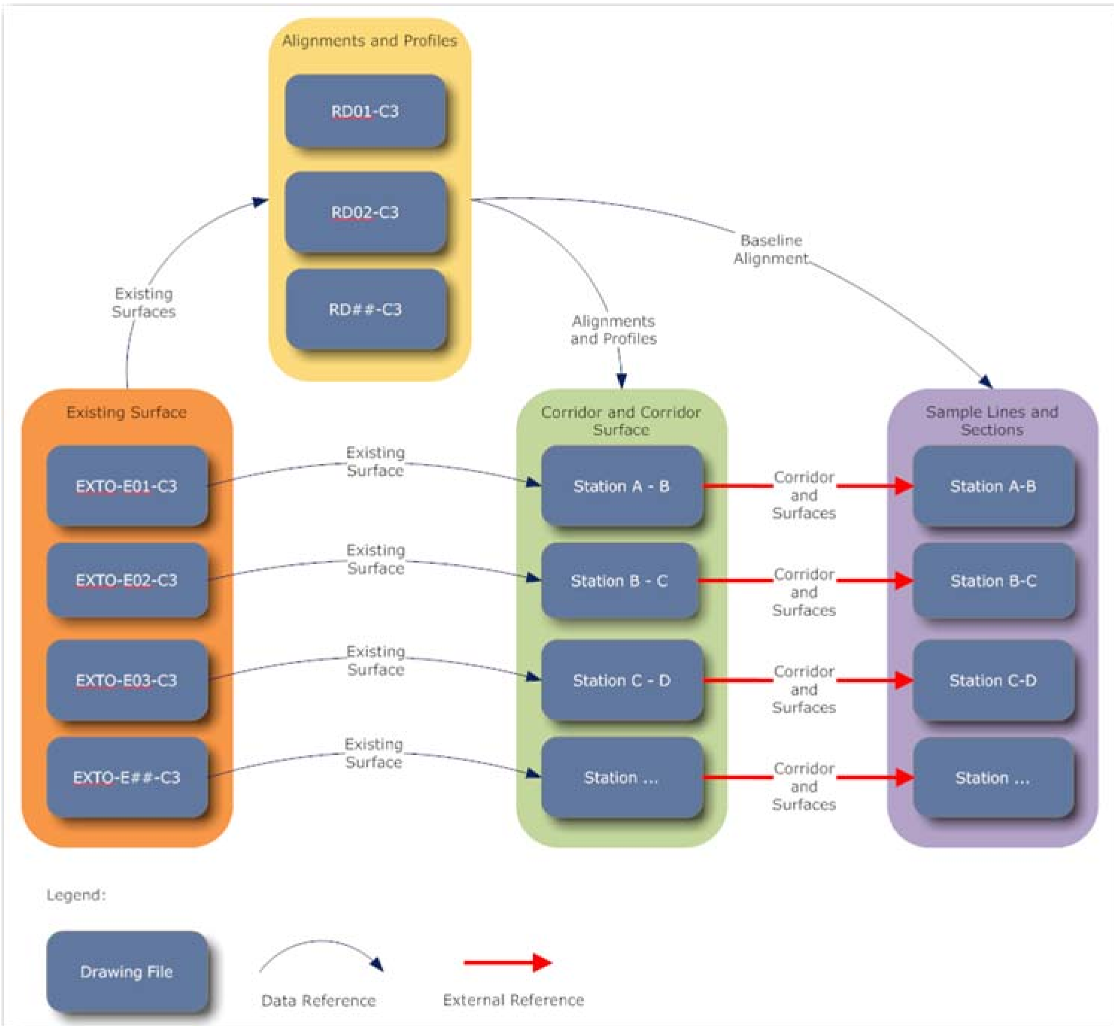
The following image shows the most basic drawing setup for corridor modeling. This works well for short alignments and projects with a single designer. In this scenario, all of the alignment, profile and corridor data is stored in a single drawing. While you could also create the sample lines and cross sections in this file, it is still better for system performance to create them in a separate file.



As the data files become larger, you may want to consider splitting the corridor model and corridor surfaces into a separate file from the alignment and profiles. This file organization is shown in the following diagram. The main benefit of organizing the drawing files this way is that multiple designers can access the alignment data simultaneously. Or, you can have several corridor models, representing different scenarios, all referencing the same alignment and profile.



The next chart shows how this drawing organization can be expanded by dividing surfaces and corridors into multiple tiles or regions. Sorting Civil 3D model objects into multiple files helps to manage the amount of data that is being accessed, thereby improving the performance of Civil 3D. An additional benefit of this organizational structure is that multiple designers can simultaneously work on different portions of the design.

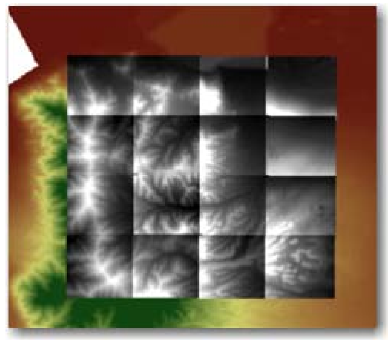


Since the data is contained in multiple files, you need to make sure that all model elements are properly built and saved in the source drawings, and that the data shortcuts are properly synchronized in subsequent model files.

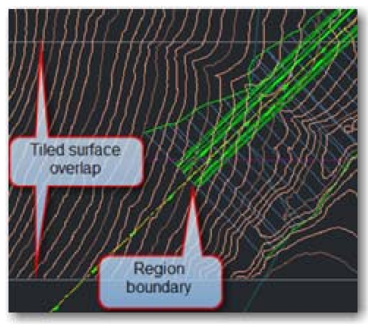
Managing Surface Data

Excessively large surface models are usually the cause of most performance issues when working with Civil 3D corridors. This section highlights the commands and techniques that can be used to manage surface data to make it more efficient to work with. These processes will require you to spend a little upfront time before you create your models, but the resulting time savings will be worth the effort.

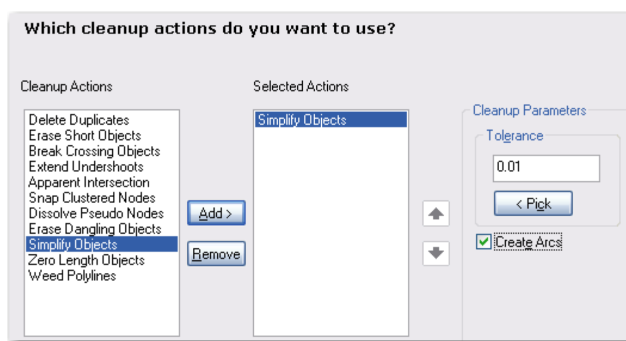
- Software performance can be improved for projects covering large areas by cutting the surface into tiles. Then reference only the tiles that are needed for the portion of the model that you are working on.



- When creating tiled surfaces, you want to make sure that adjacent surface tiles overlap enough so that corridor regions can daylight to both sides. It becomes difficult trying to map different target surfaces to opposite sides of an assembly, or to control where the region boundaries need to occur.

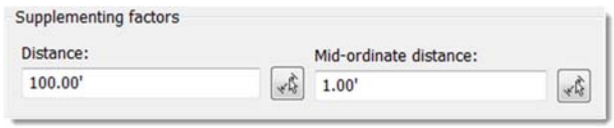


- When creating existing ground surface models from contour data, use the Map Drawing Cleanup Tools, Simplify Objects action to remove excess vertices in contour polylines. Set a tolerance distance to define the maximum displacement of the simplified polyline relative to the original. Toggling on the Create Arcs options will remove even more vertices from the drawing, further reducing the file size.

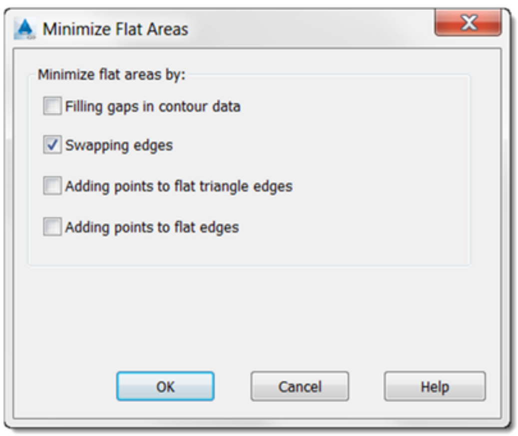


- When adding the contour data to the surface use the weeding factors to reduce the number of points in the surface. Increase the Distance value for greater reductions in model size. Don't use a weeding angle much greater than 4 degrees. Lager angle values will start to degrade the accuracy of the model.

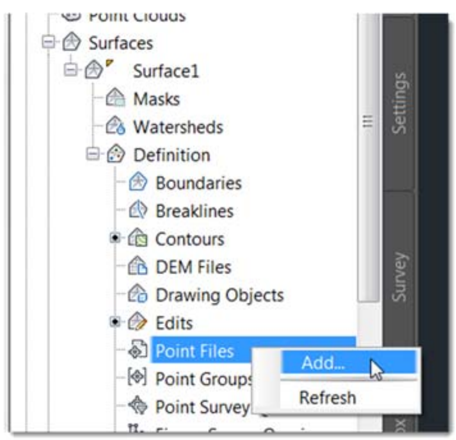
- Supplementing factors will add points to the surface on long segment and on curves. This can add back some of the vertex information that was removed with the Map Drawing Cleanup Tools.



- Use the Swapping edges option when minimizing flat areas in the surface caused by the triangulation of contour data. This method will create a surface that closely matches the original contour data without adding points to the surface.



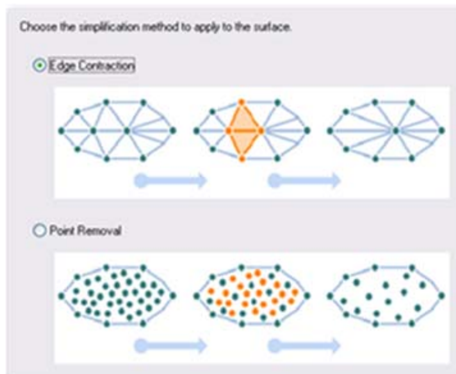
- Add large point data files directly to the surface model. Do not import the point file as COGO points.



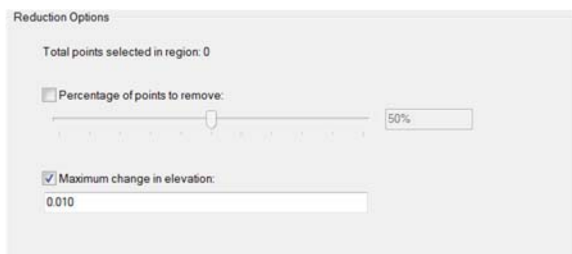
- For surfaces that are created form point data, you can use the Simplify Surface command to reduce the number of points and triangles in the surface. This command does not work on surface points created from breakline or contour data.

  Run the Simplify Surface command only once. Use the Finish button instead of the Apply button. Each time either button is clicked the function executes, comparing the elevation change to the current state of the surface, as opposed to the original. This compounds the changes in elevation.

  It appears that the Edge Contraction method does a better job of removing point data without adversely affecting the accuracy of the model.



  If you use the Point Removal method, the best results seem to occur when specifying only a maximum change in elevation.



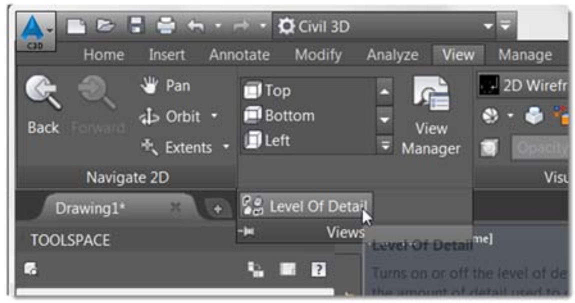
- Define data clip boundaries to restrict the point data that is added to the surface definition. The data clip boundary must be the first operation in the surface definition.

- Define an outer surface boundary to limit the surface triangulation to the extents of your design area. The target surface models along a corridor only need to be wide enough to accommodate daylighting and cross section sampling. Since boundaries are dynamic, you can easily adjust boundary vertices and extend the surface area to accommodate design changes.

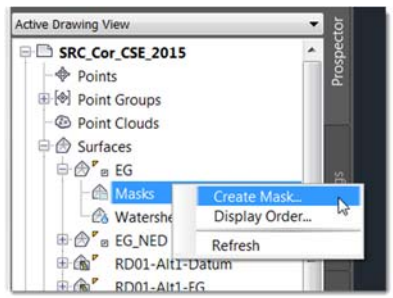
- Surface cropping became available in Civil 3D 2010. Surface cropping allows you to extract a smaller area from a surface that remains dynamically linked to the original.

- When data referencing surfaces, assign styles that display only the border. This will keep your corridor drawing file size smaller and improve regeneration times. By displaying the border, you will still be able to verify that your surface area is great enough to accomdate your corridor model.

- Turn on Level of Detail when you need to display the contours for large surface models.

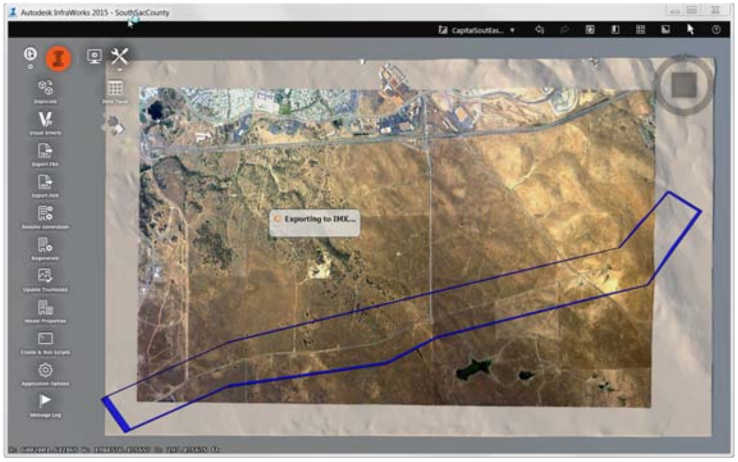


- If you must have contours displayed, apply a surface mask to limit the extents of the contours. This will improve regeneration times, and minimize the file size.



- Set the PROXYGRAPHIS variable to 0. This will prevent Civil 3D/AutoCAD from creating a bitmap of the surface, or other Civil 3D objects, in the drawing file. This helps reduce the file size. Remember that without the proxy image in the drawing, Civil 3D objects won't display in plain versions of AutoCAD, unless the Civil 3D object enabler is installed.

- The InfraWorks program can be used to aggregate, and convert terrain data to project coordinates. It has an IMX export command that allows you to crop and export the surface data to a file that can be import into Civil 3D. The IMX file format is used by Civil 3D and InfraWorks to share road and surface data. The following image is a screen capture of InfraWorks during the IMX file export process. The blue fence indicates the area of the model that is being exported. In this example, the resulting surface drawing was reduced from 100 MB to 10 MB in size.



Managing Alignment Drawings

Since alignments and profiles are fairly basic objects, there is not much you can do, or need to do to improve performance when working with them.

The main benefit of a separate drawing file for the alignments and profiles is that the same alignments can be data referenced to multiple corridor drawings.

In addition to the alignment and profile data, this drawing will also be where you apply design criteria to generate offset alignments and superelevation data. Widening parameters can also be applied with corresponding superelevation attainment methods to adjust offset alignment widths.

Managing Corridors

The following suggestions are applied in the corridor model drawing file.

- Keep the rebuild automatic option turned off. This will prevent the corridor from rebuilding whenever you make changes to alignments, profiles or assemblies. You can manually run the rebuild command after a series of edits to apply any changes.

- Create separate code sets for displaying assemblies, corridors and cross sections. The code set for assemblies should display all links and nodes to make picking subassembly attachment points easier. The code set for corridors should only display the links, without any points, shapes or shape hatches. This allows quicker processing when updating the corridor and viewing it in 3D. The code set for sections should include all the lines, hatches and labels that are typical in all the cross section views.

- You probably do not need corridor sections every foot. Start with a large frequency value to work out the layout of the corridor model. Then, as you get closer to final design, reduce the frequency distances to values that you will need for sections or display purposes.

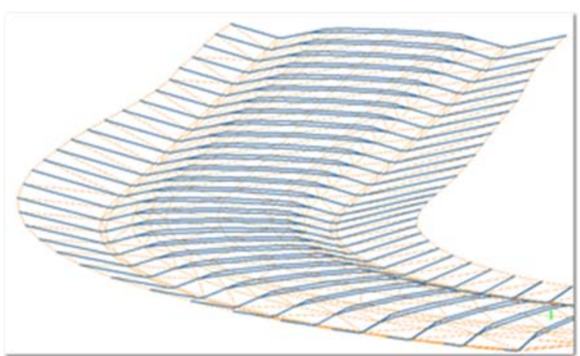
- For most corridors use only a single baseline. Only apply multiple baselines when modeling elements that are not parallel, or if you need to use different profiles. If you do create multiple baselines from the same alignment, be careful not to overlap regions because assemblies from both baselines may be applied at the same section.

- Prior to the 2012 version of Civil 3D, multiple baselines would not show up in section views or stake out reports. If this was required, a single baseline with complex assemblies had to be used. A complex assembly can be created with one or more assembly offsets or by using the generic links to define the offset locations.

- A region will be required for each portion of the corridor model that uses a different assembly, requires a different frequency, or has different targets.
  Regions can also be used to manage large corridors by allowing you to turn off regions that are not being worked on. This improves Civil 3D's performance while you are making edits to the model. Once you have the design refined, toggle on all regions and rebuild the corridor.

- The corridor drawing file will also contain the surfaces that are generated from the corridor. Like the corridor, it is best to keep their rebuild automatic option toggled off. You can also reduce file size and improve regeneration times by displaying corridor surfaces with the simplest style possible.

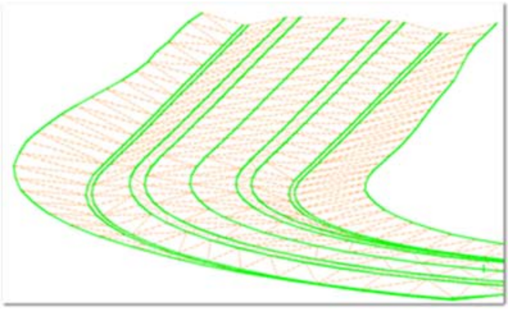
- Links are the easiest method to create a surface from a corridor. Most stock subassemblies will have 'Top' and 'Datum' links.



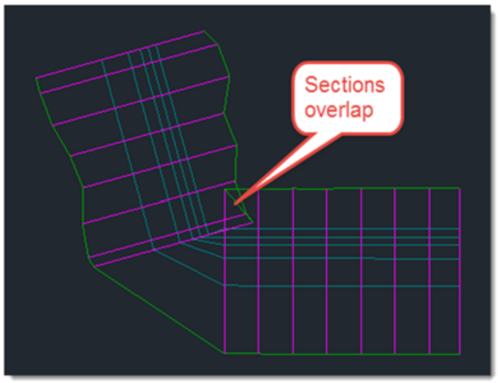
Depending on the subassemblies used to define an assembly, there may be instances where links, with the same name, may overlap each other. The Edge Overhang Correction option can be used to control how Civil 3D interprets these overhangs when outputting data to a surface model. Typically, you will want to assign the 'corrected to top links' option for the Top surface and the 'corrected to bottom links' option for the datum surface.

You will need to output the link data as breaklines if there are areas of the model where the link lengths exceed the frequency distance.   This usually occurs on long daylight slopes.

- Creating the corridor surface by using feature lines can improve triangulation along walls and curbs where the geometry is not parallel to the corridor baseline.

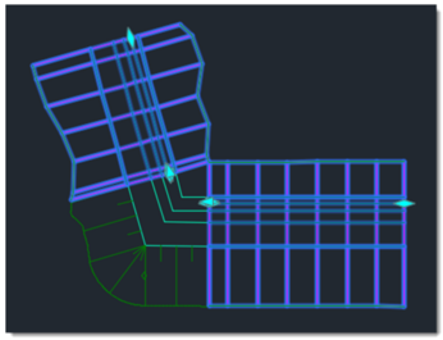


- Avoid short radiuses and tight angle points on alignments. Since each corridor section is calculated independently from all other sections, they will overlap adjacent sections where tight bends occur in the alignment, or if the curve radius is less than the assembly width.



Corridor with overlapping sections

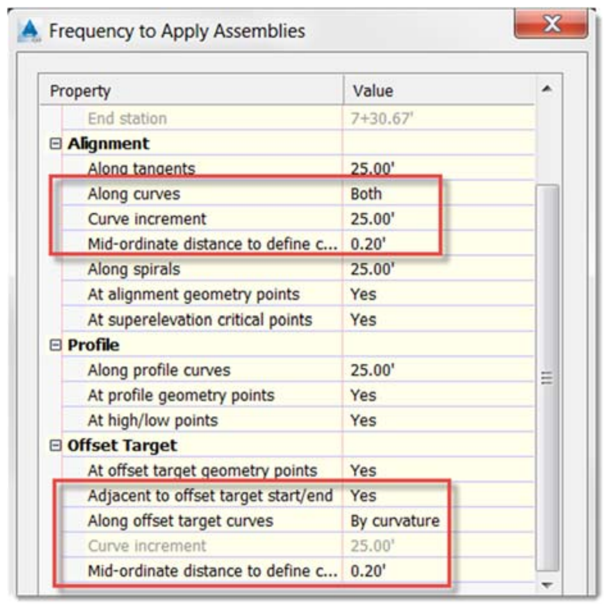
Other than avoidance, the only way to correct this situation is to split the corridor into regions and remove the region where the overlap occurs. Or you could assign an assembly that only contains the portions of the model that do not overlap. The remaining portion of the model can be completed with feature lines and grading objects.



Corridor with overlapping region replaced with feature lines and grading object.

- The 2015 version introduced some new corridor frequency options. You can now generate frequency points along arcs by specifying mid-ordinate distances. This method provides more detail to the corridor model by reducing the frequency distance in curves with shorter radiuses.

  To improve computer performance when working on large corridor models you should leave the frequency options for curves set to use only the incremental distance value. As the model is refined, you can experiment with the ‘by curvature’ options and reducing the mid-ordinate distances. Adjust them to values that result in the desired amount of detail in the model.



Managing Section Drawings

It is best to create the sample lines and cross sections in a separate file from the corridor model. The reason that you don’t want to put these items into your corridor drawing is that they will dramatically increase the time of the corridor rebuilding process that Civil 3D does whenever edits are made. Anytime you modify alignments, profiles or assemblies Civil 3D will need to rebuild the corridor, and then rebuild any corridor surfaces. If the cross sections are also in this drawing, Civil 3D will then update the sample line sections and all of the section views. These additional processes can increase the corridor rebuild time several minutes or more.

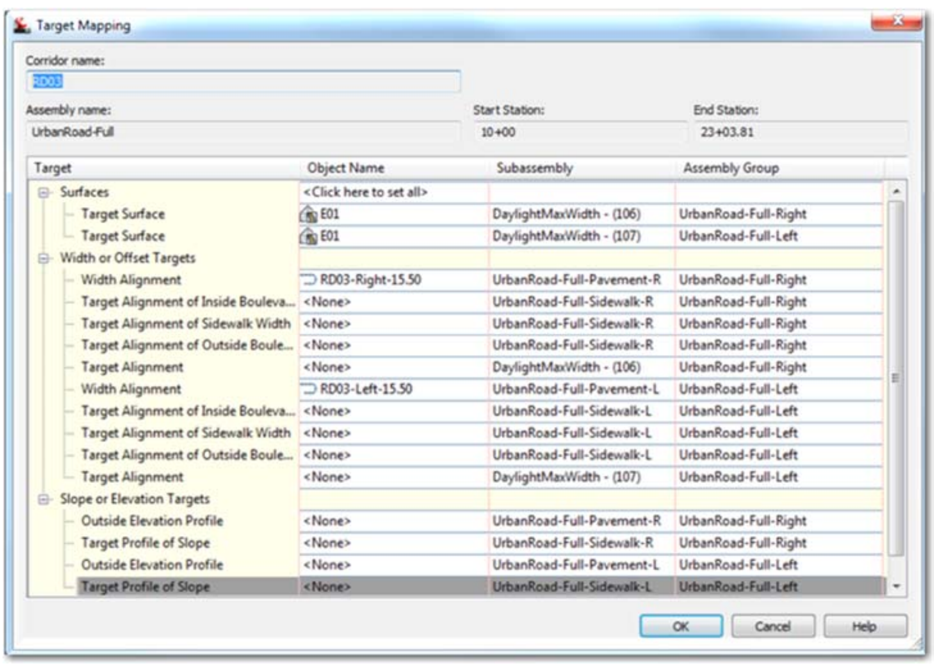
To create sections in a separate drawing you will need to data reference the base line alignment so you can define the sample lines. You will only need to xref the drawings containing the corridor and surfaces so that they can be sampled for sections.

To avoid confusion, do not data reference any of the surfaces that are within the drawings being externally referenced.

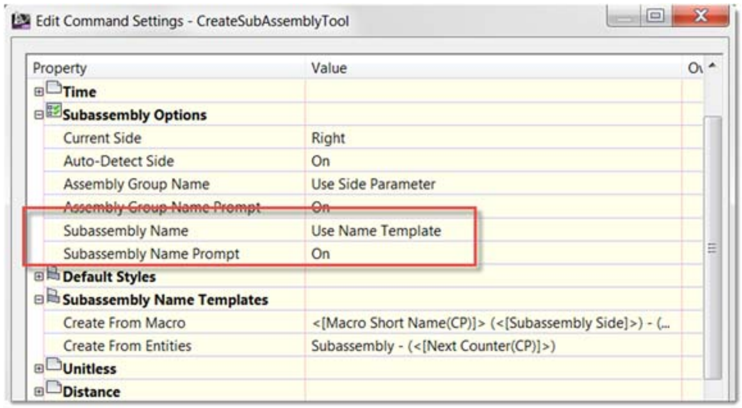
Assemblies

- Invest the time to establish a good naming standard for your subassemblies and assemblies. Be consistent in the application of these standards to your models. You will benefit because it makes the management of multiple assemblies, and the dozens of supporting subassemblies much easier.

  The best example of this is when you are mapping targets to subassemblies. A Good naming standard will allow you to easily identify the subassembly and which side of the alignment it is applied.

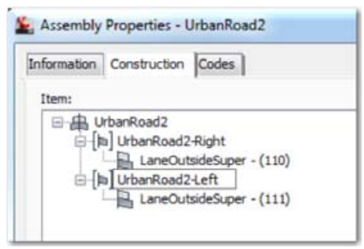


- Turn on subassembly name prompting. This will cause the program to prompt you for subassembly names as you add them to your model. These options can be found in the settings for the CreateSubAssemblyTool command.

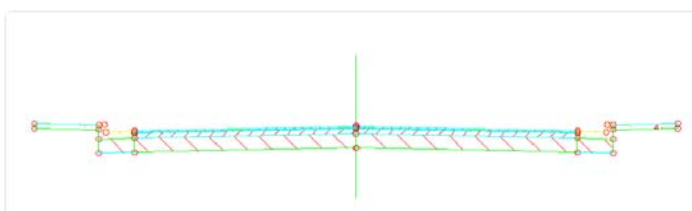


- Identify the basic assembly components that will apply to all regions of the corridor model. Create this assembly first. It can then be copied to create variations.

- Another thing that helps with subassembly management it to create right and left groups in the assembly. This is done by selecting the assembly object when applying the first right and first left subassemblies.

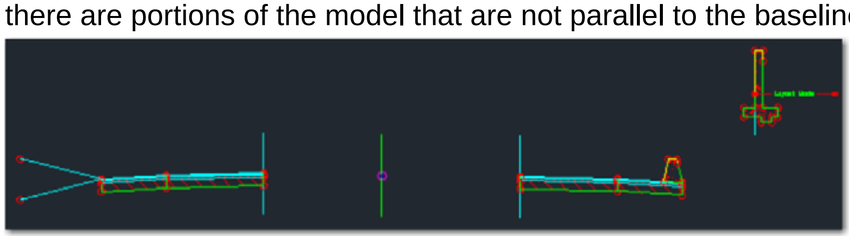


- Assign a code set to the assembly that displays point markers. This will make the attachment of subsequent subassemblies easier.



- Use stock subassemblies whenever possible. Don't create a custom subassembly when there are stock subassemblies available, or if you can achieve the same result with conditional subassemblies.

- Offset assemblies work well for models where you want to have a single baseline for the corridor, but there are portions of the model that are not parallel to the baseline.



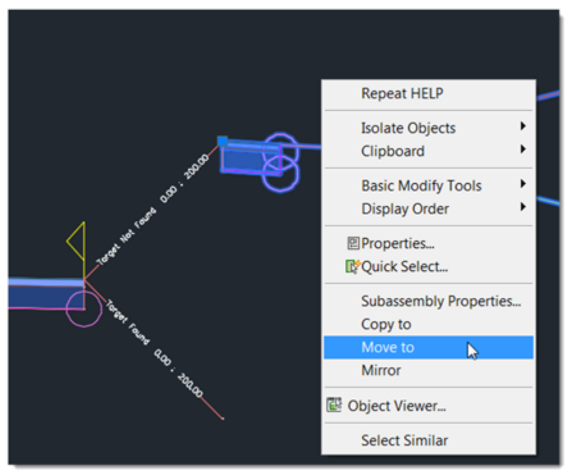
There were some issues with offset assemblies in the 2012 and earlier releases of Civil 3D. These problems appear to have been corrected in the 2015 version:

- In older versions of the software the offset portion of the assembly did not appear in the Corridor Section Editor command.
- In older versions of the software feature lines from offset assemblies were not created if the alignment assigned to the offset did not have the same station range as the baseline alignment
- Material volumes coulc not be generated for subassembly shapes that are connected to offset assemblies if the offset alignment did not have the same station range as the baseline alginment.

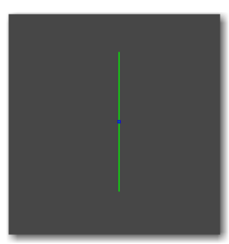
Prior to the 2012 version of Civil 3D, offset assemblies that were not raidal to the corridor baseline would not appear in section views. Since the 2012 release, Civil 3D no longer has this problem because it can now sample and display corridor shapes from any location on any corridor.

Some of the issues associated with offset assemblies can be mitigated by modeling the offset with generic links. The generic link can be hidden by setting its omit property to "Yes".

- Do not use the AutoCAD MOVE command to rearragne subassemblies because this my break the link between adjacent elements. This becomes noticable when portoins of the assembly do not follow target offsets or elevatoins. Use the Move To, Copy To and Mirror subassembly commands to ensure that they remain properly connected to adjacent subassemblies.

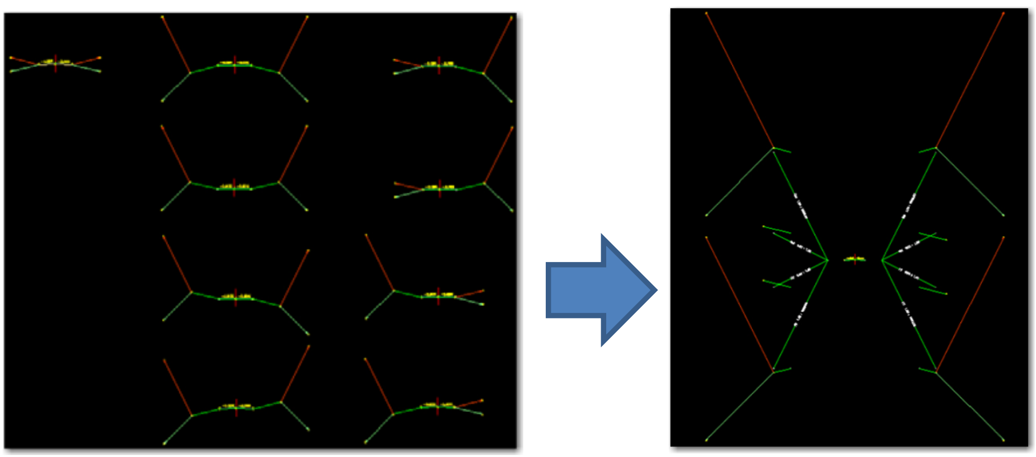


- A 'null' assembly can be used to force gaps in corridor regions where Civil 3D may still be connecting feature lines. This occurs sometimes in corridors created from intersections. A 'null' assembly is created without any subassemblies attached to it.



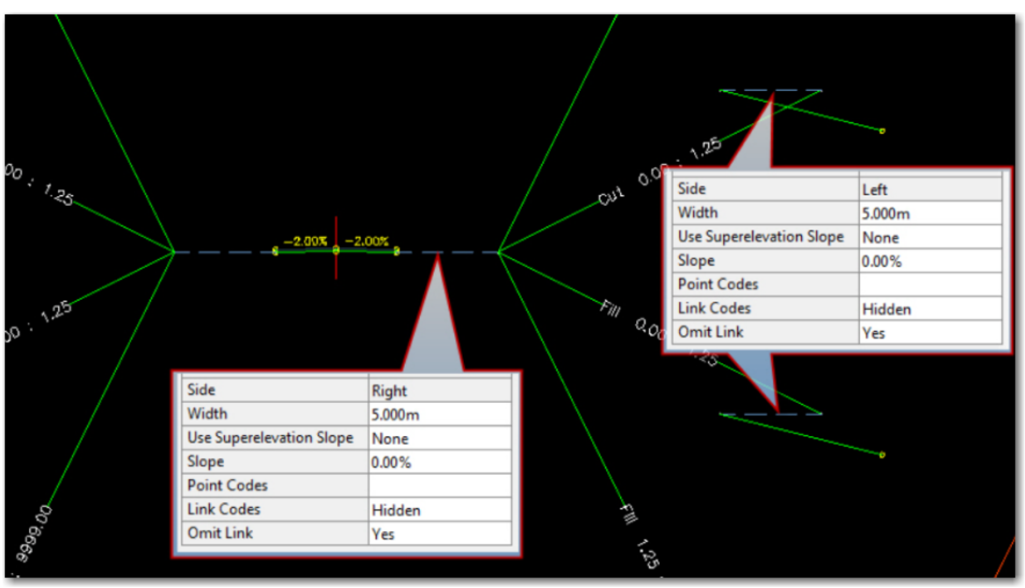
Conditional Subassemblies

Conditional subassemblies create a complex assembly that will apply different subassemblies based on testing the cut/fill depth, or the location of horizontal targets, at runtime. The function of a conditional subassembly is to let the program decide where to change assemblies. By using conditional subassemblies you will need fewer assemblies and fewer regions in your corridor models.



The above images show a conditional subassembly (right) that simplified a corridor model by replacing nine different assemblies and dozens of regions with a single assembly and one region.

You can use generic link subassemblies to locate where the cut/fill test conditions occurs. The following image shows an assembly configured to test cut and fill at an offset of 5 meters from the edge of the road.

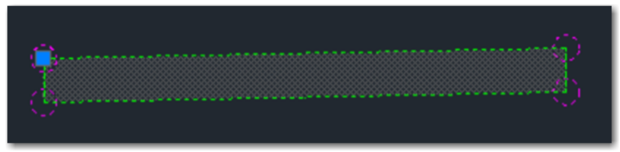


A generic width and slope subassembly with the Omit Link option set to “Yes” was used to locate the test condition 5 meters from the edge of road. Then, on each condition, another generic link was used to return 5 meters back to the edge of road.

Custom Subassemblies

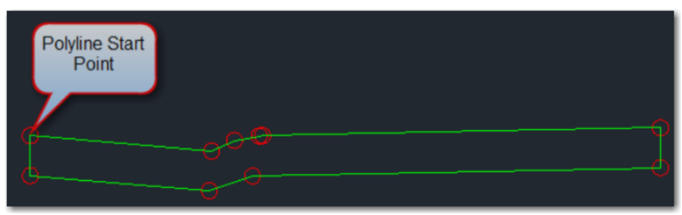
The ultimate power of corridor modeling can be realized by creating custom subassemblies. There are four main ways to create custom subassemblies. You can create static shapes with the MultiLink subassembly or by converting a polyline. For more advanced capabilities you can create them with Visual Basic .NET programs or the Subassembly Composer.

- The MultiLink subassembly is the easiest to use. It lets you create a static shape by entering delta x and delta y offset values. This subassembly is limited because the same code names get assigned to all points, and all links. You are not able to create separate Top and Datum links.



ADVANCED	
Parameters	
Version	R2012
Side	Right
No. of Links	4
Link Codes	Top,Datum
Point Codes	MultiLink_Point
Shape Codes	MultiLink
dW1	4.00'
dZ1	0.08'
dW2	0.00'
dZ2	-0.33'
dW3	-4.00'
dZ3	-0.08'
dW4	0.00'
dZ4	0.33'
dW5	0.00'

- You can also create static subassembly shapes from polylines. The origin point of the polyline will become the subassembly insertion point. This method is more robust than the MultiLink subassembly because you can assign unique code names to each point or link. For example, you will be able to assign a “Top” code name to the top links and a “Datum” code name to the bottom links.



- You can expand the power of corridor modeling by creating your own dynamic subassemblies with .NET. Prior to the 2011 version of Civil 3D this was the only way to create advanced subassemblies.

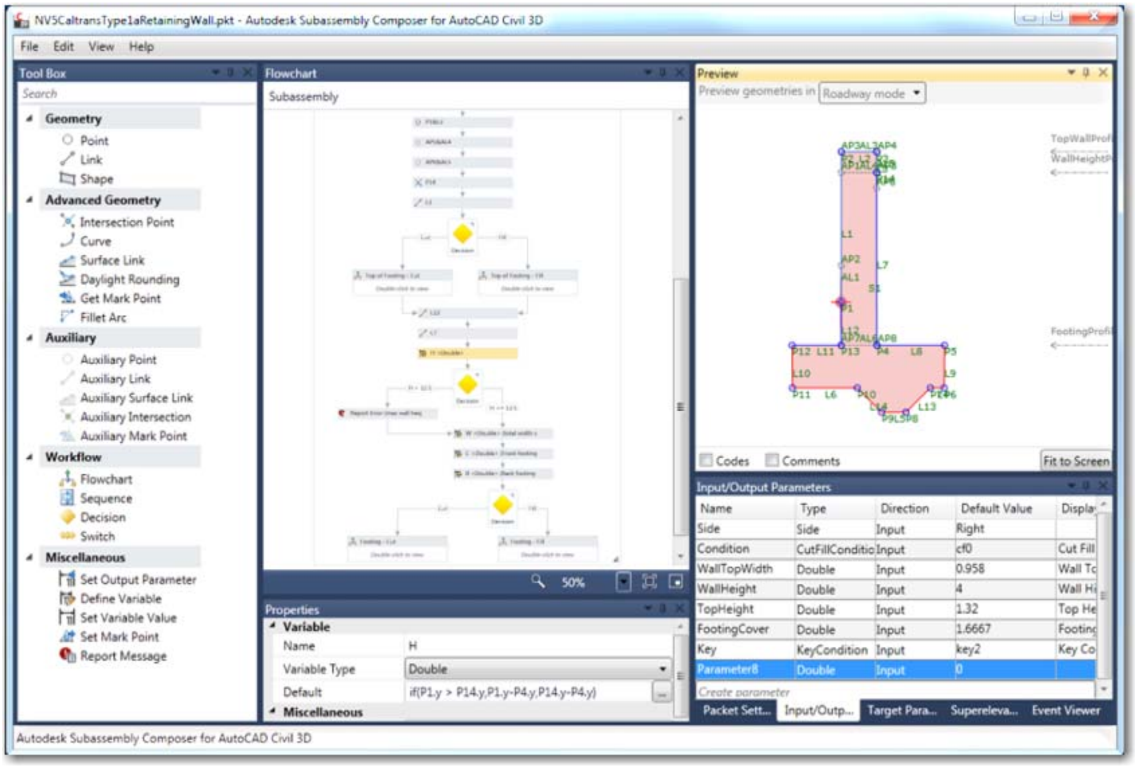
  However this is not easy to do. You will need to know how to write and compile Visual Basic dynamic link libraries. On top of that, you will also have to figure out how to create a catalog.xml file and add it to the Content Browser or Tool Pallet. And if that wasn't difficult enough, you will need to update your subassembly programs for each new release of Civil 3D.

  Custom subassemblies created with Visual Basic .Net can have custom variables and input parameters. Plus they can be linked to target surfaces, alignments and profiles, just like the stock subassemblies that install with Civil 3D.

  The most compelling benefit of using .NET to create subassemblies is that that you are able to access the entire programming interface provided with AutoCAD, Civil 3D and AutoCAD Map.   This provides unlimited capabilities to what can be done with custom subassemblies. This benefit can also be extended beyond the AutoCAD environment with .NET's ability to reference program libraries from other software. Such as being able to query spreadsheets and databases.

  However, there is a price to pay for this level of functionality. It can take about 20-40 hours to create a subassembly in Visual Basic .NET. Plus, you will also need to invest time to migrate your custom subassemblies to each new release of Civil 3D.

- Because of the complexities of learning and programming in .NET, most people do not leverage the power of custom subassemblies in Civil 3D. Recognizing this limitation, Autodesk created a program called the Subassembly Composer. It became available with Civil 3D 2011 and 2012, and has been updated with each following release.



The Subassembly Composer is a graphical programming interface. Subassemblies are created by dragging program elements from the tool box to a flowchart where they are sequentially linked. Element properties are modified to define point locations and how links and shapes are generated.

Immediate feedback is provided via the graphical preview window, eliminating the need to import the subassembly into Civil 3D for testing.

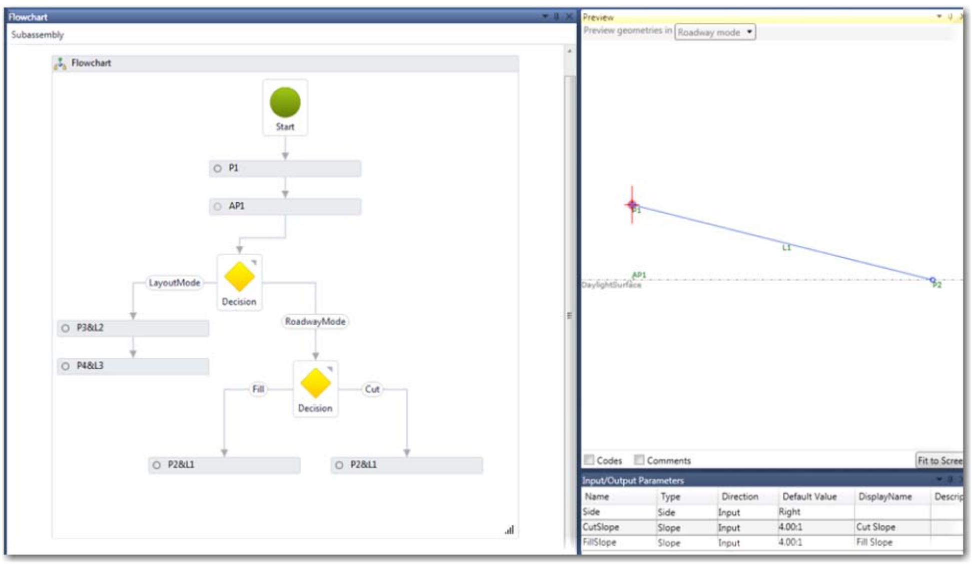
In addition to defining the point, link and shape geometry, there are tabs where input variables, target parameters and superelevation properties can be defined.

It does take some time to learn how to work with the Subassembly Composer, but it is nowhere near as daunting a task as learning VisualBasic.NET. Once proficiency is obtained, subassemblies can be created in less than half the amount of time required to create similar subassemblies with .NET. It takes approximately 2-4 minutes per element to create subassemblies in the Subassembly Composer. The retaining wall subassembly shown in the above image has 110 different, point, link, shape and parameter elements. It took 6.5 hours to create that subassembly with the Subassembly Composer. For comparison, it took more than 16 hours to create a similar subassembly with Visual Basic.NET.

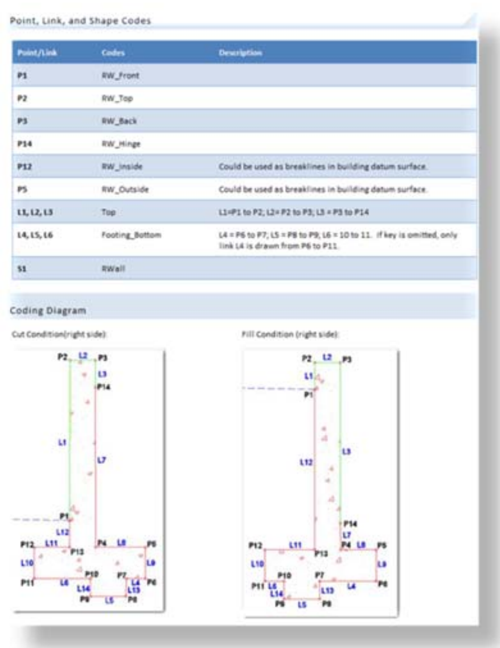
Additionally, the subassemblies created with the Subassembly Composer are easily integrated into Civil 3D with the Import Subassembly command.

Subassembly Composer Tips

- Start simple. The following image shows the first subassembly that I created with the Subassembly Composer. It is a simple daylight subassembly that allows the entry of negative slope values. This will model ascending slopes when the assembly is in fill conditions, and descending slopes in cut. It will work in most instances where the target surface is at a steeper grade than the subassembly link.



- Before creating your subassembly create a document that is similar to the help documents available with the stock subassemblies. Plan out how the subassembly will behave, list the input parameters and targets you will use, and sketch out the geometry with point and link numbers.


Point, Link, and Shape Codes

Point/Link	Codes	Description
P1	RW_Front	
P2	RW_Top	
P3	RW_Back	
P14	RW_Hinge	
P12	RW_Inside	Could be used as breaklines in building datum surface.
P5	RW_Outside	Could be used as breaklines in building datum surface.
L1, L2, L3	Top	L1=P1 to P2; L2= P2 to P3; L3 = P3 to P14
L4, L5, L6	Footing_Bottom	L4 = P6 to P7; L5 = P8 to P9; L6 = 10 to 11. If key is omitted, only link L4 is drawn from P6 to P11.
S1	RWall	

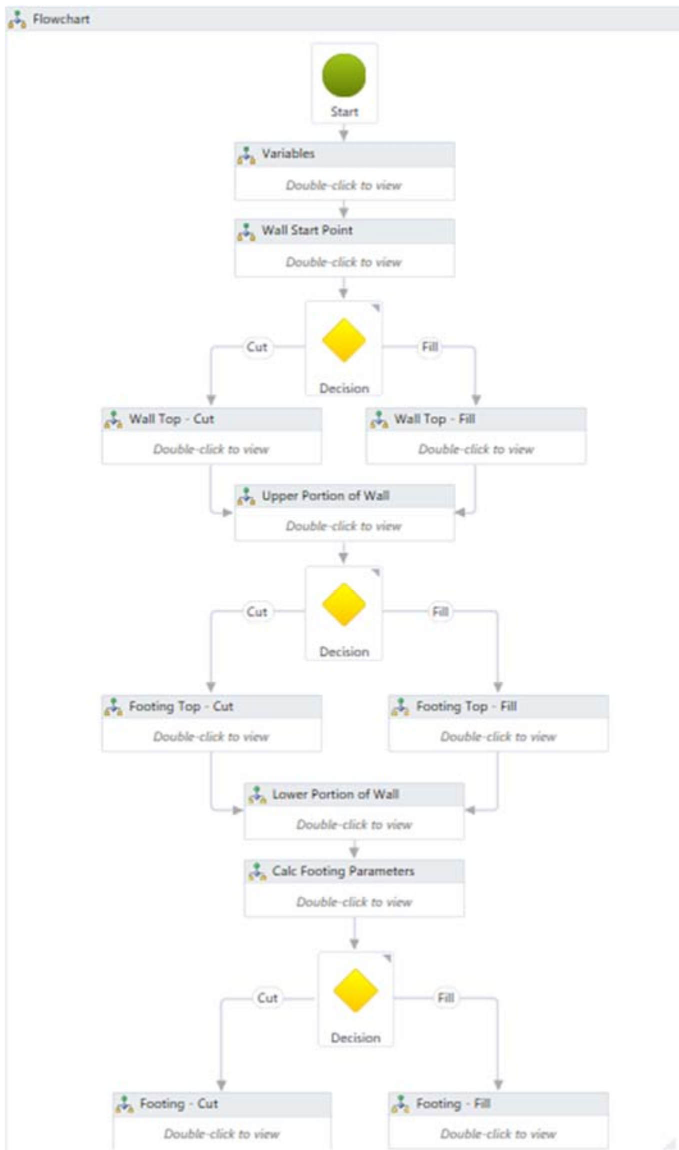
Coding Diagram

Cut Condition(right side):

Fill Condition (right side):

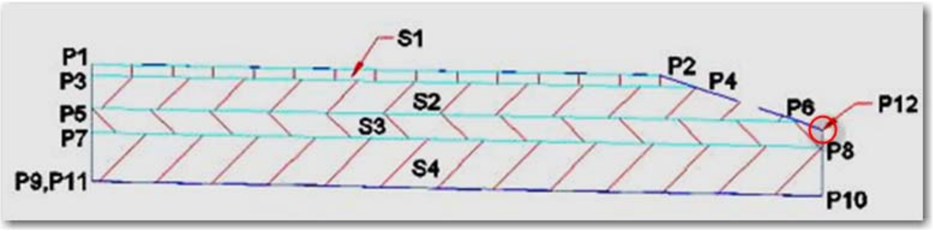
- Use the same code names that Civil 3D uses with the stock subassemblies. This will insure that your current code sets will work with any new subassembly that you create.

- Use imbedded flowcharts to organize and manage the components of the subassembly. Think of each flowchart as a subroutine that generates a specific portion of the model. Variations can be quickly created by cutting and pasting flowcharts within the assembly.

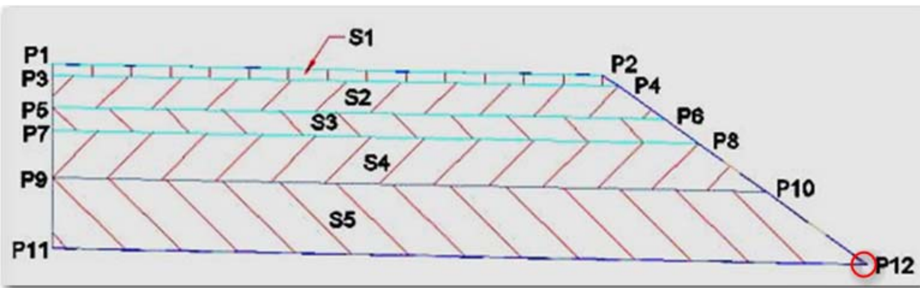


- If you have conditions that change the shape of the subassembly, make sure that you use the same point number for the outside point where other subassemblies will be attached.

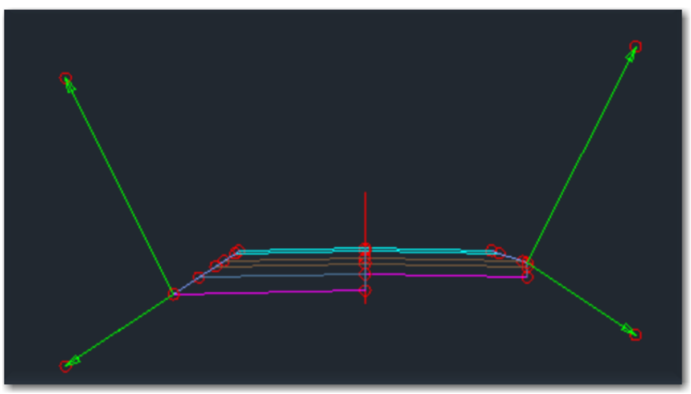
The following images show a highway subassembly that has different layouts for cut and fill. Point number 12 was assigned to the ditch flowline point while in cut and the toe of the shoulder slope while in fill. When a daylight subassembly is attached to these locations in layout mode, it will shift to the correct position as the corridor alternates between cut and fill conditions.



Point codes in cut.

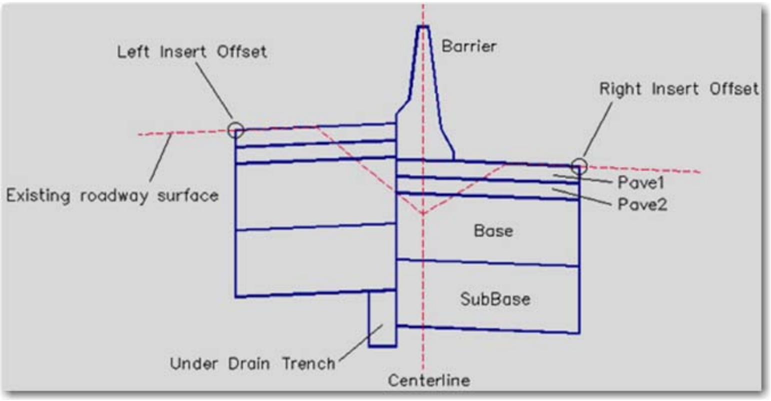


Point codes in fill.

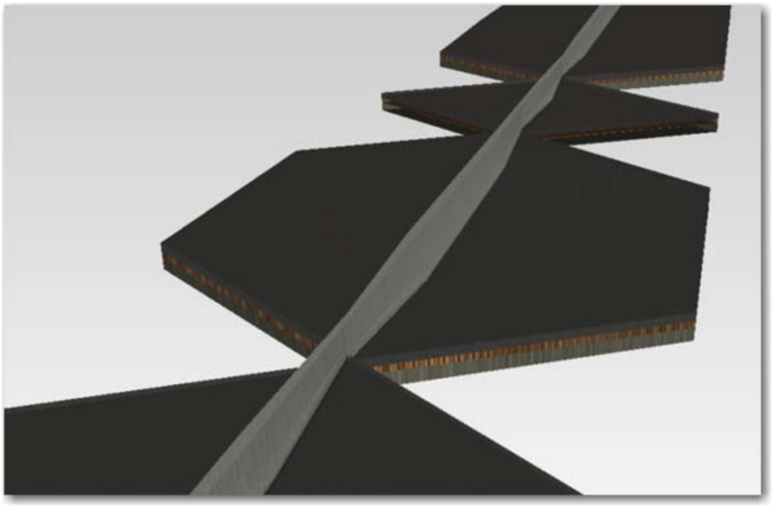


Subassembly in layout mode with daylight subassemblies attached.

- Don't use variables that will mirror asymmetric components. The following images show a median widening subassembly that was created from the stock OverlayMedianAsymectrical subassembly. This subassembly calculates the point offset locations based on the right side of the highway being lower. If the left side is lower, it multiplies the offset values by negative one. When viewed in cross sections this subassembly will look fine. But when feature lines or solids are extracted they will crisscross as the low side condition alternates between left and right.



Subassembly layout based on right side being lower.



Resulting solids created from a “mirrored” subassembly with the Corridor Solids Extractor.