

Modifying Default Airport Terrain with *Sbuilder*

Introduction

This tutorial describes the installation and use of two tools, Sbuilder 2.05 and LWMViewer 1.3, to review and modify airport terrain. This tutorial assumes that an AFCAD file has been developed which has modified the airport geometry such that the default terrain no longer works properly. There can be a couple of reasons for this:

- MSFS put the airport in the wrong position

- The airport has been modified since FS9 was released

The airport we will work on is Cleveland-Hopkins (KCLE) in Ohio, USA. This airport has a new runway added, 6L/24R, which expands the "footprint" of the airport. The revised AFCAD file is by Jim Vile and named `kcle_ils_jv.zip`. This is part of Jim's "Active ILS" series of airport fixes. A feature of this series is that it includes the necessary approach data in a separate file which allows AI and user aircraft to use the new runways using ATC in IMC weather. Note that the zip file actually has two sets of files: One is intended for use with "Shez"'s KCLE scenery for this airport, and the other set for the default scenery. We will use the default set, which consists of two files: the first is a standard AFCAD file which is installed into your preferred AFCAD scenery folder, such as `Addon scenery\scenery\`. The second file has the ILS approach data and must be installed into the default scenery folder `Generic\scenery` in order for FS9 to update the necessary information for the GPS and AI functions. Note that there are some modifications to the files in other library files if you are using the Shez version of KCLE.

Step 1: Obtain and Install Tools

Latest full version of sbuilder is here:

<http://www.ptsim.com/downloads/sb205.zip>

and revised exe file:

<http://www.ptsim.com/downloads/sb205rev06.zip>

You also need sb205objects from AVSIM here:

<http://library.avsim.net/download.php?DLID=69398>

Support forum is here:

http://www.ptsim.com/forum/forum.asp?FORUM_ID=18

To make sbuilder work, I think you might need some MS visual basic files. Not sure which ones. (Once you get the right ones installed they work with many different programs.)

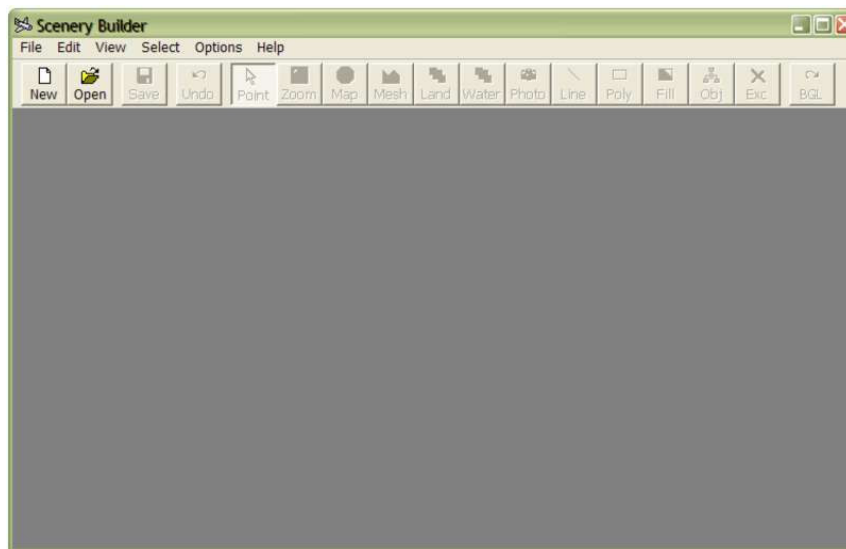
The other thing you need for sbuilder is the assembler SCASM 2.96, which is here:

<http://www.scasm.de/>

One other utility to make life easier is LWMViewer.
<http://www.jimkeir.co.uk/FlightSim/LWMViewer.html>

Install sbuilder in this method:

1. Unzip to install sbuilder 2.05 to a new folder sbuilder, using the paths in the zip file for subfolders. This will result in the following:
 Sbuilder -- with the main program and help
 Tools -- various helpers
 Bmps -- some bitmap files
2. Open scasm296.zip, and unzip the scasm.exe to Sbuilder\tools\
3. Start sbuilder. It should take some time to pull in some textures and then open to the main screen. Go to Help -> register on the menu bar. It used to be necessary to buy the payware version to unlock all features. Now I think it is freeware/donationware, but on the register dialog is a code you can use to register your copy. It will open a web page and you type this code in, then it will send you an email with the "password" which you can copy and paste into the text box.
4. Unzip sb205objects.zip using the path in the zip file. This should place some more files into your sbuilder folder and create an sbuilder\tools\ASD folder.
5. Unzip the sbuilder.exe file in sb205rev06.zip to the sbuilder folder, overwriting the one in there.
6. Start sbuilder and verify the main window:



Install LWMViewer by creating a new LWMViewer folder and unzipping the contents of the zip file to this folder.

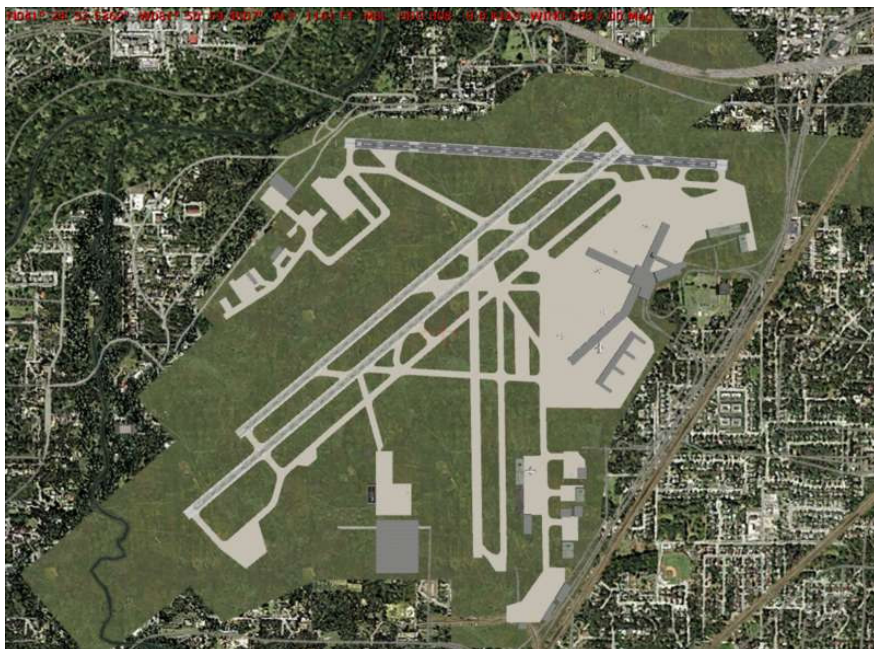
Step 2: Determine the Scope of the problem and obtain FS9 Screenshot

Default scenery in FS9 is organized in a folder tree under \scenery\. There are a couple of world-wide folders (Base, World, and Generic) and the rest are in regional folders. The world is divided up into squares known as "LOD5". These LOD5 squares are numbered and arranged into files for each type of scenery. For our project, the types of interest are FL -- these are the airport flatten areas, and AB -- these are the airport backgrounds. Both of these are sets of polygons. The flatten polygons have an altitude attribute (to set the elevation of the polygon). The airport backgrounds have a texture ID attribute (to determine the correct texture bitmap to use), and a layer attribute (which determines the order in which overlapping polygons are drawn by the terrain engine). The texture ID is an integer that is used to find a section in the default file, terrain.cfg in FS9. This file has a number of sections which provide instructions for various types of terrain features, but there is a set which are used for the backgrounds. The main feature of these sections is that there is a line which tells the terrain engine to exclude autogen within the texture.

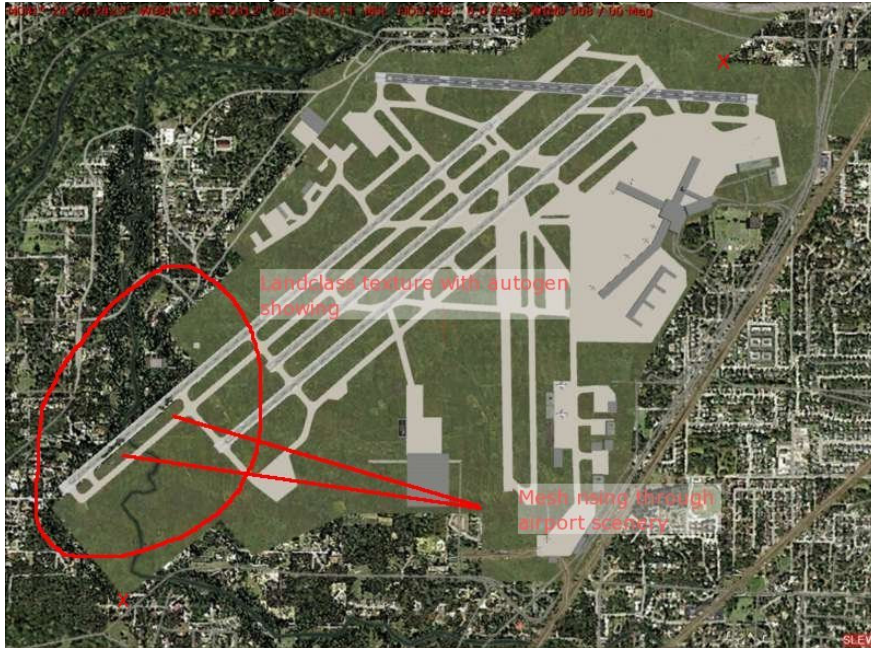
For this project, we will use screenshots from FS9 to determine where our modified polygons need to be placed. An alternative is to use some other source, such as a map or aerial photo. To use these sources, they need to be mapped using the geographic projection and either North American 1983 (NAD83) or WGS-84 datum. Usually the source will have some information (metadata) which tells you the projection and datum. In the US, the thing to look out for is that it is common to find maps projected to either Universal Transverse Mercator (UTM) or State Plane Coordinate System (SPCS). These projections are used to permit measurement of distance in meters or feet. (Geo projection has all points in lat/long which can't easily be converted to linear measurement.) You might also see maps with North American Datum of 1927 (NAD27). Positions in this datum will be several hundred meters off from what you need for FS9. Of course, outside of the US there are many other local projections and datums used. Due to wide use of GPS units, though, Geo/WGS84 is becoming more common, since this is the native projection and datum used by the GPS system (Your GPS reported position can be directly plotted on a map in Geo/WGS84.)

Start FS9 and go to KCLE. Select top-down view and slew mode. Slew/zoom as necessary to get your project area in view. Press the space bar to orient the view "north up". (Always remember to do this before taking a screen shot or you can get rotated images which are no good.)

Here is a shot of the default airport:



Exit FS9 and install your new AFCAD file. Take a new screenshot:

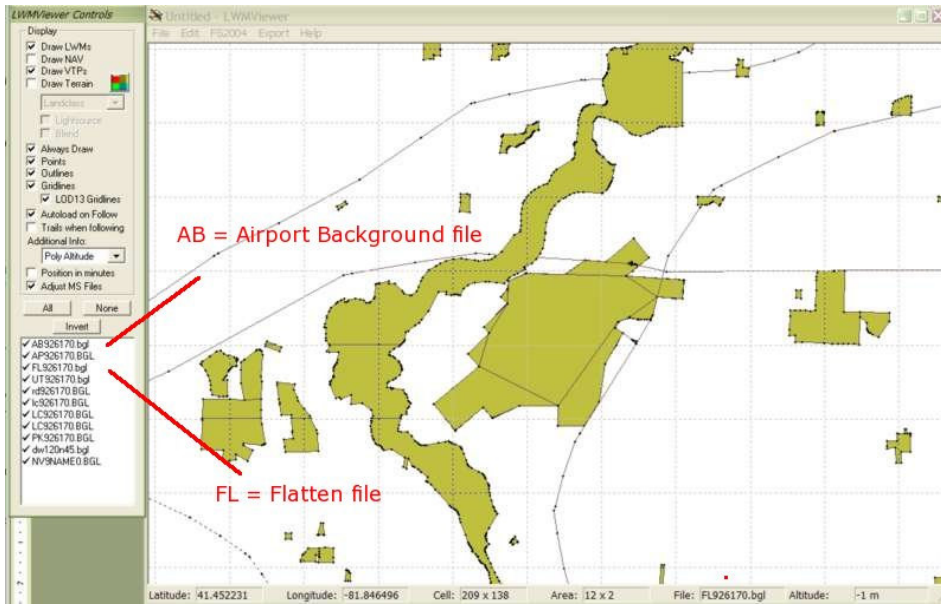


To properly position and scale the bitmap, you need the lat/long of 2 points on the image, preferably close to diagonal corners. You need points that you can easily identify on the bitmap. One way is to use the end of two runways diagonally opposite. Slew to the corner of the runway, and write down the lat/long. Another possibility is to use FSUIPC in FS9, and run sbuilder along with FS9. There is a view option in sbuilder to display a crosshair at the current aircraft position, like in AFCAD or other utilities. For this project, we will use two corners of the default background, since we will be importing this polygon into our sbuilder project. The X's mark the spots we will use.

Save the shot as a bitmap file.

Now we need to find the default files for the flattens and backgrounds.

You can read off the lat/long of KCLE from your top-down slew view. It's about 41.40 / -81.85 (in most cases it is easier to work in decimal degrees, rather than degrees and minutes that FS9 displays). Now open LWMViewer and go to File -> Open by Coords and enter the position into the text boxes. Click load, then close existing files (yes) and you should see a display of the default files for this area. In the LWMViewer Controls, make sure draw LWM and draw VTP are checked. You should see something like this (you need to pan to get to the 41.4/-81.75 position):



Note that there are more areas highlighted than we are interested in. The only files we need are AB926170.bgl and FL926170.bgl (26-17 identifies the LOD5 area). You can click on the other files to uncheck them and see just these two files. These files we will use in sbuilder as the basis for our new scenery.

Step 3: A review of FS9 scenery concepts

Before starting our mod, a brief review of FS scenery. An airport is built up from several components. The main types are terrain and scenery objects. Both consist of mesh (3d models) and textures which are wrapped over the mesh.

The terrain mesh/model is defined in files commonly referred to as "mesh". This mesh can be modified by flatten polygons. Flattens force the mesh to a single elevation within their areas. (There is a sloped flatten, but that isn't relevant to airports.) Note that a "feature" of FS9 is that flattens can't be excluded, you can only place an additional flatten in the same area (the flattens are applied in priority order as defined by scenery.cfg).

Next, textures are wrapped over the mesh. FS provides for about 100 layers of different textures, which are placed, one over the other. Only a few of the possible number are used in the default files. The first texture is the landclass texture. In principle, you could simply change the landclass texture to create an appropriate airport background. The problem is that landclass textures cover a square 1.2 km area, which may not coincide with the airport. Thus texture polygons can be used to locally modify the landclass texture. These textures can be defined in two ways: as a texture ID, or as a specific bitmap. The texture ID provides a convenient means of achieving a specific texture effect. The IDs from 0 -255 call landclass textures. The IDs above 255 result in a lookup in the file terrain.cfg. This file can provide a number of features for a texture ID, besides defining the texture. It can control the autogen features, and also affect the mesh flattening. FS has defined a set of texture IDs for airport backgrounds. The basic

texture is the same as landclass textures, but all autogen has been excluded. Thus, using an airport texture ID works better than using a landclass texture ID.

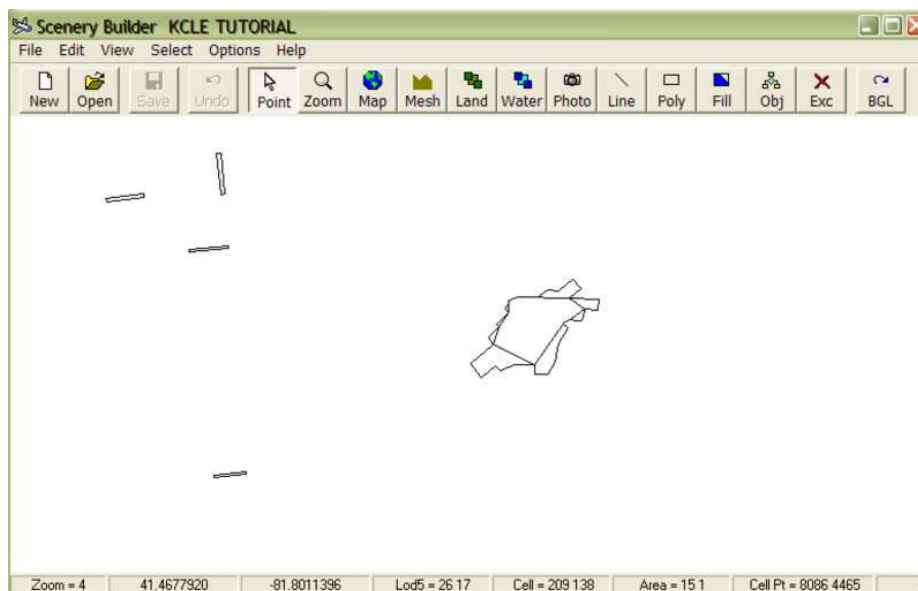
Texture polygons can be excluded. Exclusion works using the scenery.cfg priority system. For terrain, exclusion specifies the layers to be excluded. This way, it is possible to exclude some texture polygons while allowing others to remain.

Once the terrain component is completed, scenery objects can be placed. Airport scenery consists of "flat" objects such as runways and taxiways, and 3d objects such as terminals. An important part of the scenery engine is determining which objects are in view, and which hidden. Improper setting of flattens and airport ground polys can cause texture "flickers", as the shaders have difficulty determining what is in view.

Step 4: Correct the Airport Background

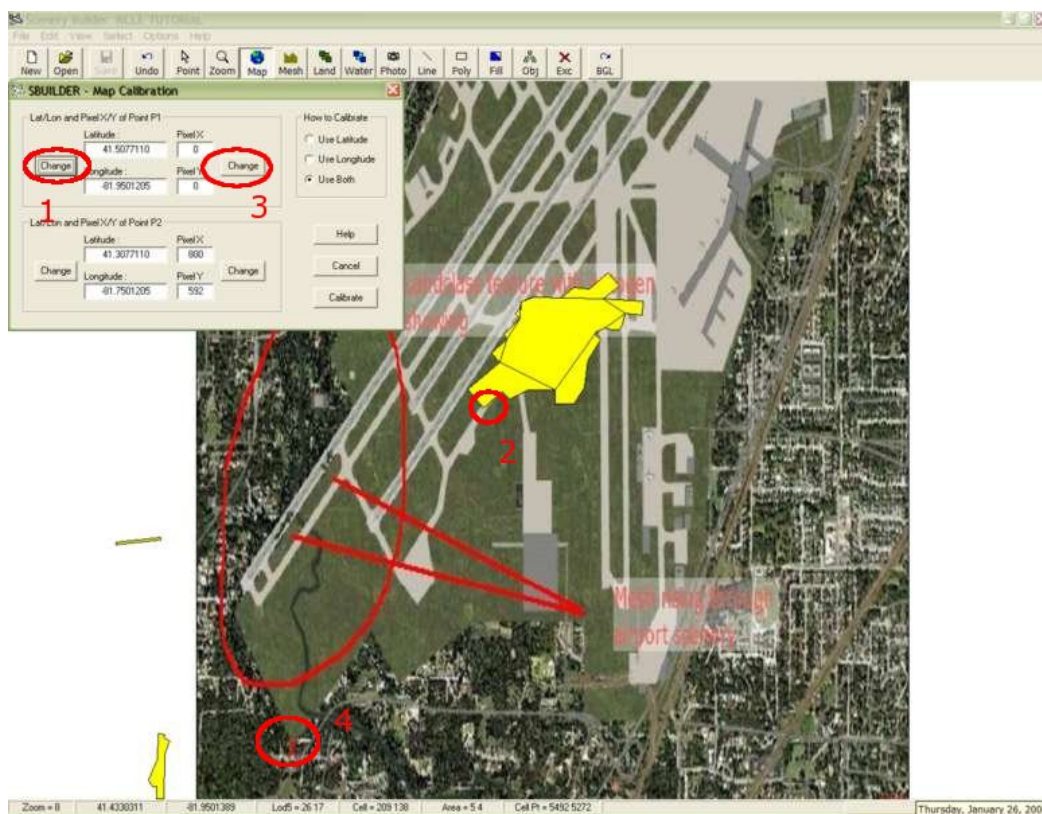
Now that the basics are out of the way, we can begin. In sbuilder create a new project. The only thing on the new project that you should consider at this point is the project name; this will be used in the filenames of the files we create. Note that if you right click on any empty part of the window, you can call up the project properties. (If you attempt to select an object's properties and "miss it", you will get this too.)

First, we will import the airport background file. Sbuilder calls this "append". Menu File -> append -> VTP bgl brings a file browse window. Navigate to scenery\name\scenery\ to highlight AB926170.BGL then click "open". You will see polygons in the display window. (I prefer to set menu Options -> preferences Display Lat/Long in decimal degrees, and on the color tab uncheck the "fill polygons with color".) You can pan the display to KCLE (41.4/-81.85) and see that the polygons for KCLE have been imported



We will next inset our screen shot as a background to guide our drawing. Click on the

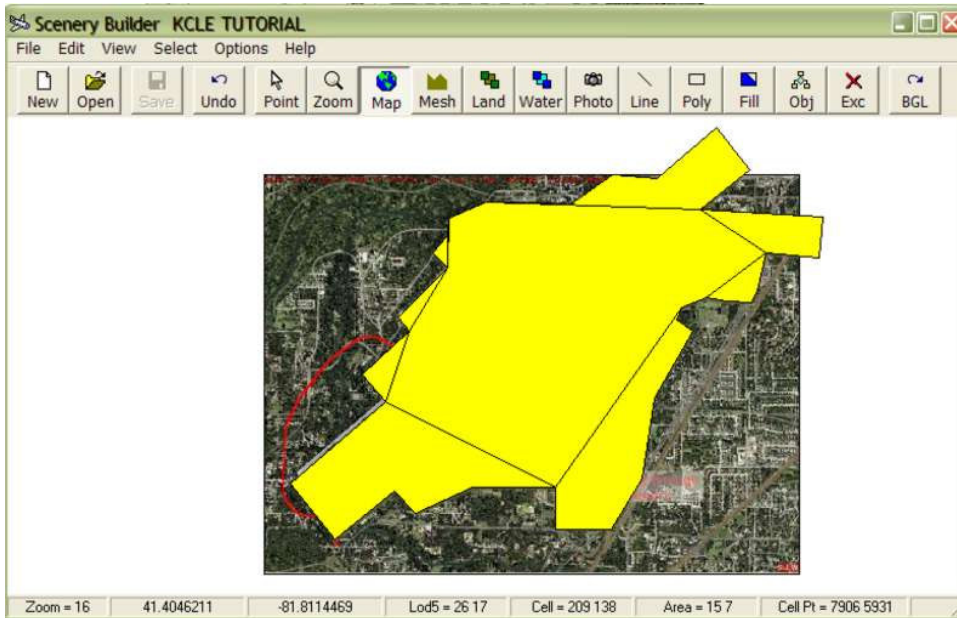
Map tool. Then click the draw window in the middle of the KCLE polys (you don't need to be exact). This will bring up a file browse dialog. Find your screen shot (remember, it must be windows bitmap bmp). Find your file and click open. The bitmap will be displayed in the approximate location, and there is an alert saying "You may need to calibrate this bitmap". Click "OK". This will place your bitmap in the drawing window, and display the calibration dialog. Note that in sbuilder, there can be a number of different bitmaps, for different purposes. Sbuilder calls all of these "maps". Since this is the only bitmap we will be working with, we can leave the name as MAP01. This calibration page is oriented towards situations where the coordinates of the bitmaps corners are known. We will do our calibration visually, so click on the "calibrate photo maps" button. We will do a two-point calibration using the background poly and the visual appearance of it in the bitmap. First we will do the SW corner point of the background.



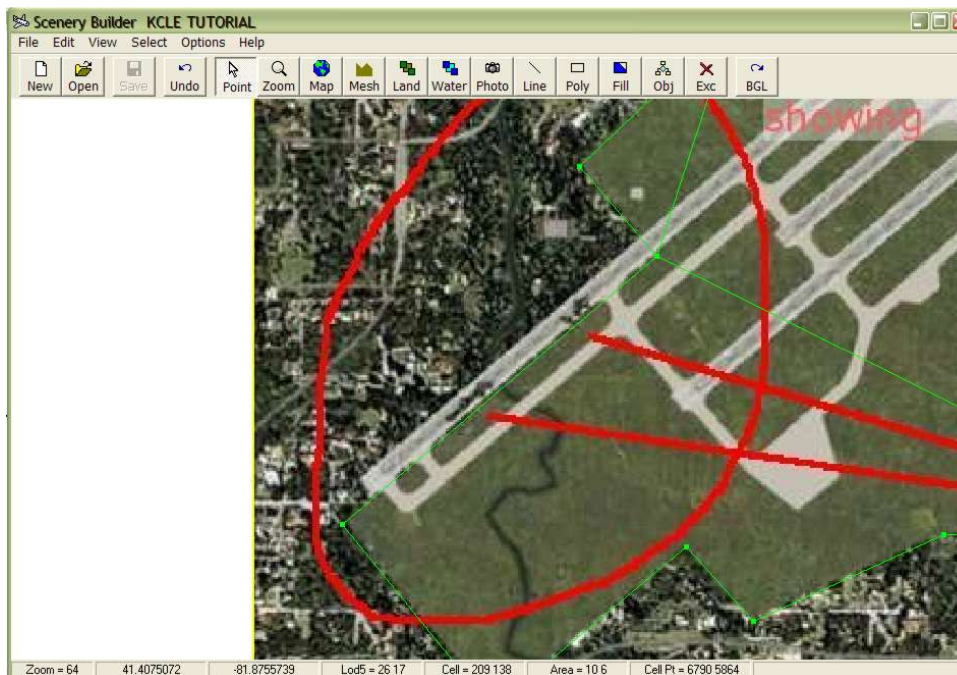
In this image, I turned on "polygon color fill" for clarity. Note the top two "change" buttons. Button 1 will allow us to set the position in the draw window (2), and button 3 will associate a point on the bitmap (4). Click change button "1". This will create a crosshairs pointer. Move to the corner of the background poly (2), and click. This will set the lat/long. Now click change button "3". Once again, use the crosshairs, this time clicking on the corresponding point (4) on the bitmap.

Now repeat the process for the lower two change buttons, this time using the NE background corner. After you are done, click "calibrate". The bitmap should be resized and positioned to correspond with the background polygons. Don't worry if it isn't exact.

If you need to, you can right click on the edge (not inside!) of the bitmap, and a popup menu has a calibrate option and you can try again. The final result should look like this:

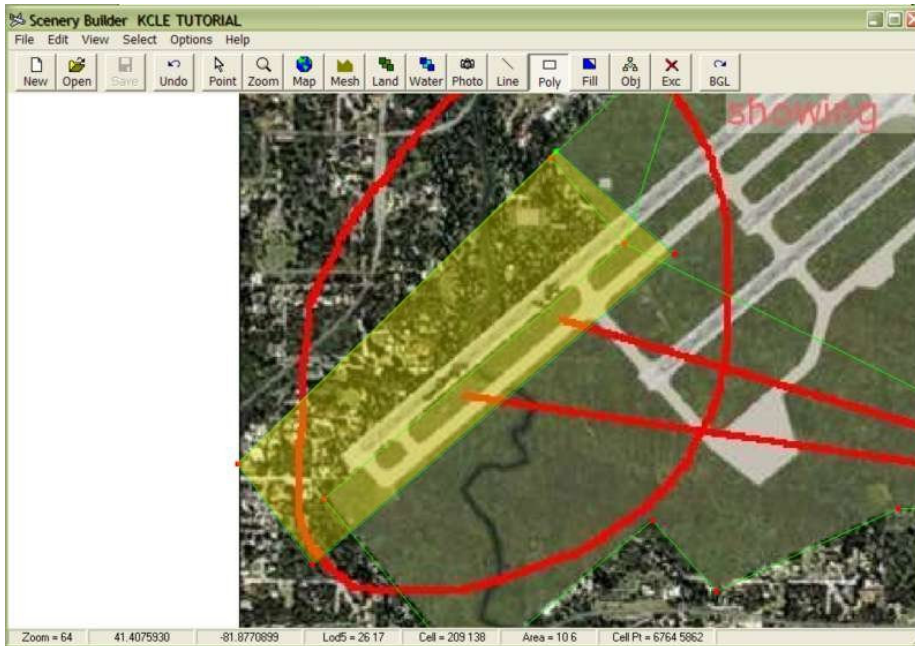


Turn off "color fill" to better check your calibration!



At this point it is a good idea to save your work with menu File -> save as. Now you can create terrain. Click on the "poly" toolbar icon to enter poly mode. First, we will check the attributes of the existing polygons. To do this, right click on the edge (not a point!) of one of the polygons and in the popup menu, select properties. A dialog will open at the

VTP tab. The two things to note are 1, this polygon is set to layer 7, and 2, the "assembled" button is checked, and the text box shows that texture 1190, "Cold Grassland" is used. We will now add a polygon covering the area where the runway has been lengthened. Click on the map on a position that overlaps the default grass area and then add points to the polygon by clicking around the area we want to enclose. The polygon is automatically closed, so you don't have to return to the start point. Hit "escape" key to get out of edit mode. Now you have your polygon. If you click on one of the points, you can edit it or drag it to move it. If you click on the edge of the poly, you can drag the whole polygon.



(Note that this new poly was highlighted for clarity.)

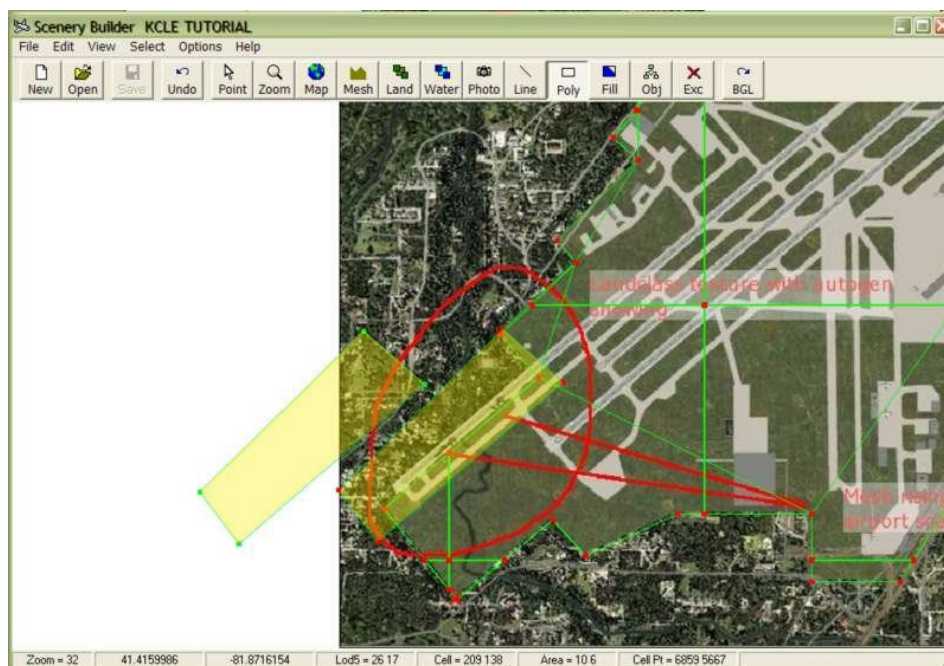
Now right click the edge of the polygon just created, to get the popup menu with a properties option. In the properties there are some tabs. On the general tab, click on VTP2 in polygon type, this will open the VTP tab. In the VTP layer enter "9". This will be higher than any default layer so it will cover any default polys. Then click on "assembled" and click on the text box next to that. That will pop up a window where you can select a texture. It gives you an idea of what the texture looks like. We want to match the default background textures, so we will select 1190. Our background poly is now complete, so click "OK" in the properties dialog. Save your project.

Step 5: Correct the Airport Flatten

We will create a new flatten polygon in a similar manner. This time we must append the flatten file, so go to menu File -> append -> LWM BGL and in the file browse dialog, find \scenery\name\scenery\ FL926170.BGL. After loading the file, pan and zoom again back to KCLE position. It will be difficult to see the flatten polygons, because they cover the same area as the background VTP polygons. Often by zooming in you can see a

slight offset between the two polygons. In builder, each polygon is assigned a number (which also controls its draw order). The flatten polygons will have higher numbers (since the flatten file was appended after the background file). As you move your mouse over a polygon edge, the polygon number is displayed. The one you want will have the higher number. You can confirm your selection by right clicking while the cursor is over an edge and selecting "properties". The flatten polygons will open with the LWM tab open. The LWM type will be "flatten" and the altitude is shown in the "Set altitude" box as 241.0935 m. We will use this altitude for our new flatten so record it. We could create a new polygon in the same way as we did for the VTP polygon, but this time we will do it as a copy -> paste operation.

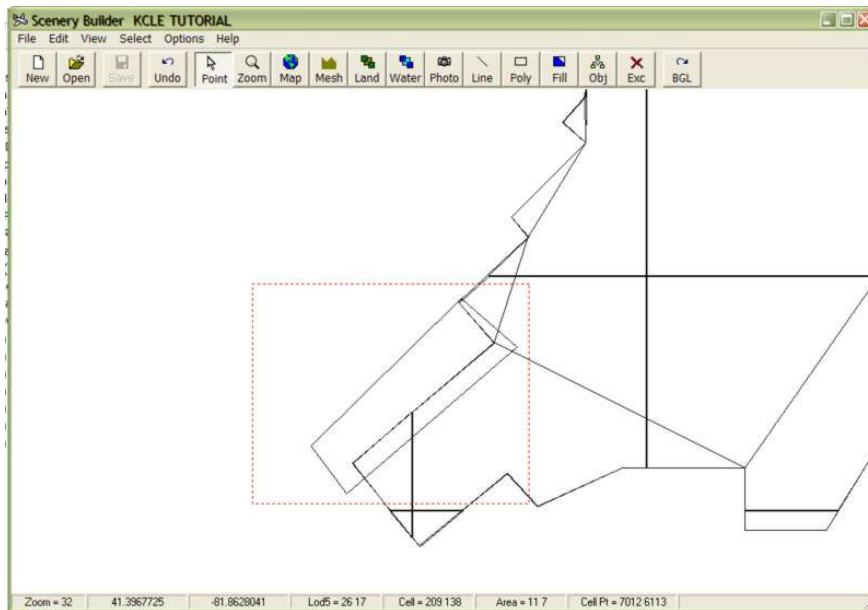
Click on the "poly" tool to go into poly mode, and click on the edge of the new polygon we created earlier. This will select the poly. Then go to the menu Edit -> copy. This copies the poly data. Now go to menu Edit -> paste. Click on the draw window and a copy of the polygon will be inserted. Move the cursor over slightly before clicking, so that the new poly will be pasted in an offset position. This makes it easier to work with. We will move it into proper position later.



Right click in the draw window to exit "paste" mode and right click again on the edge of our new polygon, then select "properties". The dialog will show the poly as VTP, since we copied our VTP polygon. We need a flatten poly, which is type LWM. Click on the "general" tab, and click on Polygon type of "LWM". This will open the LWM tab. Now select LWM type "flatten", enter the flatten altitude, 241.0935, and click "change". This will enter the correct altitude on all vertices of our flatten polygon. Our flatten is now complete, so click "OK". Now click (on the poly edge) and drag the poly until it coincides with our VTP polygon. Our design work is complete, so save the project.

Step 6: Compile, Install, and Test

We need to compile our new polygons into bgl files. We must select the polygons we wish to compile. Click on the "Point" tool icon to enter pointer mode. We could use the menu Select -> All polygons, but we only need the two polygons we created. One way to limit selection is to drag a selection box in pointer mode, to include only our new polygons. In this image I used menu View, and unselected the "all Maps" for clarity.



I dragged the selection box to include only the new polygons (polygons must be included in their entirety to be selected by this box. If you miss and select too many or too few, just reselect.

We can now compile our selected polygons. Click on the "BGL" tool icon. This brings up a dialog box. There should be checks in the LWM polygons and VTP polygons boxes. Click on the "compile" box, and you should see a command window momentarily display, and then close. This is the SCASM assembler creating the bgl files. The flatten polygons will be compiled into one file (with LWM2 in the file name) and the background texture polygons will be compiled into another file (with VTPP in the file name). By default the compiled files will be placed in the sbuilder\tools\work\ folder. There will also be files with ".sca" file extensions for each bgl file that was created. These are text files with the assembler code that SCASM uses. In advanced work, it is possible to text edit these files and manually compile them with SCASM to fine-tune the design. Save the project one last time. The two bgl files can be moved into an active scenery area \scenery\ subfolder, or a new scenery area can be created and activated using the FS9 scenery library. Note that there is no need for a paired "texture" folder for this project, as we are using default textures for the airport background polygon.

We can now start FS9 and test the result. We should see that the background texture covers our desired area, and the hills that were in the runway are now gone.



And lastly!

This provides some idea of the kinds of terrain modifications that can be done with sbuilder. You can explore the help file to learn more about this program's capabilities.

Thanks to Luis Sa, author of this excellent tool.

Scott Smart