

Creating Site Plans (Part I)

Visualization Insider

Creating 3D Site Plans – Part I

Prerequisites – a foundation level understanding of 3ds Max and an intermediate level understanding of AutoCAD.

Introduction

Creating a 3D site plan can seem like a daunting challenge and can easily frustrate experienced users. For many years, I have experimented with virtually every possible way to create site plans, and the chances are that if you've been frustrated with a particular plan of attack, I have probably experienced a similar frustration when trying the same thing. I now find myself using the same steps and tips and tricks on just about every site I work on, with minor variations in site elements specific to each piece of work. With a little practice using these routines, anyone can breeze through constructing a solid site foundation in minimal time.

In this tutorial, I wanted to demonstrate as many tips, tricks, and routines as possible, spending minimal time on repetitive steps that waste time without teaching anything new. Therefore, the site plan used in this tutorial may appear overly simplified. For example, lines that represent streets are often poorly constructed and contain countless breaks, rather than being continuous lines. Rather than giving you lines that require hours of editing and welding to fix, I have provided near perfect linework, with only a few imperfections that allow me to demonstrate problems usually commonplace to site drawings. This will allow the tutorial to progress rather quickly in comparison to real-world site planning, and therefore speed up and optimize learning considerably. The following image shows an example of how the more 'typical' plan is cluttered with dozens of layers of information that is simply not needed in the development of a 3D site.

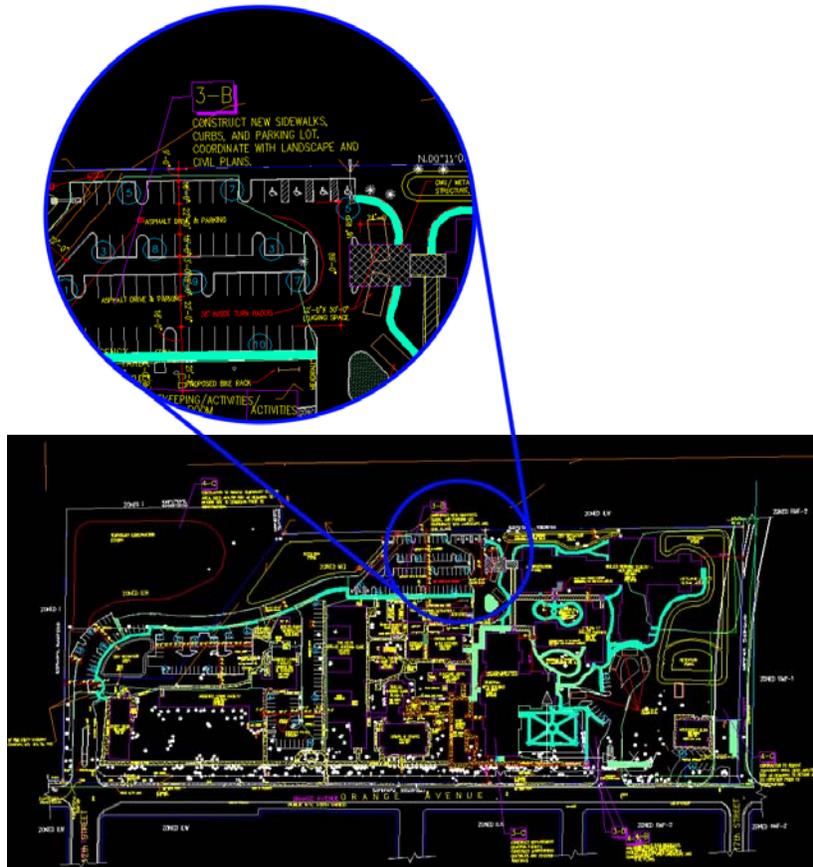


Figure 1: A typical 'dirty' site drawing

Once the unnecessary linework has been removed, and it has been modified for import into 3ds Max, the drawing would look something like this. Notice the lack of dimensions, notes, and numerous layers of extraneous data such as drainage callouts and surveying coordinates. None of this information is necessary to create a 3D site.

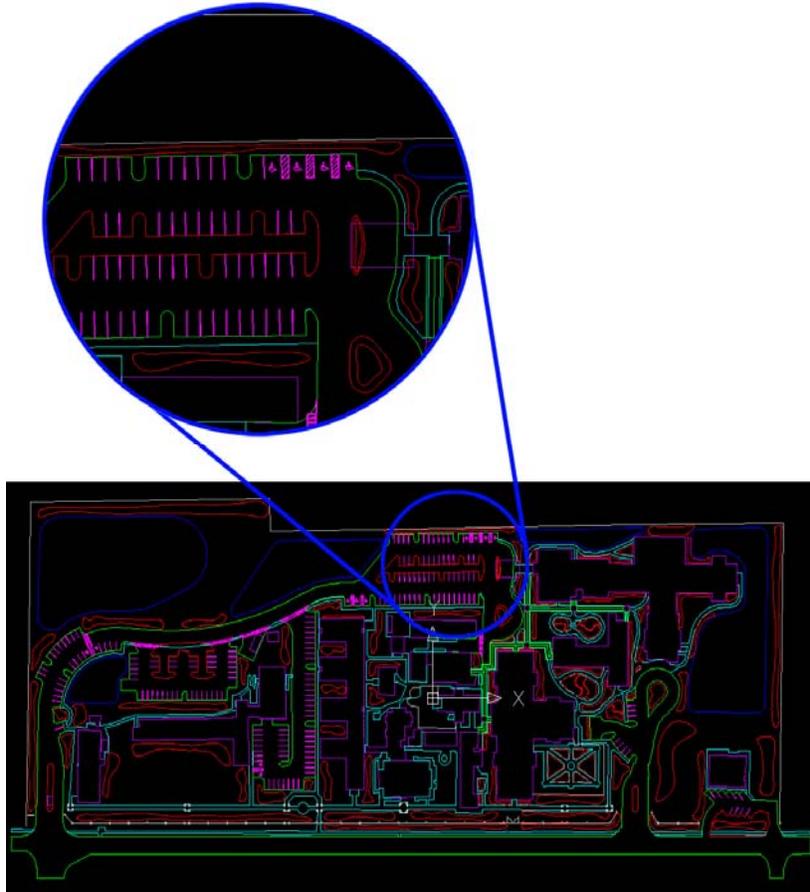


Figure 2: A 'cleaned' site drawing

Almost every project I've ever worked on has started with AutoCAD drawings and site plans are no exception. This tutorial demonstrates a project starting from AutoCAD, however, the same concepts should apply quite nicely to any CAD program. Ironically, the majority of work performed in the creation of a 3D site should usually be performed in a CAD program, such as AutoCAD. Remember the saying, 'Garbage in equals garbage out.' This saying couldn't be more applicable to 3D site plans in 3ds Max. With good linework to import into 3ds Max, one could easily build a very complex 3D site in well under an hour. With poor linework, however, the same task could take countless hours and often lead to very inaccurate models. I have often spent an entire day preparing linework for import into 3ds Max, and turned the linework into a complete 3D scene in 15 minutes or less. Of course, you always have the option of importing drawings straight into 3ds Max 'as is', however, I highly recommend avoiding this approach for numerous reasons, not least because AutoCAD has superior line editing capabilities over Max, so doing this step in AutoCAD will lead to much better overall results.

So throughout this tutorial, I will stress the importance of good CAD linework and continuously reference cases where problems arose from poorly created lines. I use the term 'clean line' to indicate a line or spline that is free from any defects that would cause problems or errors in 3ds Max. The term 'dirty line' is used to indicate a line or spline that is created or modified in a way that does contain defects that prevent its use in the creation of 3D objects.

There are several reasons why a line may be 'dirty', and hence, not work properly when imported into 3ds Max. The following is not an absolutely complete list, but will explain the vast majority of the reasons why linework is unable to work properly when imported into 3ds Max.

Overlapping lines – this occurs when a line crosses over itself, as shown in the left image of Figure 3. When converted to 3D, this type of line often results in a solid that appears hollow or is missing a top and bottom.

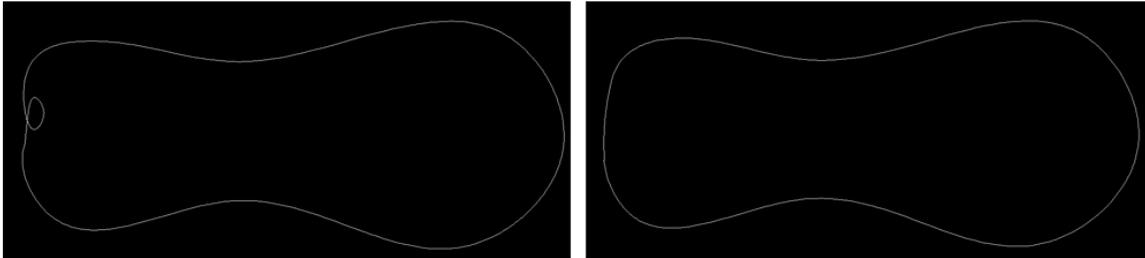


Figure 3: A self-overlapping spline, left, and a properly structured spline, right

Broken lines – this occurs when lines that should be continuous are broken into 2 or more individual segments, as shown in Figure 4. These segments may have endpoints close together, or they may be separated by a significant distance. If the end points are close enough together, they may be automatically welded when imported into 3ds Max, however, it is usually far more effective to identify these broken lines and weld them in AutoCAD. Waiting to fix this problem in 3ds Max will almost certainly take longer because of 3ds Max's slower and less efficient line editing capabilities. Perhaps more importantly, if you leave the linework 'dirty' in AutoCAD, you may have to repeat the process of fixing it in 3ds Max if you need to re-import for any number of reasons.

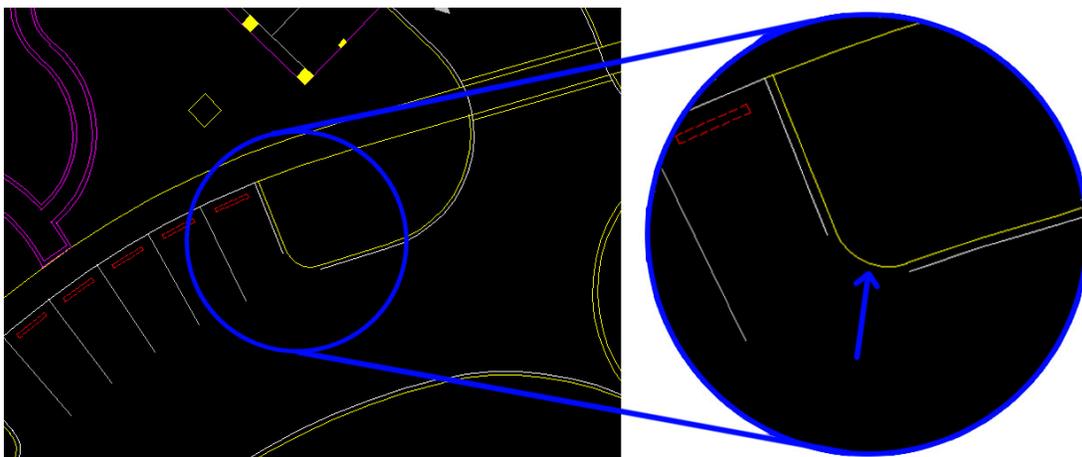


Figure 4: A broken line representing a curb

Different layers – this occurs when lines that should be on the same layer are placed on separate layers, as shown in the multi-layered lines that represent the curbs in the image below. When isolating individual layers, this problem can lead to the user not realizing that he or she has just hidden information (on the wrong layer) and result can be disastrous..

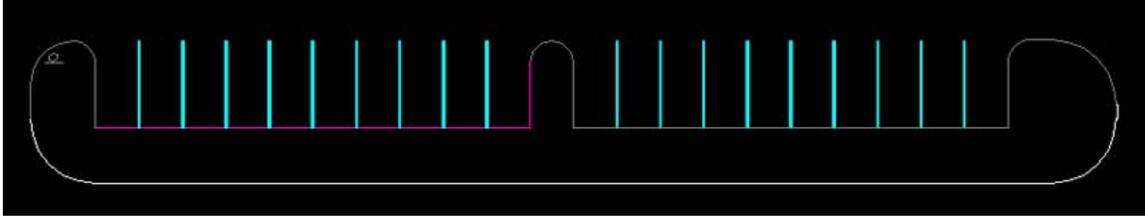


Figure 5: The same object represented by lines on different layers

Varying elevations - one of the most frustrating things about working with drawings somebody else created is the eventual occurrence of line segments existing on varying elevations. From a top view, these lines may appear to be otherwise 'clean', but the truth soon comes out when you view your linework from a perspective view, as shown in the right image of Figure 6, and endpoints at different elevations can never be welded. The only way to fix this problem and use the linework for 3D purposes is to place the endpoints on the same elevation so that the endpoints can be welded.

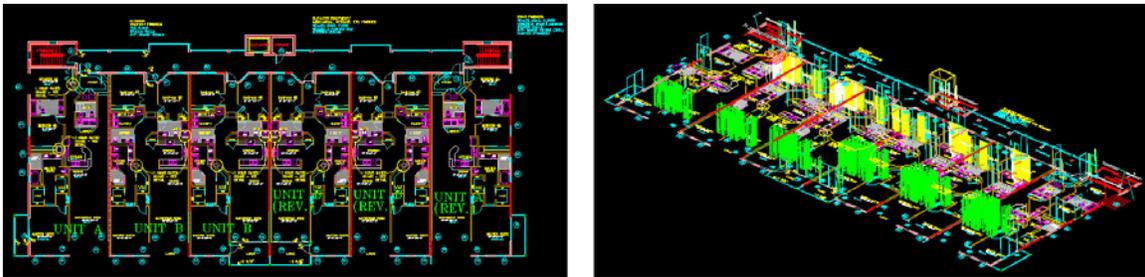


Figure 6: Various lines on different elevations

This tutorial covers the creation of some of the most prevalent types of object found in site drawings. Specifically, each of the following object types are included:

- Terrain
- Roads
- Road lines
- Curbs
- Pavers
- Paver Bandings
- Sidewalks
- Bodies of water
- Parking stops
- Mulch beds
- Vegetation
- Backgrounds

Obviously, any given scene may need to be populated with miscellaneous object types such as cars, people, signs, etc, however, these objects should be pulled from your pre-existing libraries and not created from scratch.

The Uno Chicago Grill

In this scene, I have returned to a project that I created in 2004 for the restaurant chain Uno Chicago Grill. This project called for a 30 second commercial featuring 25 seconds of 3D animation, half from an aerial perspective and half from a ground level perspective, as shown below. The raw animation can be seen at the following link:

<http://3das.com/images/gallery/movies/PizzeriaUno.mov>



Figure 7: Two images from the final animation

This project contains a single building surrounded by a paved walking area, a large number of parking spaces, and examples of all the other objects previously mentioned. Though a few years old, the image below was pulled for Google Earth and shows what the project looked like during the start of construction. As you can see, the real site matches the 3D site rather nicely.

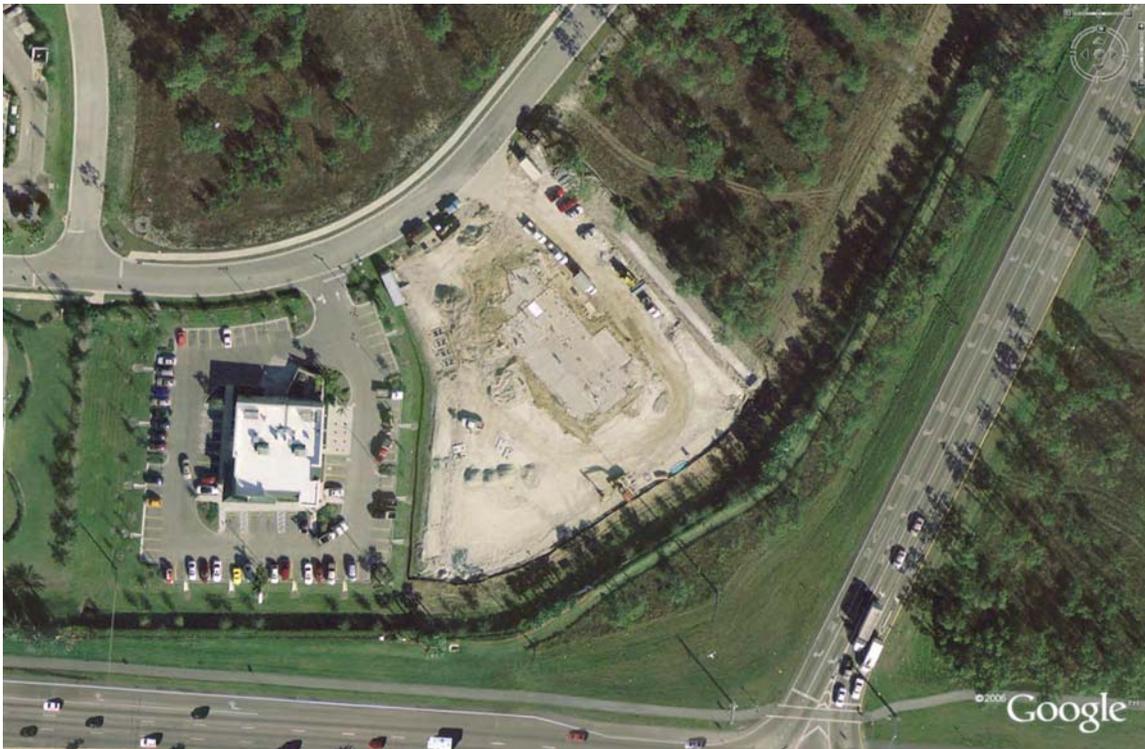


Figure 8: Satellite imagery of the project site during construction

So, like most 3D projects, this project started with a CAD drawing, as shown below. This particular drawing, **Pizzeria_Uno.dwg**, was actually one of the 'cleanest' site drawings I've ever worked on, yet it still contained many of the problems that plague our work in 3D. There was very little extraneous information needing deletion, such as sewer lines, dimensions, etc, and overall, the linework was easy to understand. However, several very big problems existed that required hours of cleaning. To start with, all of the project linework resided on the same layer, so nothing could be isolated. Second, there were no continuous polylines already prepared for export into 3ds Max, so nearly every line in the drawing needed attention.

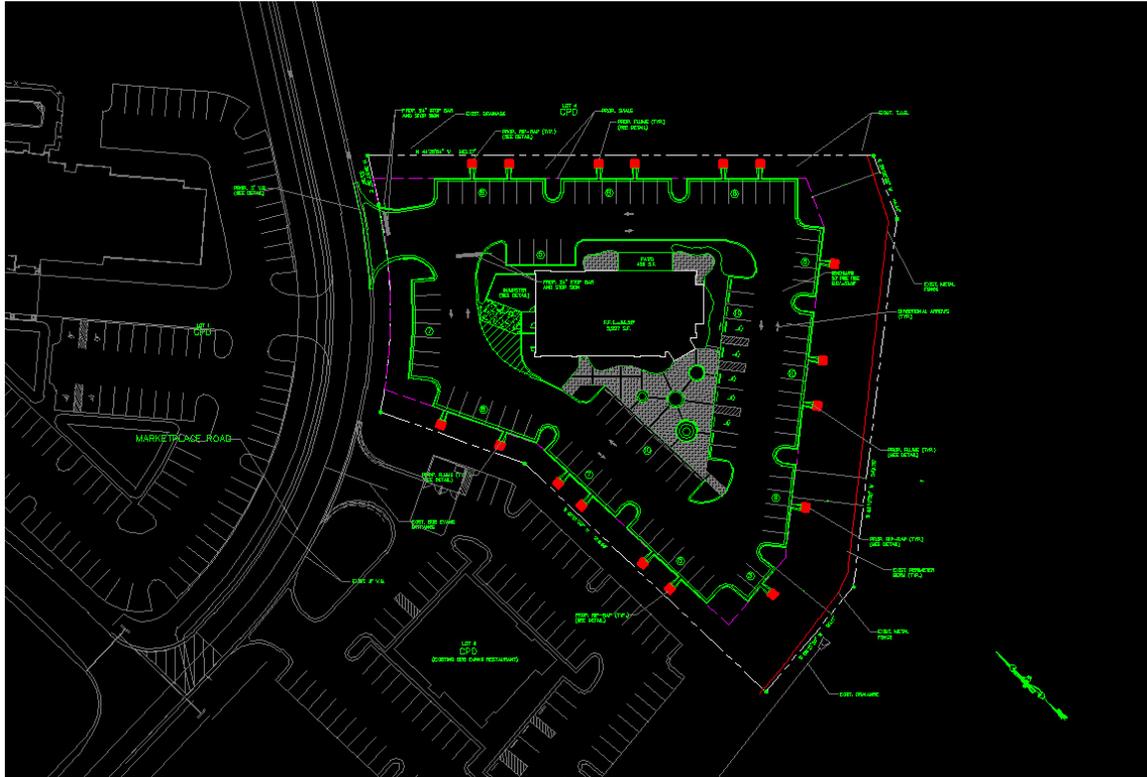


Figure 9: The 'dirty' AutoCAD site drawing for Uno Chicago Grill

So the first step in the creation of a 3D site plan is to 'clean' the drawing and prepare it for use in 3ds Max. If you would like to try creating this scene from this initial site drawing, start with the file labeled **Uno_initial.dwg** and try to prepare the linework in a way similar to that found in the file labeled **Uno_cleaned.dwg**. Preparation of this file took approximately 2 hours and the remainder of this tutorial continues with the cleaned version of this site drawing, shown below.



Figure 10: The 'cleaned' AutoCAD site drawing for Uno Chicago Grill

Now that the 2D linework is prepared, we can begin the creation process for each of the elements that will make-up the 3D site. The same methods that I cover here can be applied to just about any 3D site and whenever possible I will point out other methods of performing a given task, which may work better for a particular type of site.

It is important to note here that the linework to be imported into 3ds Max can be imported all at once or one layer at a time. In this tutorial, I chose to import all linework at the same time, however this is a matter of personal preference and in more complex sites, I would recommend importing one layer at a time. This will allow you to work on one site element at a time and not clutter your view and workspace with linework that you are not currently working on. It also helps you to perfect one type of object before starting on the next. Perhaps most importantly, it gives you time to get a 'feel' for the project think about how to create other elements later on while working on some of the easier elements that you know won't be as difficult.

Before going any further, I would like to point out a very important detail about AutoCAD drawings and how they work when imported into 3ds Max. If you look in the original drawing, **Uno_initial.dwg**, you'll notice that the origin is located a great distance away from the linework of this project. In the cleaned drawing I moved all of the linework in the project so that it is centered on the origin. This is important because, unlike AutoCAD, the accuracy of 3ds Max is greatly dependent on where you operate in the world-space coordinate system. If you are not familiar

with why this is or are not completely confident with how to setup the 3ds Max work environment, please read the following section, which is excerpted from my book, [Foundation 3ds Max 8](#),

Setting up the work environment

Setting up your work environment before beginning any project can save you valuable time and prevent unnecessary grief. Although there are an enormous number of settings that control the behavior of the 3ds Max interface, there are only a few important ones that you need to know about in your day to day work. They include units, display drivers, configure path settings, and select preference settings. Some of these need to be changed for each new scene, while others need to be changed only after 3ds Max is installed.

Units Setup

An important setting that should be looked at before you even start setting up a scene is the **System Units** setting—this setting dictates what the true value of any number entered or displayed really means. Since you will be working on architectural visualizations, it only makes sense to use architectural dimensions. Also, since much of your work will originate in AutoCAD with drawings set to architectural dimensions, there's no reason to set your units to anything else.

To change the unit settings, choose **Customize > Units Setup**. The **Units Setup** dialog box (shown in Figure 11) opens with a default **Display Unit Scale** of **Generic Units**. Check **US Standard** and ensure the drop-down menu is set to **Feet w/Decimal Inches**. These are display settings, which means any value displayed in 3ds Max is expressed in feet and decimal inches, or ' and ".

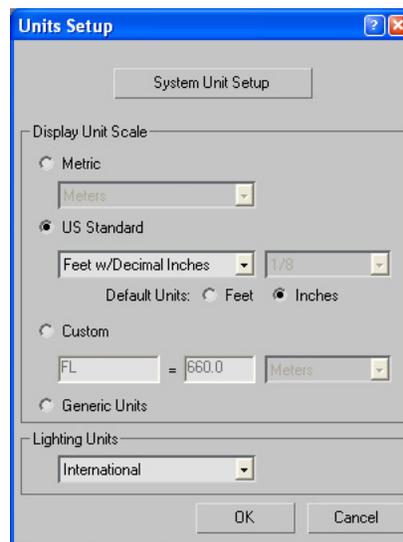


Figure 11: The Units Setup dialog box

Next, ensure **Default Units** is set to **Inches**, as shown in Figure 11. This means that any number entered into any field will be considered a number expressed in inches if it isn't followed by a unit designation. Therefore, if you change the radius of a sphere to 20 without following it with an apostrophe, the radius becomes 20 inches. But because the display settings are set to **Feet w/Decimal Inches**, the radius will actually change to read 1'8". You can also enter 1'8" yourself. Remember that an apostrophe (') indicates feet and a double quote mark (") indicates inches.

The reason I suggest changing the **Default Units** to inches is that when you want to enter a certain number of inches into a field, such as 4.5", you won't have to do any math to figure out what the number is when expressed in feet (0.375' in this case)— it simply saves time. In addition, **AutoCAD** uses inches as its default units, and it can be confusing when working in two programs with two different default unit types.

Last, click the **System Unit Setup** button and ensure that the **System Unit Scale** is set to **1 Unit = 1.0 Inches**, as shown in Figure 12. Changing this to read **1 Unit = 1.0 Feet** will cause your entire scene to be scaled up twelve fold. The objects won't actually change in size; they will simply be given new units. A sphere that was expressed as 5 inches will now be expressed as 5 feet.

This also means that when you import an AutoCAD drawing with default units set to inches, all imported objects will retain their original size. If you change your **System Unit Scale** to read **1 Unit = 1.0 Feet**, then a sphere with a radius of 5 inches in AutoCAD will change to a sphere with a radius of 5 feet when imported into 3ds Max.

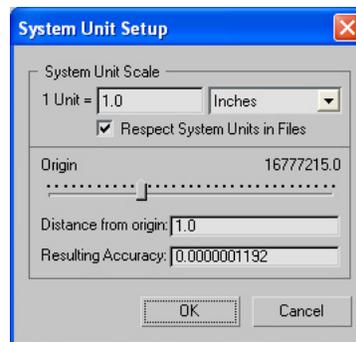


Figure 12: The System Units Setup dialog box

System Units Setup

There are no other settings that need to be altered within the **Units Setup** dialog box; however, I would like you to take note of the bottom half of the **System Unit Setup** dialog box. These values control how accurate 3ds Max is in the use of numerical values. Basically, the closer objects lie to the origin of the World Coordinate System (i.e., 0,0,0), the more accurately objects are displayed and parameters computed. The values in this dialog box indicate that objects would have to be more than 16 million inches away from the origin in order to be inaccurate to an inch. Although this may seem to be more than accurate enough, problems can still arise, as I once learned the hard way. I was once knee-deep in the modeling of a very large project for a golf course community with site elements (streets, lakes, etc.) that I created straight from the line work of an engineering firm's site plan that I imported from AutoCAD. I inserted all my individual objects, such as houses, streetlights, cars, etc. What I didn't realize is that for some reason the drafter had placed all the line work in the AutoCAD file at an incredibly great distance from the origin. So when I inserted all the objects, they too were a long way from the origin and were no longer accurate. I didn't catch it at first, and couldn't figure out why all my modeling operations were messed up. With 3ds Max being 1 inch off at a distance of 16 million inches from the origin, you can't move or modify objects in increments less than one inch. This won't cut it for architectural models. So now the absolute first thing I do with any AutoCAD drawing that I'm going to use is move all line work so that it's centered on the world origin. The moral of the story is to always keep your scene centered on the origin.

One final and very important note about units is that engineering firms typically produce site drawings for architects using engineering scales. In engineering scales, the unit of measure is 1

foot, which means that an object 12 feet long will actually appear only 12 inches long when its line work is imported into 3ds Max. Therefore, you must scale site plans 1200% prior to or immediately after importing.

Importing the 2D linework

When you prepare linework in CAD, it is important to figure out ahead of time which scene elements are going to be combined into the same scene objects. Minimizing the number of object types will reduce the number of times you have to perform the same 3D operations in 3ds Max and can greatly improve speed and efficiency. For example, if you had two separate lines that represented the same type of sidewalk that would receive the extrude modifier, it wouldn't make sense to have those lines on separate layers in AutoCAD, import them as separate objects, and extrude them separately from each other when you can combine both lines on the same AutoCAD layer and perform the procedure just one time.

In this project, I have simplified the AutoCAD drawings so that only 7 unique layers exist besides the default '0' layer, as shown below. This allows you to turn the linework into a 3D site much more quickly. I like to retain very strict layer management to aid efficiency and workflow. I give all my layers very descriptive names so that there is no doubt what is contained on them and I go to great lengths to ensure that all linework is placed on the appropriate layer. Notice that all the layers are preceded with the prefix '0-'. This is a personal technique I use to keep my own layers at the top of the layer manager. I don't like spending time looking through sometimes hundreds of layers just to find a particular layer of my own. I also categorize my layers by giving them an additional prefix, such as 'Site' or 'Bldg'. This allows me to quickly and easily isolate all my building or site linework in AutoCAD as well as in 3ds Max.

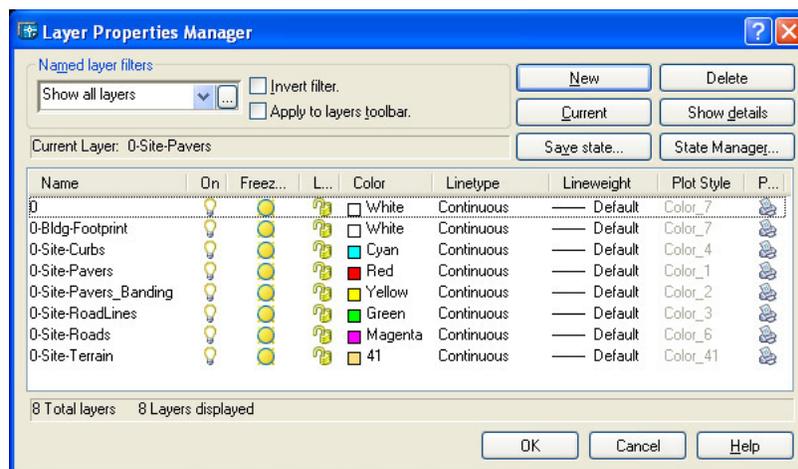
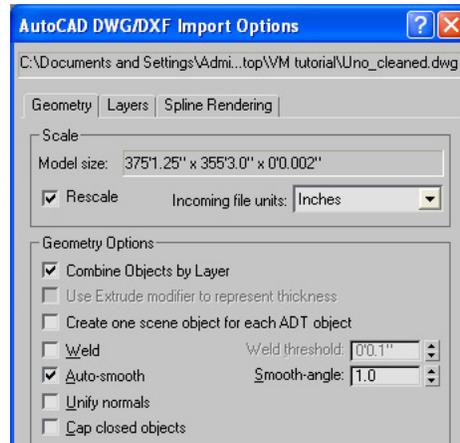


Figure 13: The layers of the cleaned site drawing

Now that we have done everything possible to streamline our work in 3ds Max, let's go ahead and import the linework.

1. In 3ds Max, open the **File** menu and select **Import**.
2. From the **Files of Type** drop-down list, select **AutoCAD Drawing (*.DWG,*.DXF)**.
3. Select the file named **Uno_cleaned.dwg** and click **OK**.
4. The AutoCAD DWG/DXF Import Options dialog box opens.
5. Ensure the **Combine Objects by Layer** option is enabled, as shown below, and click **OK**. This ensures that all objects on the same layer are combined as one object. As a

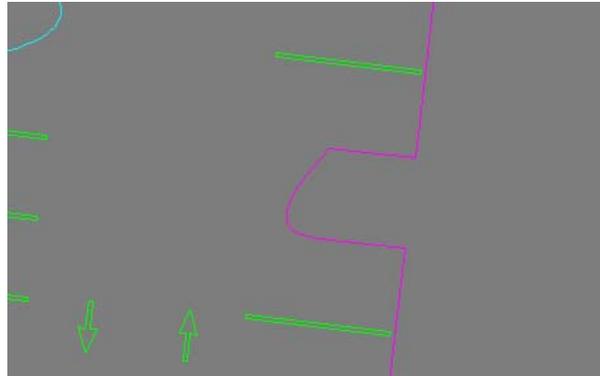
side note, you could use the **Rescale** option here instead of setting up the scale through the **Units Setup** dialog box.



6. Click the **Zoom Extents All** icon in the bottom-right corner of the screen. The scene should look like the image below.



7. Zoom into any area containing curved lines, such as the line representing the curbs in the image below. You may notice some very noticeable irregularities with the curves of many lines. In this case, the curve is completely deformed from the way it is shown in AutoCAD. If this happens, it is probably because of the file type you are telling 3ds Max to import—the way to fix this is to reimport the linework as a different file type, as shown below.



8. Delete all of the linework you just imported.
9. From the **File** menu, select **Import**.
10. From the **Files of Type** drop-down list, select **Legacy AutoCAD (*.DWG)**.
11. Select the file named **Uno_cleaned.dwg** again and click OK.
12. The DWG Import dialog box opens.
13. Select **Merge** objects with scene and click **OK**. The **Import AutoCAD Drawing File** dialog box opens.
14. Ensure that **Derive Objects by Layer** is enabled as well as the **Weld** option. Click **OK** to complete the command.
15. The linework is once again brought into 3ds Max, but this time without any imperfections in the curves.
16. Click the **Zoom Extents All** icon.
17. Save your file as **My_Uno_01.max**. It is good practice to save incrementally and save often. Each time we save, we will increase the number at the end of the file name.

Terrain (Part I)

The first rule I like to stress in creating terrain is to 'keep it simple'. Users often make the mistake of over-complicating what could otherwise be a very simple piece of terrain by trying to perfect the lay of the land. Almost no site in the real world is completely flat, and therefore, when you receive a drawing originating from a surveyor or some other type of civil engineering firm, the chances are pretty good that you will find topographical lines that show some variation in the elevation of the terrain. However, when looking through the lens of a camera, whether in the real world or in the virtual world, you will often not see the subtle variations in elevation that would justify the use of 3D terrain. For instance, if a parking lot had a 6 inch downward pitch towards its center to allow for water to flow into a drainage feature, this is something that you shouldn't ever have to worry about modeling. Creating this simple little change in elevation can bring about a number of different problems that would otherwise not have to be addressed if the parking lot were to remain completely flat. In this scenario, one would have to ensure that the parking lines conformed to the slope of the parking lot and that cars were positioned just right so that their tires weren't seemingly buried in the pavement or floating a few inches above it. For such small variations in terrain, it is far better to avoid the complication of accommodating this kind of detail. At the end of this tutorial, I will present some ways of creating 3D terrain, but for this particular site we are going to simplify things and create the terrain as a flat piece of land.

1. Continue from the previous exercise or open the file **My_Uno_01.max**.
2. In the **Top** view, draw a rectangle that fits just inside the view and completely encompasses the project linework, as shown below.
3. Name the rectangle **Site-Terrain**.



4. In the **Command** panel, add the **Edit Mesh** modifier. This changes the rectangle into a renderable mesh object.
5. Save your file as **My_Uno_02.max**.

For now, this will serve as the terrain in this scene, although later on, when we create the roads for our scene, the terrain will look very different; at that point we will continue with Part II of creating the terrain, but for now it's time to add curbs to our scene.

Curbs

There are numerous types of curb used in the real world; in this project there are two unique types, both of which are shown below. For the sake of simplicity, I have labeled these curb types as type A and type B.

Type A is used along the outer perimeter of the parking lot and is beneficial for allowing rain water to drain into the drainage basins surrounding the parking lot. In this project, it has an overall cross-section width of 2'0", which includes a 6 inch wide raised portion that retains the surrounding topsoil.

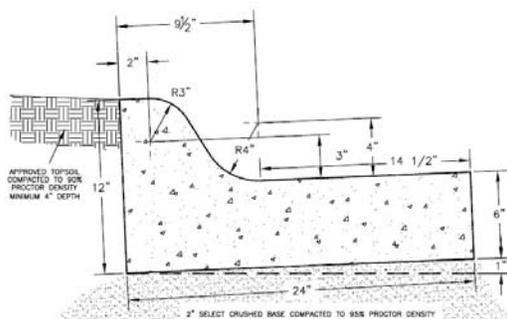


Figure 14: Curb Type A

Type B is a curb with a simple rectangular cross-section and an overall width of 6 inches. It is used in the inner island area immediately surrounding the building and separates the grass from the road in some areas and the pavers from the road in other areas.

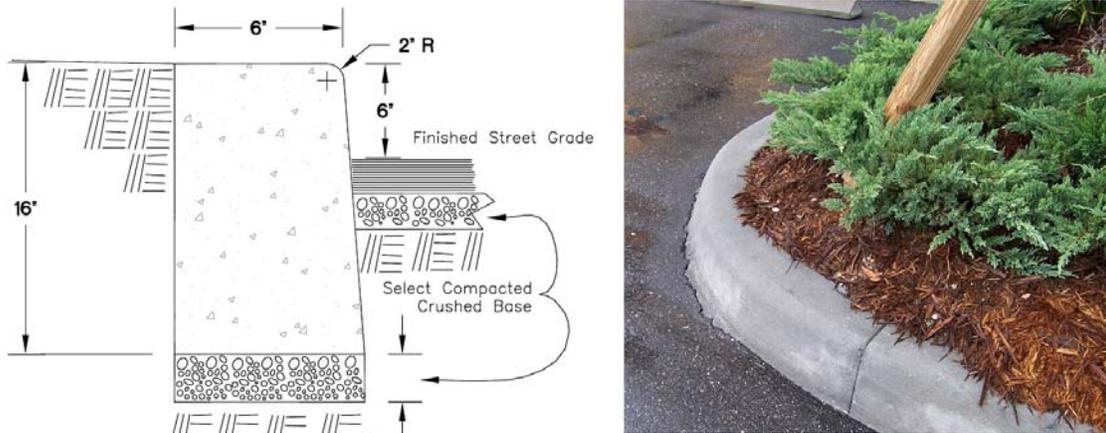


Figure 15: Curb Type B

The first curb we will create is Type A, which surrounds the parking lot. This type of curb is usually represented by three lines, and the drawings for this project make no exception to this. The outer line closest to the grass represents the back side of the curb, the middle line represents the top inside part of the curb, and the inside line represents the point where the curb meets the road.

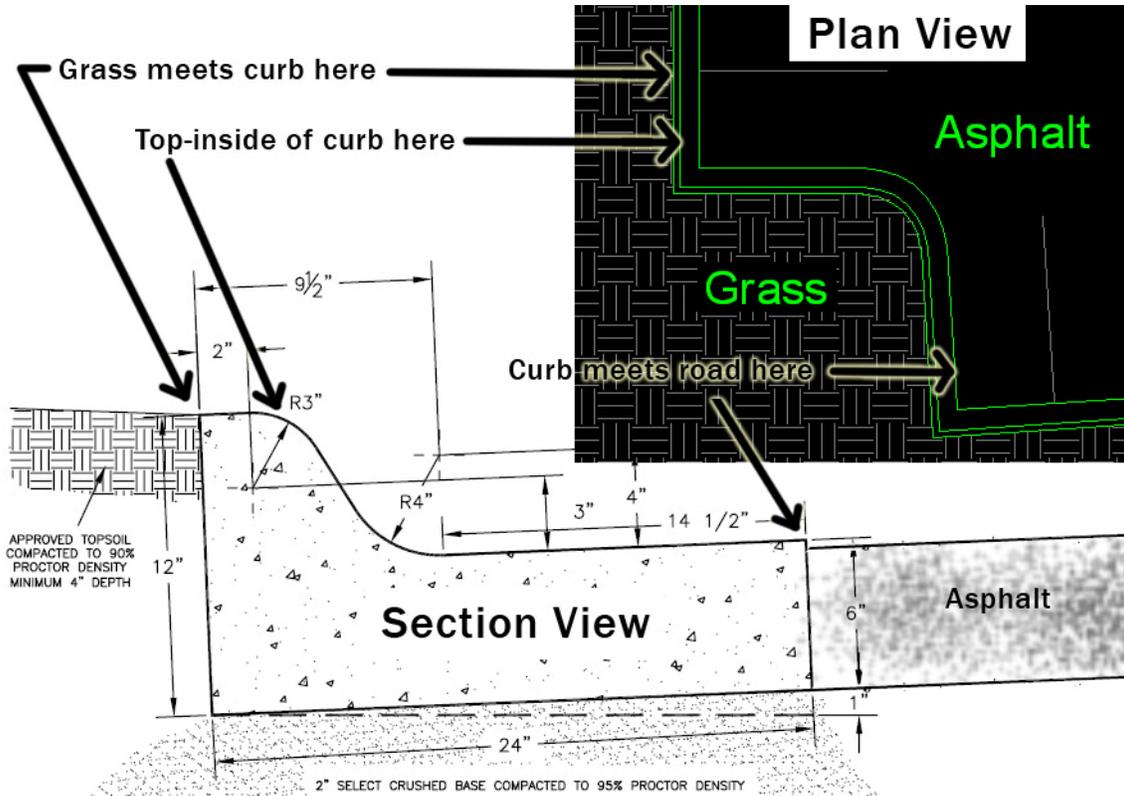


Figure 16: A plan and section view of the curb

The first thing to decide is what type of method you're going to use to model the curbs. For this particular type of curb, the **Loft** command or the **Sweep** modifier works great and with good linework in hand you can create the model in just a few seconds. As a matter of personal preference, I always use the **Loft** command, but either will suffice.

In this tutorial, we will use the **Loft**. The first step is to create a shape that represents a cross-section of the curb and a path with which to loft the shape along. The shape I created for this curb, shown below, is a simplified version of the curb section shown above. This curb is 6 inches tall, 24 inches wide, and contains two curves with a 2-inch radius.

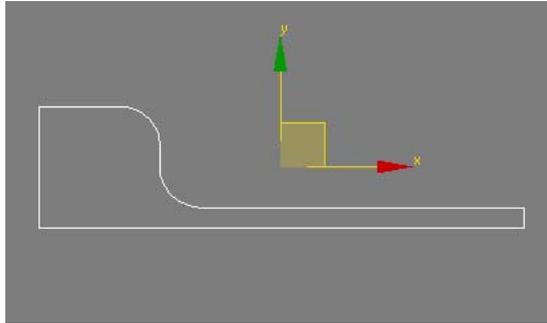


Figure 17: The shape to be lofted with the pivot point centered on the shape

Before you can use any shape for a loft, you have to adjust the shape's pivot point so that the loft is created in the appropriate place in 3D space. As a rule, I always establish my roads at an elevation of zero, i.e., aligned with the home grid, and most other objects tend to be at an elevation at or above this point. In the image above, you can see that the pivot point is located at the center of the object. However, this is not an appropriate place with which to align a loft in 3D space because the center of the loft will be aligned with the home grid, and therefore the top half of the curb will be above the roads, and the bottom half will be below the roads.

So once your shape is created and selected, open the **Hierarchy** panel, select **Affect Pivot Only**, and move your pivot point to the bottom of the shape and just to the left of the point in the curb profile represented by the line that was selected for the path of the loft. I recommend using the **Align** command to at least align the pivot point perfectly with the bottom of the shape. The result should be something similar to the image below. If you wish to use this shape in creating the loft, the file name is **Shape_Curb_A.max**.

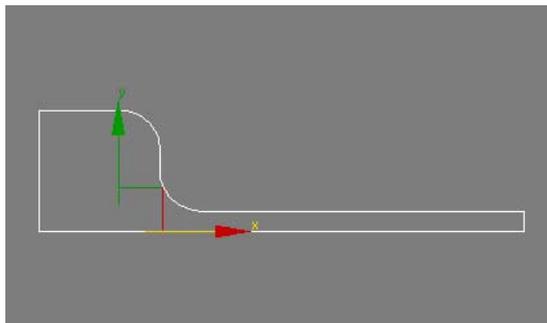


Figure 18: The shape to be lofted with the pivot point aligned properly

As I mentioned before, in the original drawing (**Uno_initial.dwg**) there were three lines that represent the structure of the curbs, shown in the image below, yet in the cleaned file, **Uno_cleaned.dwg**, you'll notice that only one line exists. Again, this is because only one line is needed to loft the shape and anything brought into 3ds Max would be unnecessary and only

serve to clutter your workspace. The line that I chose to use as the path for the loft is the middle line, which represents the top inside portion of the curb (where the shape's pivot point is located.)

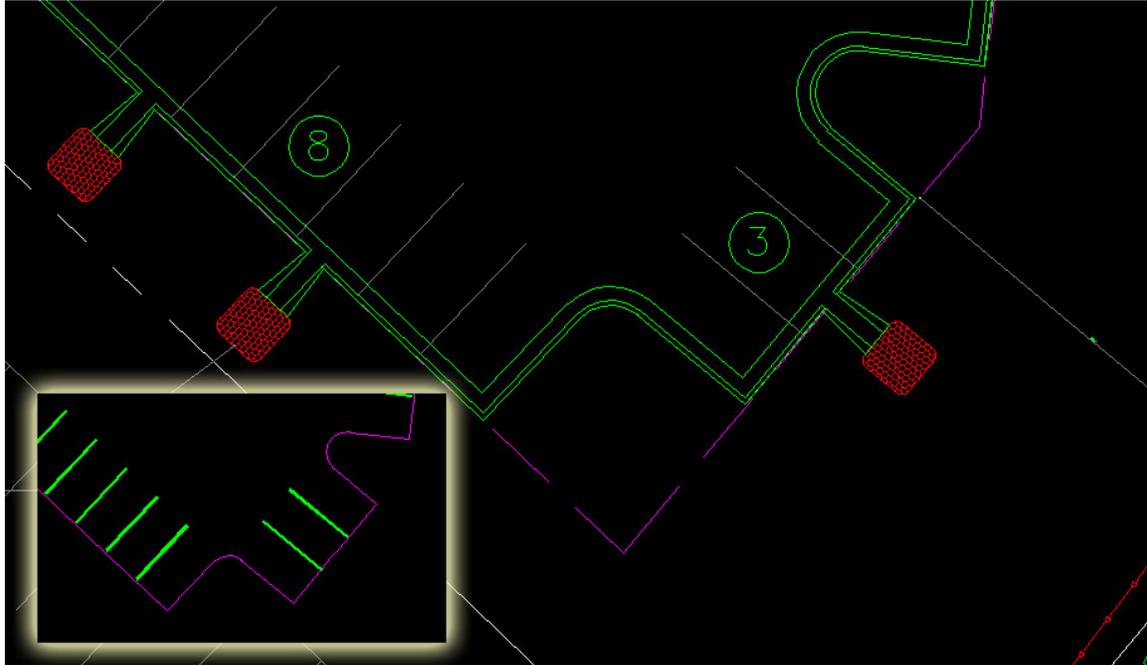
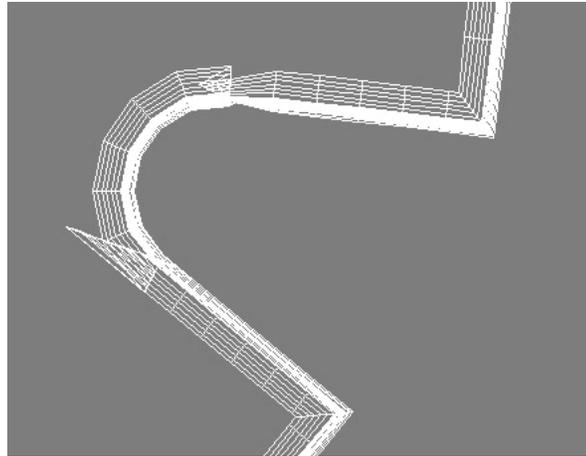


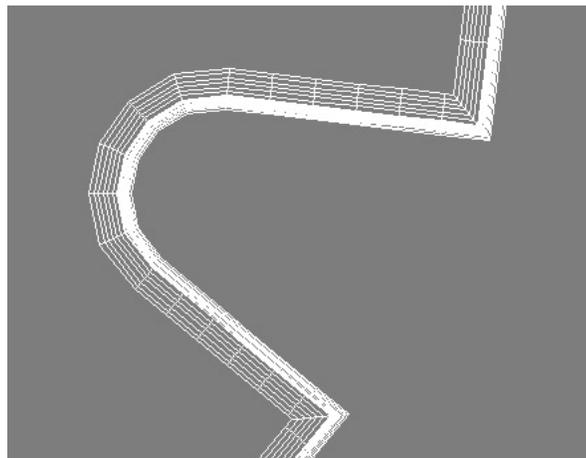
Figure 19: Original linework representing the curbs and the cleaned linework representing the loft path (inset)

Let's continue creating our curb:

1. Continue from the previous exercise or open the file **My_Uno_02.max**.
2. Merge the curb shape you created or the one found in the file **Shape_Curb_A.max**. Notice that the shape is merged into your scene at the scene origin. I placed the shape at the origin for convenience in locating it later.
3. Select the object **0-Site-Roads.01**.
4. In the **Command** panel, select **Create > Geometry > Compound Objects > Loft**.
5. Click the **Get Shape** button and click on the shape you just merged, **Shape_Curb_A.max**. A loft is created.
6. Zoom into any curved area of the loft you just created. You'll probably notice that the loft appears to be highly deformed in just about every area, as shown below. This is because the vertices did not weld properly when they were imported into 3ds Max.



7. Delete the loft you just created.
8. Select the object **0-Site-Roads.01** again.
9. In the **Command** panel, enter the **Vertex** sub-object level of the **Edit Spline** modifier.
10. Click the **Zoom Extents All** icon and select all the vertices in the object.
11. In the **Command** panel, click the **Weld** button. This welds all the endpoints that may have not welded properly during the import into 3ds Max.
12. Repeat steps 4 and 5 to loft the curb object again. The result this time should look like the image below.



Although the deformations have been removed, notice in the image above that the curb appears chiseled rather than smoothly curved. This is because there are both not enough vertices in the original path and not enough default steps used when the loft was created. You could increase the number of step in the loft from the default value of 5 to 10 to achieve much smoother curves, however, this will create a new issue: the faces will overlap each other at corners because 3ds Max cannot fit so many steps into such tight areas. The left image below shows the curb object with a default step value of 5 and the right image shows the effect of increasing that value to 10.

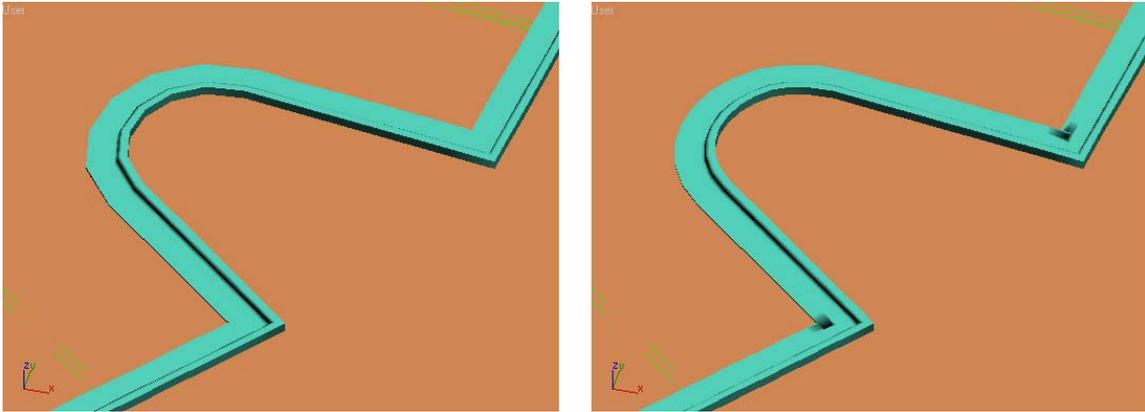
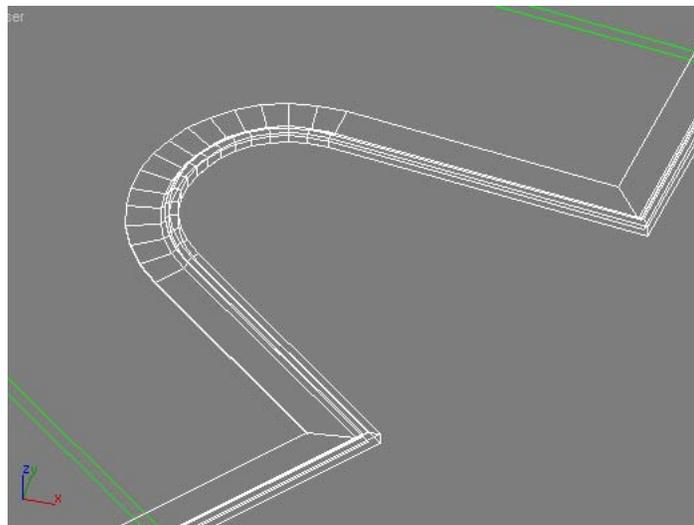


Figure 20: The curb object lofted with 5 steps (left) and 10 steps (right)

So even if increasing the steps is not a problem, and doesn't result in the imperfections shown above, you are still left with a mesh containing far too many faces in areas of the loft that don't need it— this is very inefficient, and will result in a performance hit if you have lots of areas of curb like this in one project. So what is the solution? Let's try one.

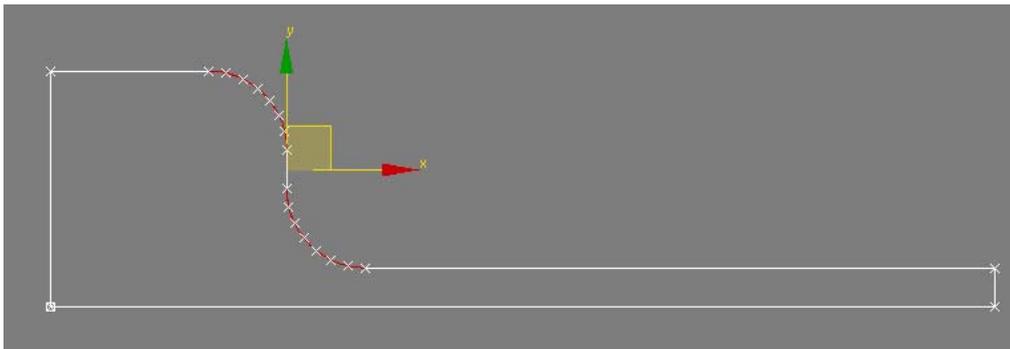
13. With the loft selected, open the **Skin Parameters** rollout in the **Command** panel and reduce the **Shape Steps** and **Path Steps** value to 0. Notice that the loft loses all curves along both its cross section and along the path. This ensures that there will be no unnecessary faces along portions of the loft that should be completely straight. Now let's add faces where they are needed.
14. Select the object **0-Site-Roads.01** and enter **Segment** sub-object mode.
15. In the **Top** view, select the segment of the spline that you would like to make smoother.
16. In the Command panel, pan down to the bottom of the **Geometry** rollout and click the **Divide** button. Notice that the highlighted loft segment is divided into 2 segments of equal length.
17. Click the **Divide** button again. The 2 loft segments are now divided into 4 segments of equal length and a curve is reappearing in the loft. Continue to click the **Divide** button until this portion of the loft achieves a reasonable level of curvature, as shown below.



18. Repeat this same process with each portion of your loft that should be smoothly curved. You can highlight multiple segments of the path and divide those segments at the same time. To better see the segments you want to highlight, delete the loft and recreate it once the path is structured properly.

In addition to the path, the shape needs to contain an adequate number of steps, which it presently does not—we'll get onto that problem below:

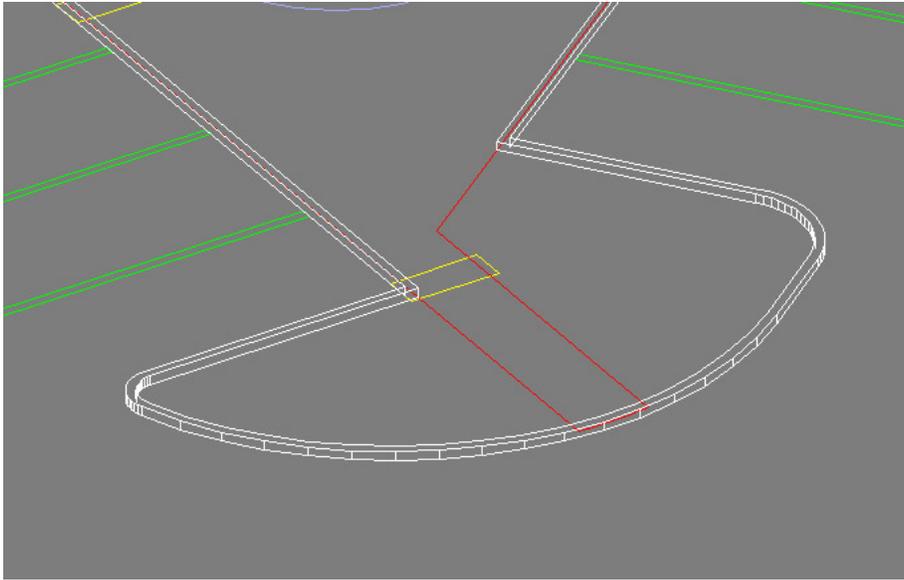
19. Select the shape **Shape-Curb_A** and enter **Segment** sub-object mode.
20. Select the two curved portions of the shape and divide these segments with a segment value of 6, as shown below. This will provide smooth enough curves without too many unnecessary vertices.



21. Recreate the loft and you should have an object that accurately depicts the curbs around the perimeter of the parking lot.
22. Change the name of your loft to **Site-Curb1**.
23. Save your file as **My_Uno_03.max**.

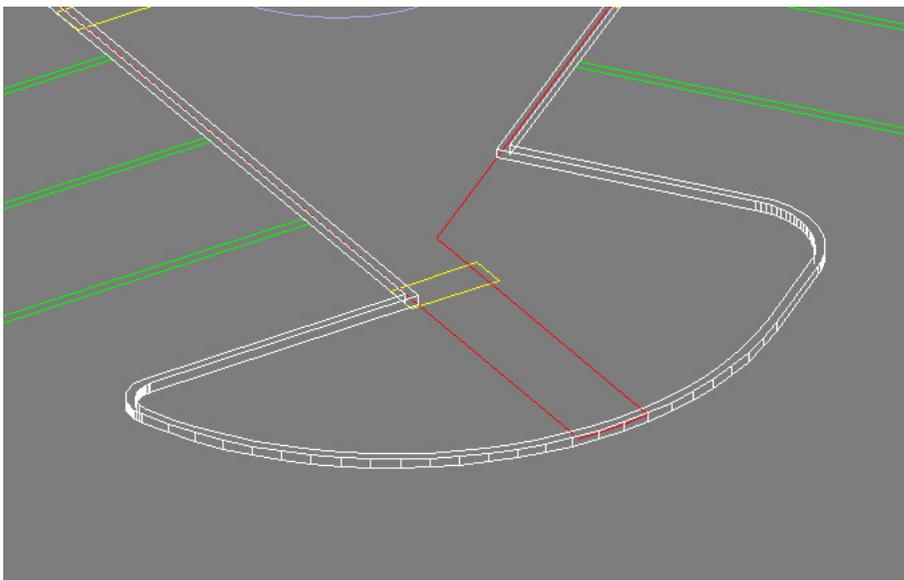
Now we need to create curb type B, which surrounds the paved walking area and the restaurant. Because of the structure of this particular curb type and how the linework was developed, this curb should only take a few quick steps to create.

1. Continue from the previous exercise or open the file **My_Uno_03.max**.
2. Select and isolate the object **0-Site-Curbs.01**.
3. Rename it to **Site-Curb2**.
4. Enter **Vertex** sub-object mode and weld all of the spline's vertices.
5. Enter **Spline** sub-object mode, and within the **Geometry** rollout, enter a value of **-6** in the field next to **Outline**.
6. Click the **Outline** button. This creates a duplicate spline 6 inches inside the initial spline.
7. Add the **Extrude** modifier and enter an **Amount** of **6**. This will extrude the spline into a 6-inch high curb, as shown below.



Let's assume for a moment that the curved portions of this object don't contain enough steps to make the curves smooth enough. This can be quickly remedied.

8. Go to the **Edit Spline** modifier within the modifier stack.
9. Open the **Interpolation** rollout and change the **Steps** value from 16 to 24.
10. Return to the **Extrude** modifier in the modifier stack. You should now see a much greater number of steps that make-up each curved portion of the curb, as shown below. Higher step values will give even smoother curves.
11. Save your file as **My_Uno_04.max**.



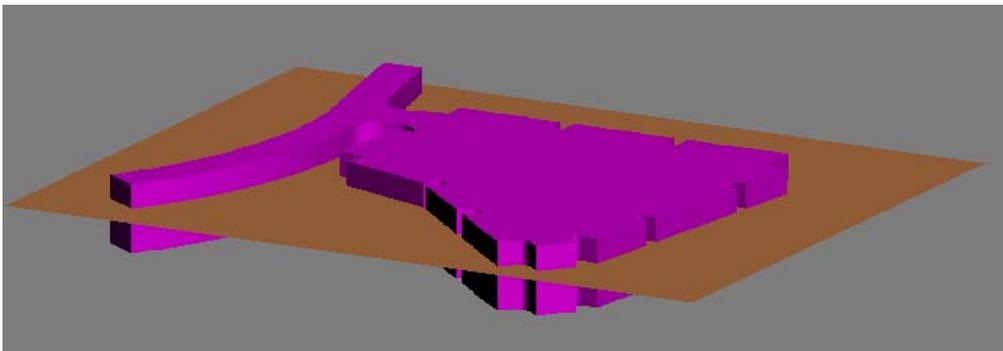
If you want a curve to appear on the top-outer edge of the curb on the side that meets the road, which is usually found in most curbs in the real world, you can create a **loft** or apply the **Sweep** modifier using a shape with a curved edge. For the purposes of this tutorial, we will leave the curb as is and call it an accurate depiction of the curb needed for this project.

Now that the curbs are modeled, we need to move ahead and work on the road for this scene.

Roads

Because of the way we created the curbs for this scene, we can take advantage of a very quick and effective way of accurately creating the roads. Let start by using the path from which we created the loft.

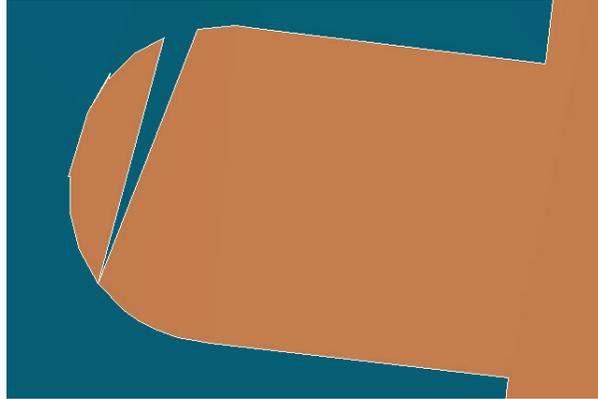
1. Continue from the previous exercise or open the file **My_Uno_04.max**.
2. Select the object **0-Site-Roads.01**.
3. Make a clone of this object and name it **Temp**. This object will have a short lifespan, so I chose to give a suitable name. Also, ensure that you use the **Copy** option rather than the **Instance** or **Reference** option.
4. Rename the object **0-Site-Roads.01** to **Path-Site-Curb1**. This will help in the future if you decide you want to make a change to the path linked to the loft you created.
5. Select and isolate the object named **Temp** and **Site-Terrain**.
6. Extrude the object named **Temp**, **500 inches**.
7. Move this object **250 inches downward** through the object **Site-Terrain**, as shown below.



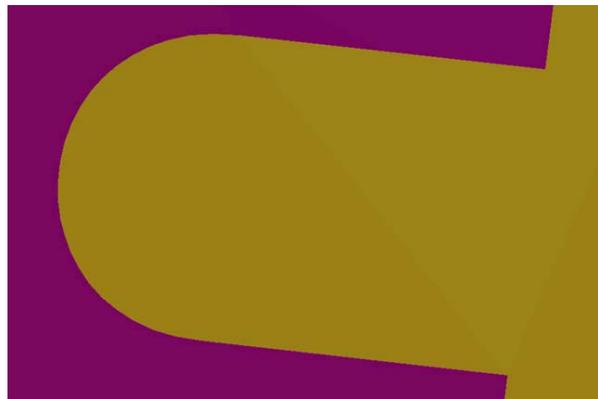
8. **This next step is very Important:** In the **Edit** menu select **Hold**. This stores a copy of your scene for later retrieval. We will need to retrieve it shortly.
9. Select the object **Site-Terrain**.
10. In the **Command** panel, select **Create > Geometry > Compound Objects > Boolean > Cut > Split**.
11. Click the **Pick Operand B** button and click on the object named **Temp** in any viewport. The object named **Temp** no longer exists and in any wireframe viewport you should see an outline of the roads cut into the **Site-Terrain** object.
12. Add the **Edit Mesh** modifier to the **Site-Terrain** object.
13. Enter **Face** sub-object level. You should see the faces selected for the area cut out of the terrain.
14. Click the **Detach** button. If the **Detach** button does not bring up the **Detach** dialog box, don't worry—this is a fairly common bug. Simply click the **Delete** button, click the **Undo** icon, and try clicking the **Detach** button again and it should work fine.
15. Name the detached object **Site-Roads** and click **OK** to end the command.
16. Close the sub-object level selection and select the object **Site-Roads**.
17. Give the **Site-Roads** object a vastly different color from the **Site-Terrain** object so that the two adjacent objects can be easily distinguished.

Look closely around different areas of the terrain object. You will probably notice that 3ds Max did not do a very good job of cutting the extruded object out of the terrain object. There is probably a number of missing and deformed faces along the edge of where the cut took place, as shown in the image below. There is a very good reason for this—boolean operations do not always work well when some of the object faces are long and skinny, as is the case in this scene. The reason the faces are long and skinny is because of the distance and angle between vertices

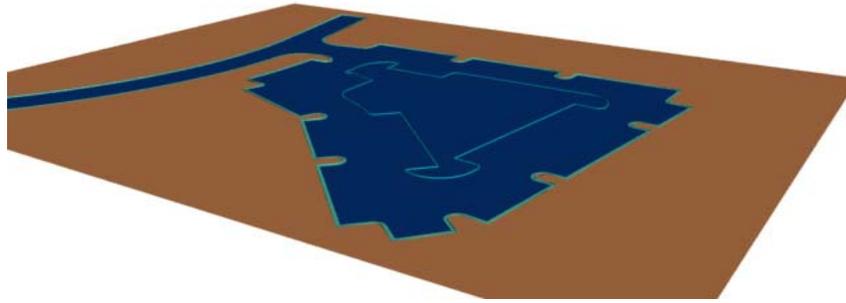
of the terrain object and vertices of the cutting object. If an operation does not work right the first time, try adding more vertices to the mesh through object parameters, or even with certain modifiers such as the **Subdivide** or **Tessellate**. In this particular case, I prefer to quickly change the rectangle we started the terrain with and place a simple plane in its place.



18. In the **Edit** menu, select **Fetch** and **Yes** to complete the **Fetch** command.
19. Press the keyboard shortcut **S** to turn on snap mode.
20. Ensure that **Endpoint** snaps are on and create a plane with the default **4** length and width segments. To do this, use the corners of the original terrain object to place the plane in exactly the same location as the original terrain. Snaps are setup through **Customize > Grid and Snap Settings**.
21. Delete the original **Site-Terrain** object.
22. Rename the plane you just created as **Site-Terrain**.
23. Press the keyboard shortcut **S** to turn off snap mode.
24. Repeat steps **9 through 17** to recreate the terrain properly. This time you should have clean edges all around your road object, as shown in the image below. Depending on the complexity of future scenes you apply this operation to, you may need to subdivide your mesh further or start with a plane object containing more segments.



25. Exit isolation mode (or unhide all objects) to make all scene objects visible again.
26. Now might be a good time to change the background of your scene to some lighter color. Black backgrounds can negatively impact the mood of your renderings and it's best to use a lighter color even for test renders.
27. Render the scene. It should look like the image below, though the colors may be different.
28. Save your file as, **My_Uno_05.max**.



Terrain (Part II) and Pavers

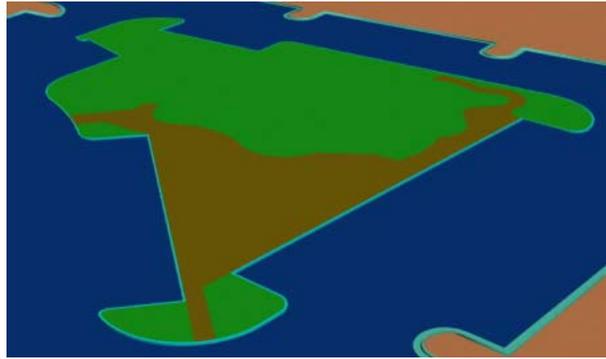
Now that we have our roads and curbs created, we have to return to finishing the terrain in our scene - the land within the perimeter of the 2nd curb object we just created. In the process of creating the inner terrain object, we will be simultaneously creating the pavers that surround the restaurant. Fortunately, the same line we used for the curb we just created can be used for the inner terrain.

1. Continue from the previous exercise or open the file **My_Uno_05.max**.
2. Select the object **Site-Curb2**.
3. Make a clone of this object and name it **Site-Terrain2**. Ensure that you use the **Copy** option rather than the **Instance** or **Reference** option.
4. Remove the **Extrude** modifier to return the object to a spline.
5. Enter **Spline** sub-object level and delete the outer spline.
6. Add the **Edit Mesh** modifier.
7. Raise the object 5.8 inches upwards.
8. Select and isolate the objects **0-Site-Pavers.01** and **Site-Terrain2**.
9. Select the object **0-Site-Pavers.01** and rename it **Temp**.
10. Enter **Vertex** sub-object mode and weld all vertices. Welding vertices on imported splines is a good habit to get into. It only takes a few seconds and it can save grief down the line.

Now we're going to repeat the same process we used during the creation of the pavers.

11. Extrude the object **Temp**, **500 inches**.
12. Move the object 250 inches downwards through the object **Site-Terrain2**.
13. Select the object **Site-Terrain2**.
14. In the **Command** panel, select **Create > Geometry > Compound Objects > Boolean > Cut > Split**.
15. Click the **Pick Operand B** button and click on the object named **Temp** in any viewport. The object named **Temp** no longer exists and in any wireframe viewport you should see an outline of the pavers cut into the **Site-Terrain2** object.
16. Add the **Edit Mesh** modifier to the **Site-Terrain2** object.
17. Enter **Face** sub-object level. You should see the faces selected for the area cut out of the terrain.
18. Click the **Detach** button. As before, if the **Detach** button does not bring about the **Detach** dialog box, don't worry. Simply click the **Delete** button, click the **Undo** icon, and try clicking the **Detach** button again and it should work fine.
19. Name the detached object **Site-Pavers** and click **OK** to end the command.
20. Close the sub-object level selection and select the object **Site-Pavers**.
21. Give the **Site-Pavers** object a vastly different color from the **Site-Terrain2** object so that the two adjacent objects can be easily distinguished.
22. Exit isolation mode to make all scene objects visible again.

23. Render the scene. It should look like the image below, though the colors may vary.



24. Save your file as, **My_Uno_06.max**.

Summary - Part I

The goal of this tutorial is not only to cover some of the fundamental concepts behind the creation of 3D site plans, it's to show you just how easy they can be to create. Hopefully, you have found this to be the case. The steps presented in this tutorial may seem confusing at first if you have never created a site plan in this fashion, but with practice you might be amazed at just how quickly they can be performed once you are familiar with the process. An experienced user who uses this same procedure to create 3D sites should be able to go from good linework to finished models in as little as 20 minutes for this particular scene.

In Part II of this tutorial, we will finish modeling the remaining critical site elements and look at ways to bring the scene to life with some effective materials and lighting. We will also merge the completed restaurant into the completed site and briefly cover how to achieve an effective final product – a short animation. Finally, at the end of Part II, we will cover how to model a site that contains 3D terrain rather than flat terrain, as used thus far, and discuss numerous other considerations that effect rendering times, file sizes, and integration