

3DCity: Create 3D city features

Workflow: 3D City – Creation

Version: 1.0

Date: September 12, 2012

Map templates and workflows are ArcGIS resources that can be used to help create 2D / 3D maps and web map applications with your GIS data. These templates and workflows can be applied as the foundation to create similar maps.

This workflow covers how you can generate 3D city features like 3D multipatch building shells and 3D streets from existing GIS data and procedural rules in CityEngine. Starting point is a Geodatabase with 3D GIS features that was created in part 1 of a series of workflows on 3D City Modeling showing you how to:

- Prepare your GIS data for generating 3D city features
- Create 3D city features like 3D multipatch building shells and 3D streets from existing GIS data (this workflow)
- Maintain 3D multipatch building shells
- Design new 3D buildings using procedural rules
- Perform 3D City analysis such as shadow and visual impact
- Publish 3D CityEngine Web Scenes to ArcGIS Online

Please have a look for the full series on the [3DGIS](#) and [CityEngine](#) Resource Centers.

3D Cities Information Model

As mentioned in part 1, we recommend you model your data according to our new 3D Cities Information Model (3DCIM - formerly known as 3DUIM) which will be released in the summer of 2013 so that all our City apps (including the CityEngine rules from part 2 of this series) will work seamlessly with your data. The 3D building shells that we will create in this workflow (Building1Shells) are part of the 3DCIM as well. Have a look on our resource center for a detailed description of the information model.

Use case

3D city models require 3D features such as 3D buildings and 3D streets. And this 3D features need to be tightly linked to the existing operational GIS data that is being used in day to day business (see 3DCIM). The ability to easily create these 3D city features based on existing GIS data is one of the key strengths of CityEngine. The workflow below describes the process of generating 3D building shells, 3D streets, trees and street furniture for Philadelphia from existing GIS data.

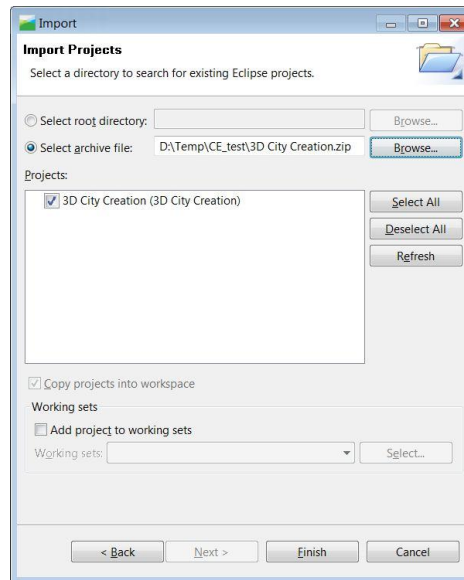
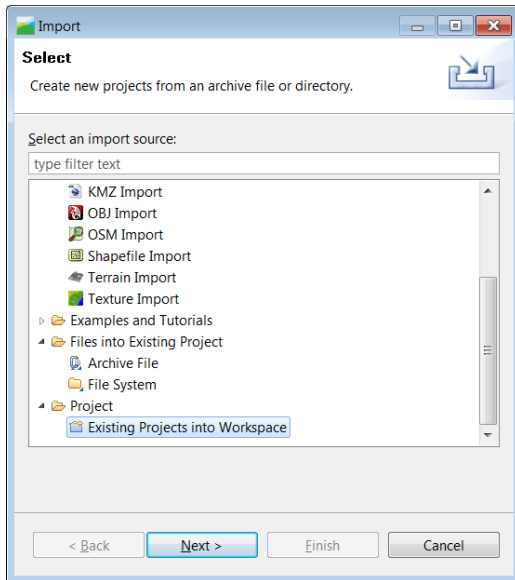
Prerequisites:

- CityEngine 2012.1
- If you don't have a license, go [here](#) to get a trial version
- Pay attention to user interface information (especially how to navigate) when installing

Loading the example:

After installation, startup CityEngine:

Import the '3D City Creation' project into your workspace: File -> import -> Project -> existing project into workspace. Search for the '3DCity_Creation.zip' archive file.

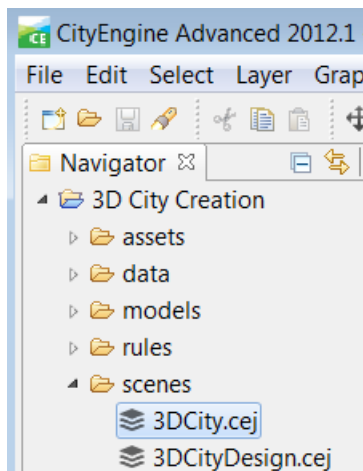


Import existing project (as archive) into workspace

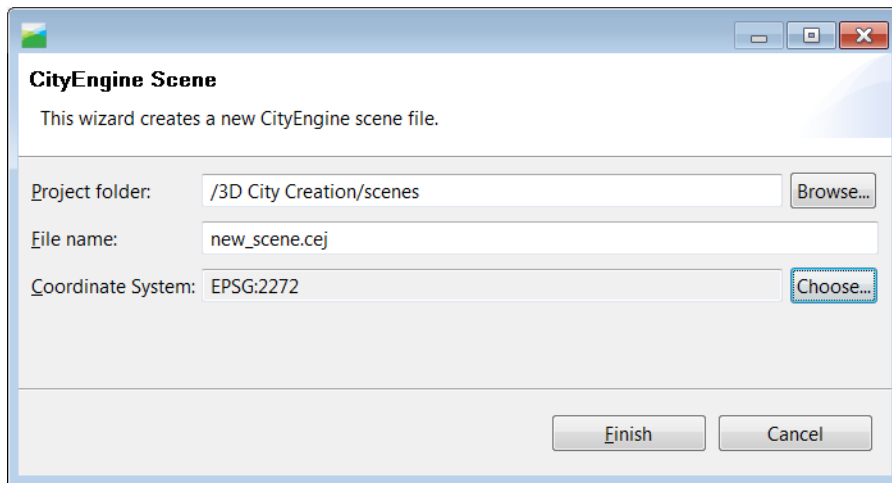
Workflow steps:

1. Create a new scene

This example comes with 2 scenes: the 3DCity scene and a prepared design scene called 3DCityDesign.cej that you can explore at your leisure.



But let's start from scratch now with a new scene. Go to 'File->New->CityEngine->CityEngine scene'

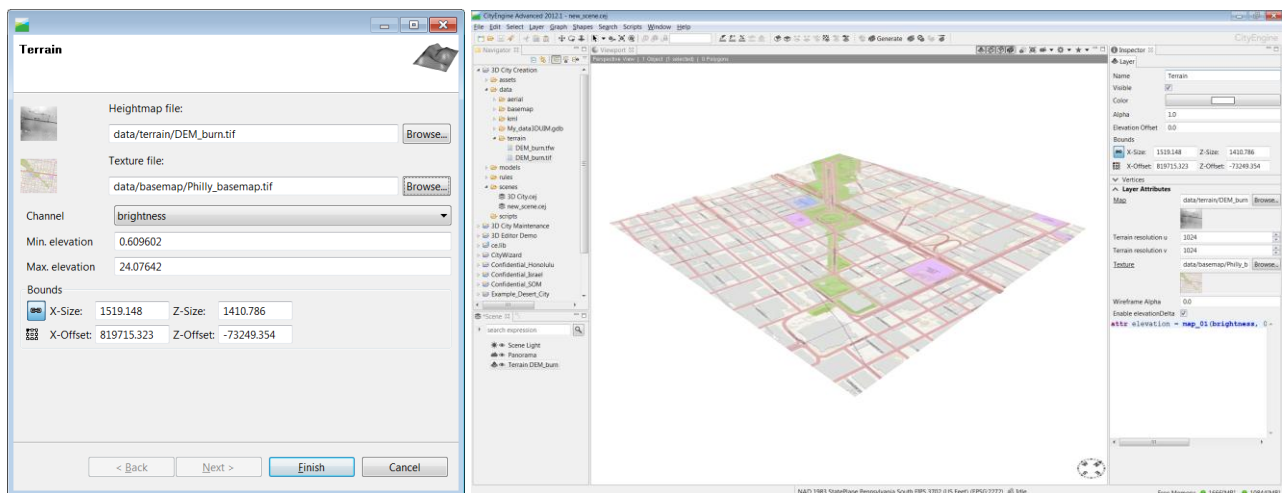


Select the project scenes folder, give it a name and set the coordinate system (EPGS:2272 in this case).

2. Add terrain and texture

First thing we want to do is add a terrain to our scene. In the CityEngine file navigator:

- go to '/3D City Creation/data/terrain' folder and
- drag and drop the DEM_burn.tif into the viewport. This is the elevation tiff we created in part 1 of this workflow
- select same the coordinate system for the terrain (EPGS 2272) and in the same dialog
- select an image to drape on the terrain ('data/basemap/Philly_basemap.tif')
- hit finish and press 'f' in the viewport to zoom to the data

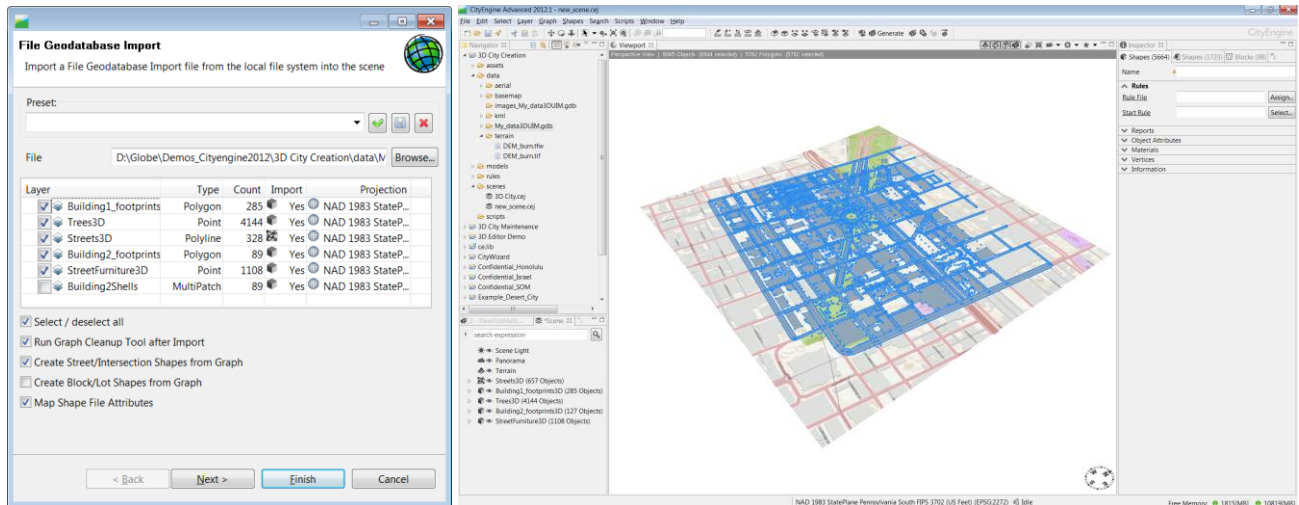


Please note: CityEngine only supports Tiff at the moment as import format for elevation. Therefore you need to clip the terrain and basemap / aerial for your area of interest in ArcGIS and save as Tiff.

3. Add GIS data

Next we want to add the available GIS data. In the CityEngine file navigator:

- go to '/3D City Creation/data/' and
- drag and drop the 'My_data3DUIM.gdb' into the viewport and for now, toggle the Building2Shells off for now
- click finish



We get the following layers in CityEngine:

- 2x building footprints with attributes such as eaveHeight, totalHeight and roofForm (polygons)
- Streets with attributes such as width (lines)
- Vegetation with attributes such as height, type and rotation (points)
- Street furniture with attributes such as size, type and rotation (points)

CityEngine can make use of this data and generate 'close to' as-built geometries by applying simple rules.

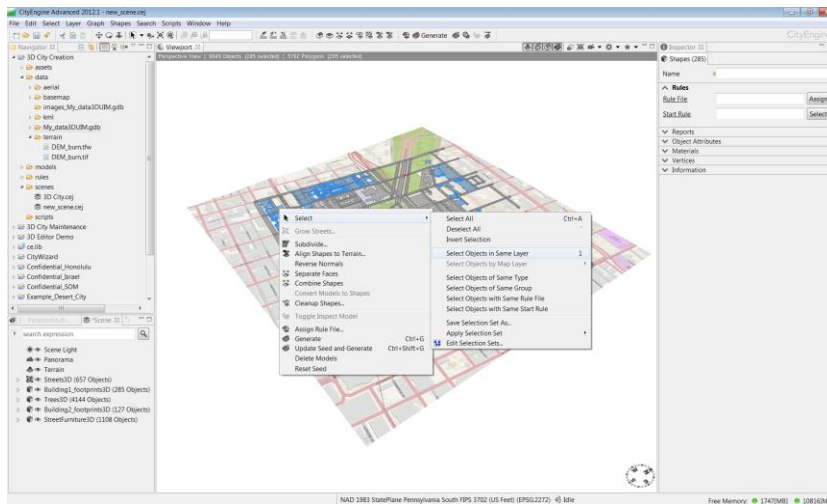
Note: to learn about how to prepare your data for import into CityEngine, please see the 3DCity: Data Preparation workflow on the CityEngine resource center

4. Applying the rules

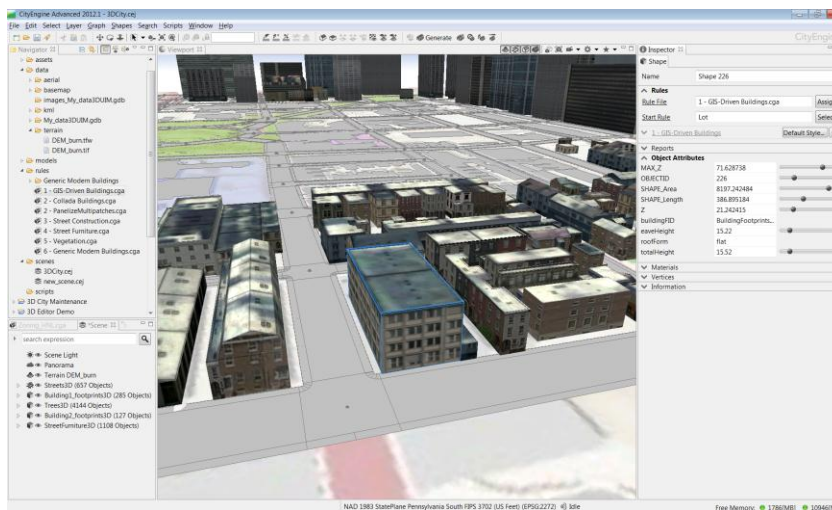
In the next step we are going to apply CityEngine rules to the GIS data, to generate 3D city features. First we are going to use building footprints and attributes eaveHeight, totalHeight and roofForm to recreate the as-built environment. Next we are going to add street furniture, vegetation and create 3D streets from our 2D street center lines. Finally, we'll load static 3D building shells into scene and compare the procedural buildings with the static ones.

Use building footprints and attributes to generate 3D buildings

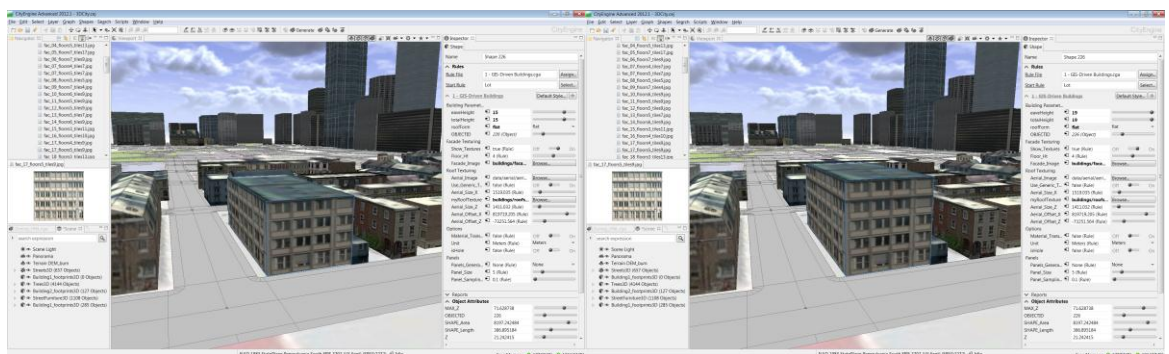
- Select all objects in the Building1_footprints3D layer: right-click on a selected object in the layer and go: Select->Select Objects in the Same Layer



- Drag and drop the 'GIS-Driven buildings.cga' rule (located in the rules folder) on the select building footprints.



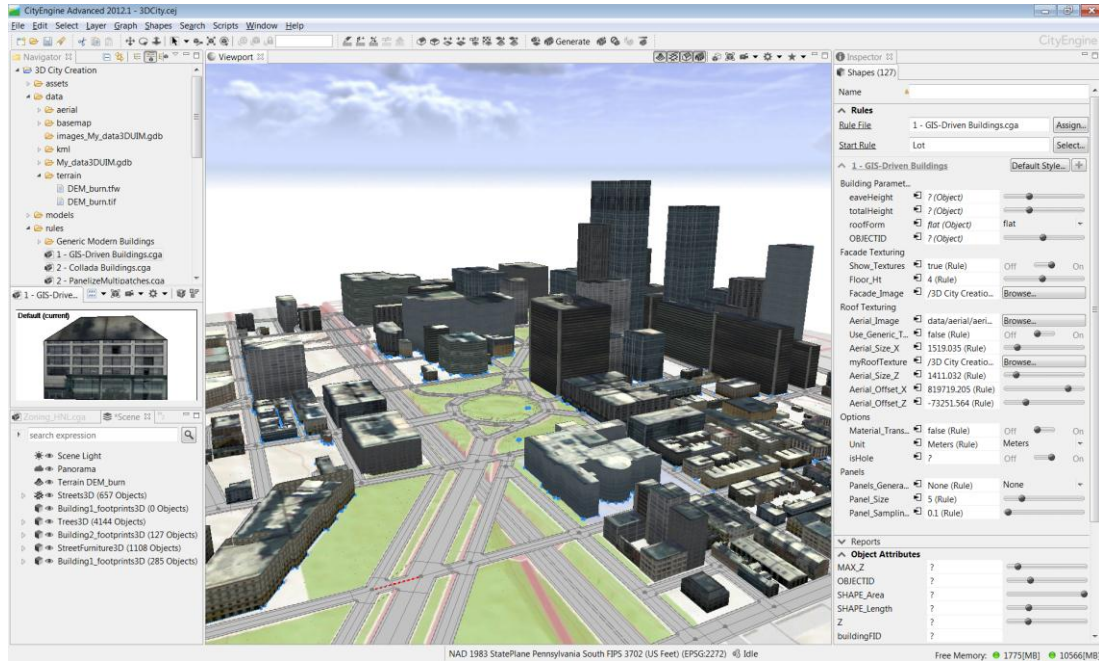
- This rule uses the eaveHeight, totalHeight and roofForm attribute in the GIS data to generate the 3D buildings. The benefit of procedural modeling is that you can easily modify the buildings by changing the rules. For example, adding floor to a building is as easy as:
 - Select the building of interest
 - Go to the Inspector dialog and add 3 meters to the eave and ridge height to add a floor or
 - Change the roof form to mansard and add another 3 meters to the ridge height to create an attic.



Note: when adding a floor (compare images above), the corresponding façade texture showing the extra floor, needs to be present in the assets\buildings\facades directory i.e. fac_17_floors4_tiles9.jpg for 4 floors, fac_17_floors5_tiles9.jpg for 5

floors etc. If such a texture does not exist the rules select another texture in the directory which might fit better than the original one.

- Repeat the process for Building2_footprints3D layer

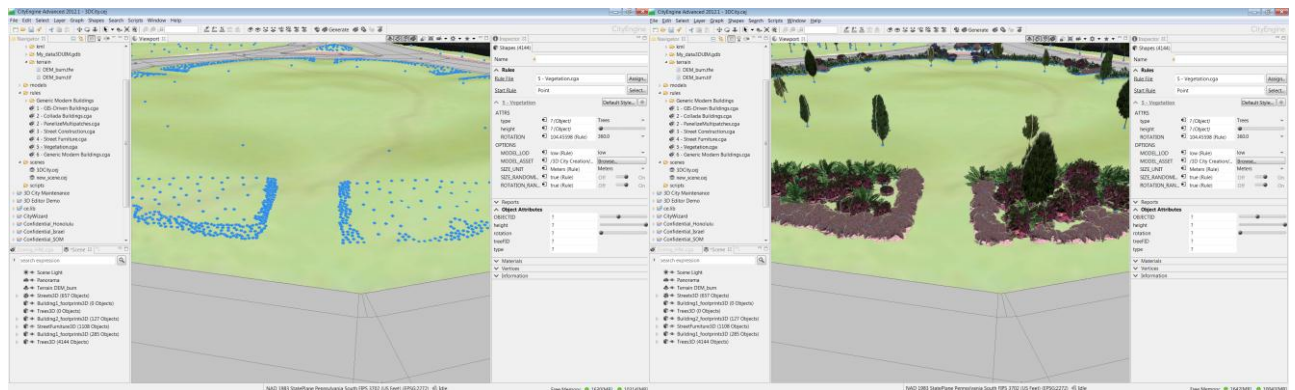


For this feature class, we have less information: no roof form and no eave height so the generated 3D building shells are less complex.

Add vegetation and street furniture

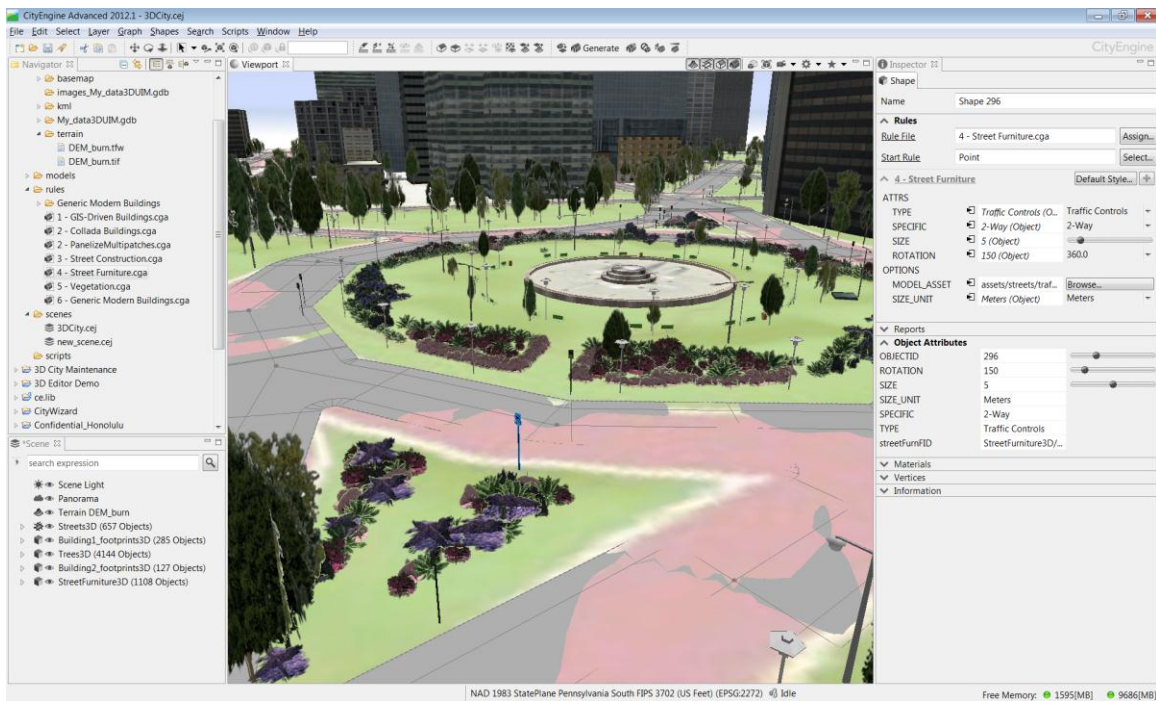
The trees point file in our Geodatabase contains vegetation point locations with attributes size and type. We'll use these attributes to load representative 3D models of the vegetation with the correct size.

- Select all objects in the Trees3D layer: right-click on an object in the layer and go: 'Select->Select Objects in the Same Layer'
- Drag and drop the Vegetation.cga rule file on the selected vegetation points (zoom in closely)



We can do the same for the street furniture layer.

- Select all objects in the StreeFurniture3D layer: right-click on an object in the layer and go: 'Select->Select Objects in the Same Layer'
- Drag and drop the Street Furniture.cga rule file on the selected street furniture points (zoom in closely)

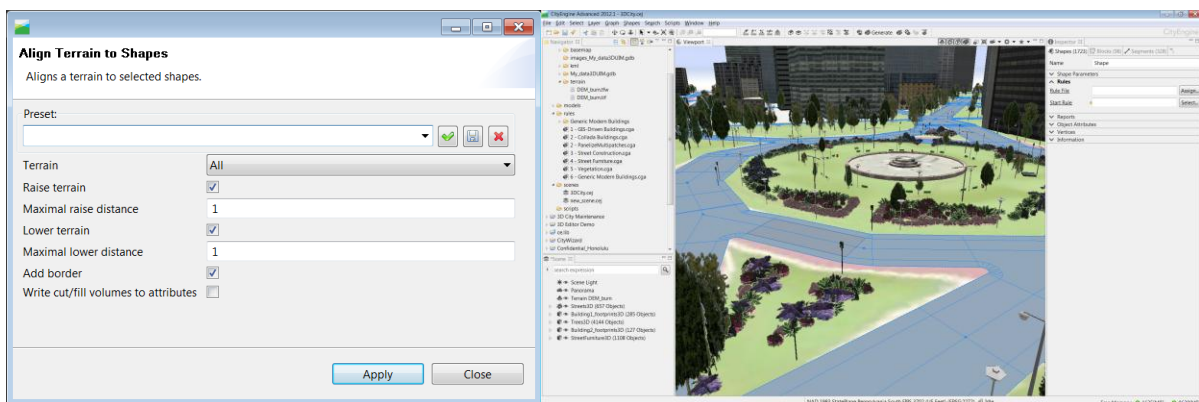


Note: you might have noticed that streets are not aligned nicely to the terrain. So let's fix this and model some 3D streets!

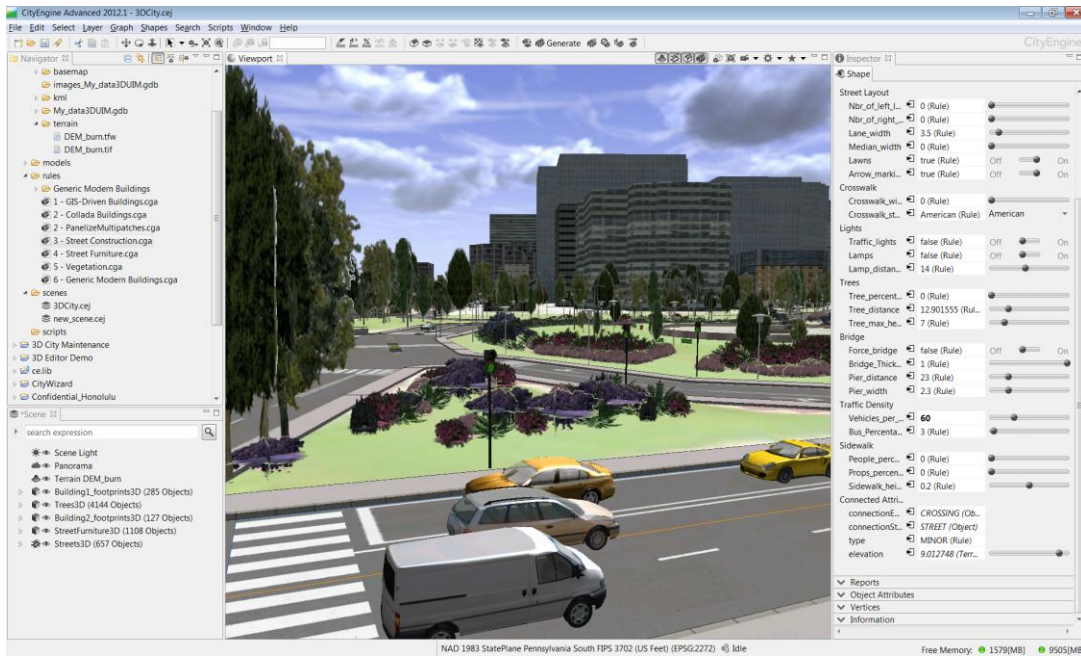
Add streets

As mentioned, the streets do not align nicely with the terrain. This can be caused by low terrain resolution or sparse street vertices. We'll fix this in CityEngine:

- Select all objects in the Streets3D layer: right-click on an object in the layer and go: 'Select->Select Objects in the Same Layer'
- Go to: 'Layer->Align Terrain To Shapes'
- Select a maximum raise distance of 1 meter and hit 'Apply'



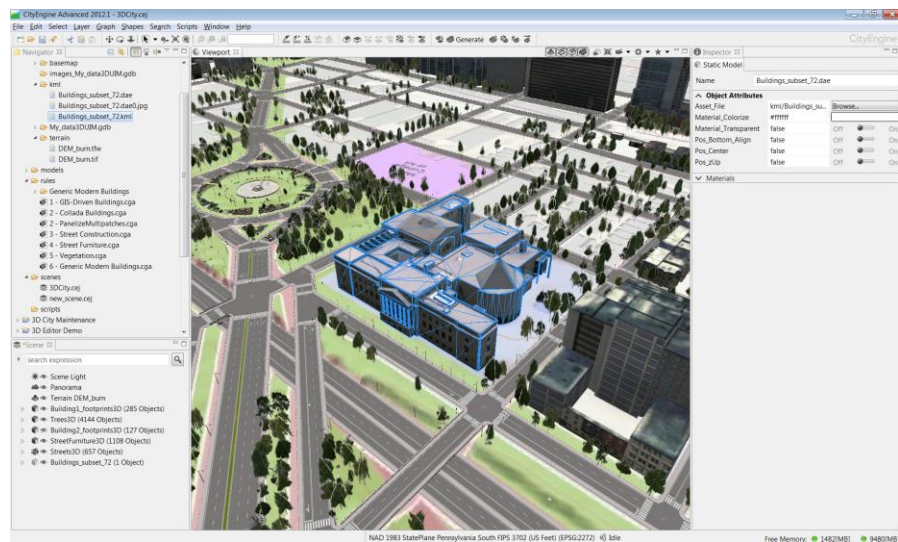
- Drag and drop the Street Construction.cga rule file on the selected streets
- The Street Construction rule is quite an extensive rule with lots of parameters such as 'Vehicles per km' and 'People percentage'. Have a play with changing the parameters...



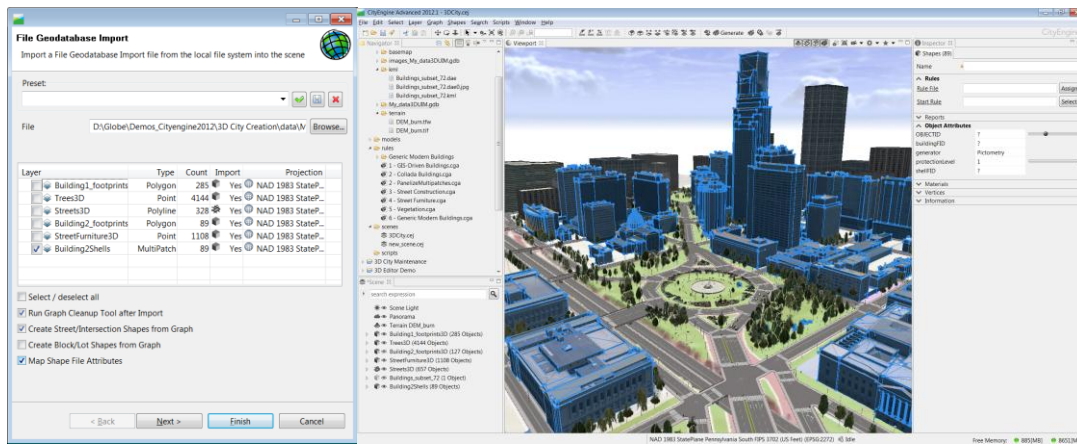
5. Importing existing 3D building shells (data aggregation)

It might be that you already have some 3D models of your buildings, either created internally or by 3rd party providers such as Pictometry. These models can be imported into CityEngine in 2 ways:

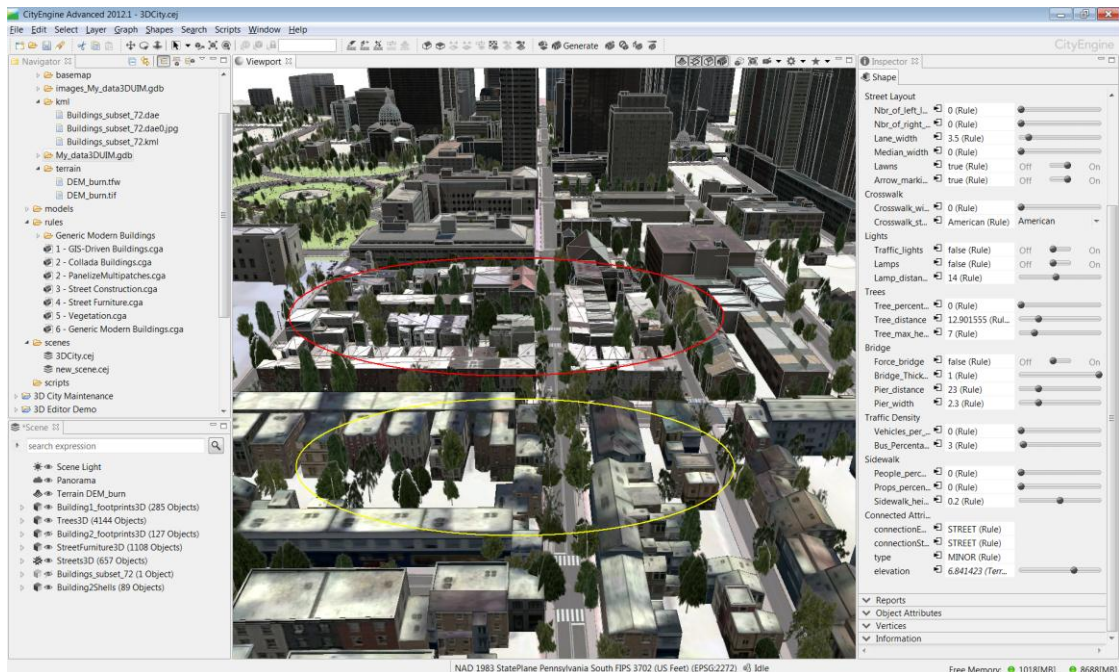
- If the models are georeferenced, for example dae/kml, you can simply 'drag and drop' the kml files into CityEngine. Alternatively import all kml files in a directory using the right-mouse button -> Import... and then type *.kml in the file dialog.
 - Turn off the Building2_footprints3D layer
 - Go to 'data\kml' folder and drag 'n drop the Buildings_subset_72.kml file into the 3D viewport.



- if you have them as multipatches in the Geodatabase, simply load the multipatch feature class in CityEngine. In this case we have static 3D multipatch models to replace the Building2_footprints3D layer.
 - drag 'n drop the My_data3DUIM.gdb in CityEngine
 - toggle all layers off except the Building2Shells layer
 - Turn off the Buildings_subset_72 layer



As you can see, the procedural buildings (yellow) resemble the static 3D models (red) very well with the added benefit being that procedural models can be changed very easily by changing the rule parameters.

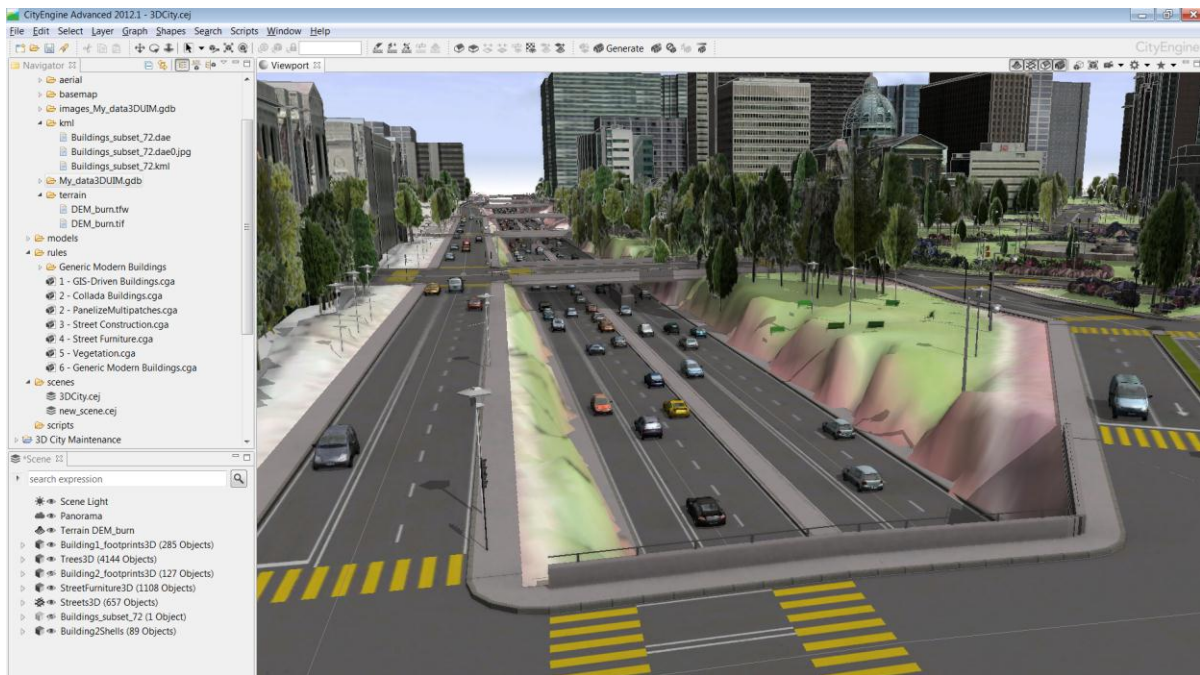


Important to know is that Multipatch features are loaded into CityEngine as 'Initial Shapes', which means you are able to edit them using the new Shape edit tools in CityEngine 2012 and/or apply rules to them. See the 3DCity: Building Maintenance workflow, also available on the [CityEngine resource center](#).

6. Export the result

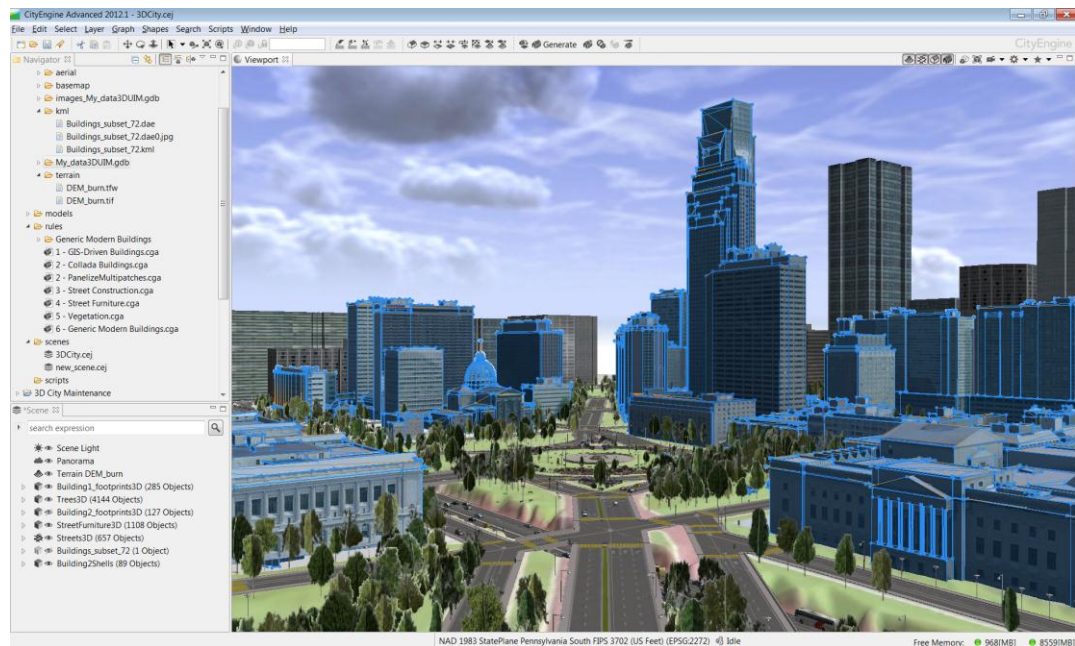
Once you are done creating your 3D city features, you might want create high quality screenshots or save the appropriate 3D models back to the Geodatabase to be part of your 3D city model based on the 3DCIM.

- For high quality screenshots, enable Shadows (8) and Ambient Occlusion (9) in CityEngine. This will affect performance but results in very nice screenshots:

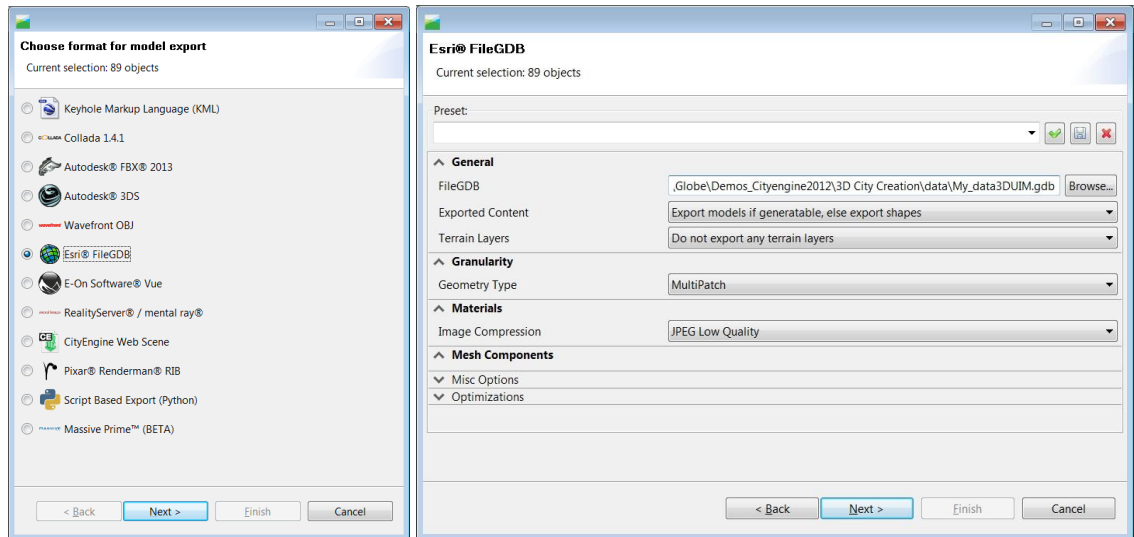


Make sure you turn Shadows (8) and Ambient Occlusion (9) off again after taking the screenshots.

- To bring the 3D models back into ArcGIS simply:
 - select the features you wish to save as multipatch features. In the case of updating your 3DCIM, this would be 3D building shells generated for Building1_footprint3D only. 3D streets have not been defined yet in version 1 of the 3DCIM. Trees and street furniture can be symbolized in 3D directly in ArcGIS through the layer settings.

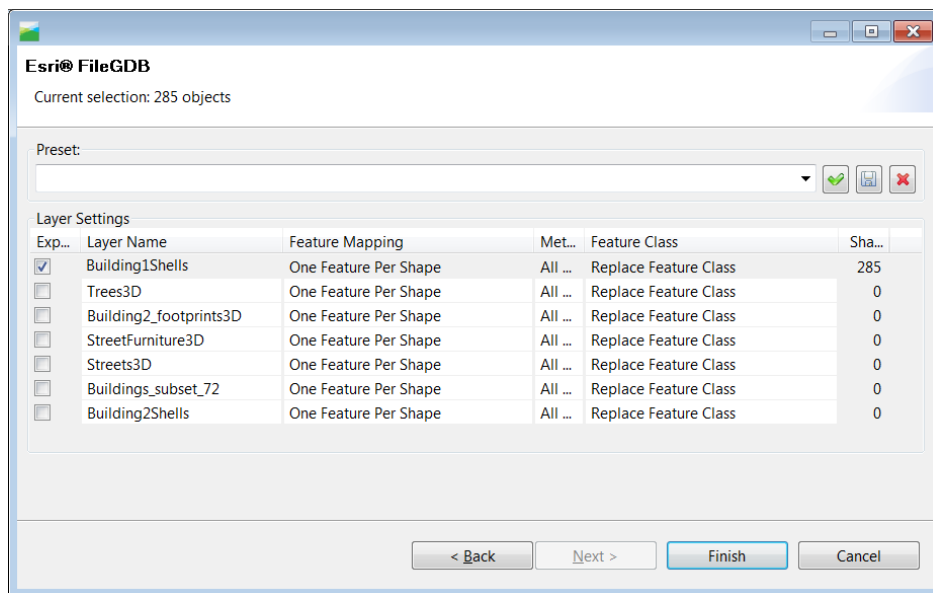


- Go 'File-> Export Models' and select Esri FileGDB. Enter the My_data3DUIM.gdb as output Geodatabase and set texture quality initially to JPEG Low Quality (you can always up the resolution later). Make sure the exported content is set to 'Export models if generatable, else export shapes'.

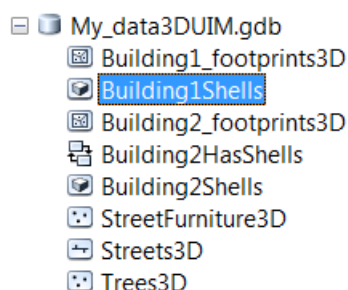


- In the next dialog you get a lot of control over how each layer is exported. You can choose the:
 - Name of the new feature class
 - Feature mapping: how each CityEngine shape is mapped to a GIS feature.
 - Meta data: feature attributes and CityEngine report data
 - Feature class: how the features are written to the Geodatabase

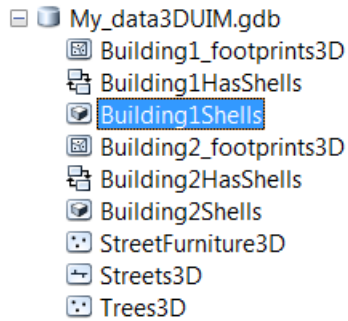
Typically when creating new 3D city features, the default options are good. Make sure you rename the output feature class (in this case Building1_footprints3D to Building1Shells)



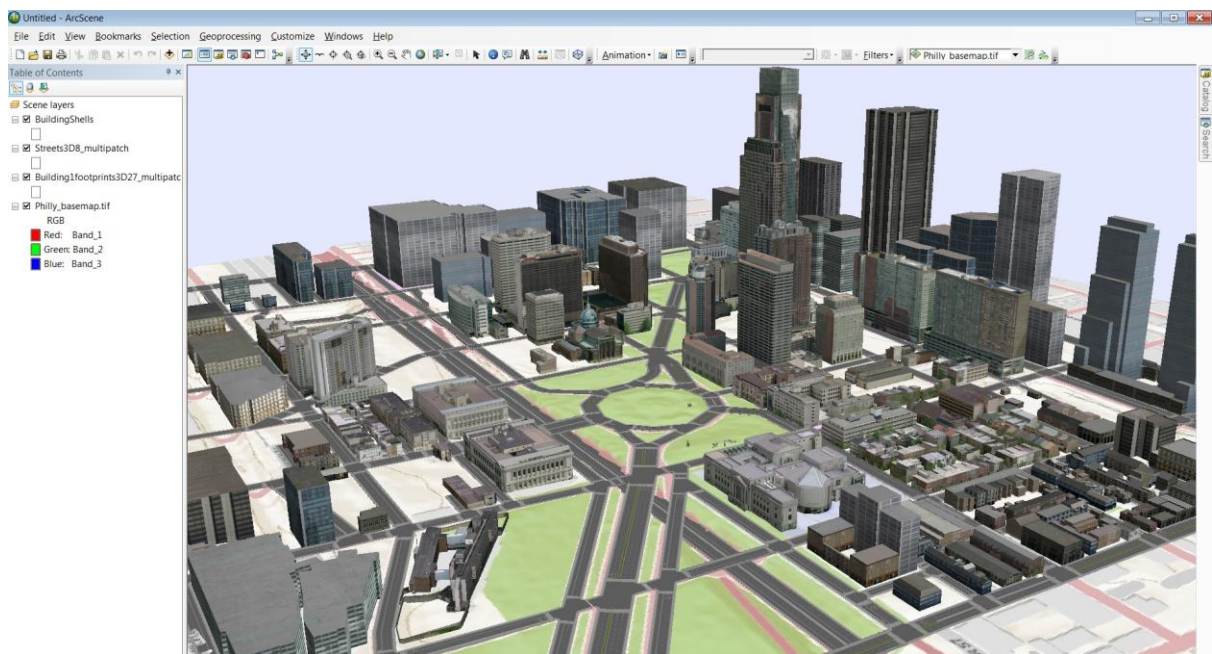
- CityEngine updates My_data3DUIM.gdb with the new Building1Shells feature class.



- Final, you need to link the Building1Shells to the Building1_footprints3D. You can do this by creating a relational table. Make sure you use the buildingFID key to link.



- Streets can be exported in a similar way. Select all the streets, export models to multipatch and load into ArcScene or ArcGlobe to see the result.



Enjoy... The CityEngine / 3D team.