

OKSCAUG 2017 Conference

3D GIS Implementation using City Engine

19 September 2017



The CEDRA Corporation

Implementers of ArcGIS, ArcGIS Online and City Engine Technology

EXPERTISE

EXPERIENCE

COMMITMENT

The following presentation describes using Esri's City Engine software to develop a 3D GIS from 2D GIS. The presentation begins with a few background slides on creating a 3D GIS and its applicability, while the remaining slides discuss the process that was used in creating the 3D model.

3D GIS Implementation using City Engine

Presentation Agenda

Formal Presentation

- Slides discussing how City Engine can be used to develop a 3D GIS.
- Commentary based upon our experience creating a 3D GIS using COE data.
- Interactive presentation. *Feel free to ask questions during the presentation.*

Question and Answer Session

- Responses to questions.



EXPERTISE EXPERIENCE COMMITMENT

During the presentation we encourage questions being asked. This leads to a more interactive presentation. So feel free to ask questions at any time.

3D GIS Implementation using City Engine



Typical Questions

How do we get to this?

How long does it take?

How difficult?

How much?



EXPERTISE

EXPERIENCE

COMMITMENT

Let's begin by showing an image of what we want to achieve. That is, a realistic 3D image of our GIS. As we go through the presentation we will explain each of the components comprising this image. In reviewing the image typical questions most people ask are shown in the slide. Hopefully by the end of the presentation we will have addressed these questions. For those interested, the south end of the image is the intersection of Broadway and 15th.

3D GIS Implementation using City Engine

Before starting the 3D development, the first Question to be Answered is:

How much detail is to be displayed in the 3D GIS?

The less detail to be displayed, the less time and less cost to create the model.

The greater the detail, the more time and cost to create the model.

As the level of detail increases the more realistic the model appears.

Model implies an **approximation**, not an exactness.



EXPERTISE EXPERIENCE COMMITMENT

The first thing to keep in mind in developing a 3D GIS is the more detail you desire, the more time intensive and expensive the model building effort will be. Additionally, one word that you will hear a lot with a 3D GIS is “model”. This is important because model implies an approximation, not an exactness.

3D GIS Implementation using City Engine

Level of Detail (LOD) Definition as defined by Esri



Level of Detail	1	2	3	4
Definition:	3D Extrusion	3D Extrusion with Roof Form or Building Shell	High Detail 3D Building Models	Interior Spaces and Floors
Data Required:				
Building Footprints	# of Stories or Total Height & or Usage	# of Stories or Total Height or Usage & if available Roof Height	3D Building Shell with Fine details (Textured or Untextured)	3D Building Shell Interior Spaces (CAD or BIM)
Streets	Street Centerlines	Street Centerlines with Width attributed	Detailed Streets (CAD)	Street Furniture, Stop Lights, Signage, Infrastructure lines (GIS/CAD)



EXPERTISE EXPERIENCE COMMITMENT

Here's an Esri slide that illustrates the 4 levels of detail that Esri associates with a 3D GIS. As you can see, as you move up in level the more information that is included in the model. This also has the effect of the more realistic the 3D GIS becomes. So when creating a 3D GIS, consideration should be given as to what level of detail is the 3D GIS to fall into.

3D GIS Implementation using City Engine

Applications of a 3D GIS

Realistic Visualizations.

Comprehensive City Planning
Activities such as:

Flood Modeling.

Visibility Analysis of
New Developments.

Line of Sight analysis.

Redevelopment Scenarios



EXPERTISE EXPERIENCE COMMITMENT

Once the decision has been made to develop a 3D GIS the visualization benefits that are available can not be understated. Here is an image from the Philadelphia Example which is available for download from the City Engine gallery. As a matter of fact, there are a number of examples that are available for download from the City Engine gallery. The rules in these samples can be used as is or modified in other 3D models, if desired.

3D GIS Implementation using City Engine

The following datasets are typically used in creating a 3D GIS:

- | | |
|--|--|
| • Ortho-imagery, | Basemap Image |
| • Digital Terrain Model - DTM, | 3D Ground Surface Layer |
| • Digital Surface Model - DSM, | 3D Surface Layer |
| • Normalized Digital Surface Model - nDSM, | Difference between Ground and Surface |
| • Street Centerlines, | Road centerlines with pavement width |
| • Building Footprints, | Outlines of building footprints |
| • Parcel Data, | Parcel boundaries |
| • Vegetation Data, | Point locations of vegetation features |
| • Zoning/Land Use Data, | Zoning/Land Use delineations |
| • Land Cover Data. | Land Cover delineations |



EXPERTISE EXPERIENCE COMMITMENT

The first step in building the 3D GIS is determining which datasets are to be included. The list shown are typical datasets people utilize. Obviously other datasets may be desired such as utilities, such as, sewer, water, storm water, electric, street signs, street lights, etc.

3D GIS Implementation using City Engine

Ortho-imagery / DEM – Two Options

Use in-house developed/purchased data

Use ArcGIS Online from within City Engine

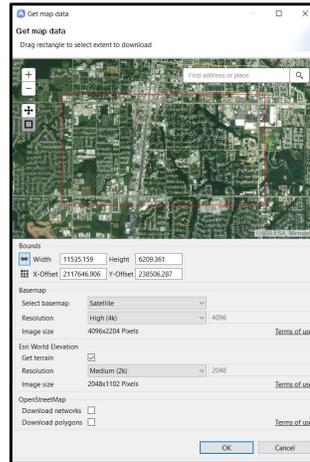
Get Map Data... command

City Engine 2016 or greater

Data is stored in Maps folder

Elevation as TIFF (DEM)

Texture as JPG (Imagery)



Regarding the basemap there are a couple options. The first is using an in-house ortho-image or cartographic basemap. Now this basemap could be 2D or 3D. That is to say, the basemap may or may not contain elevation information. The second option involves using ArcGIS Online. As can be seen from the slide, using the ArcGIS Online option allows the user to define the area of interest by defining a rectangle. We'll discuss this further in a subsequent slide.

3D GIS Implementation using City Engine

Option 1

Use in-house developed or purchased Ortho-imagery



EXPERTISE

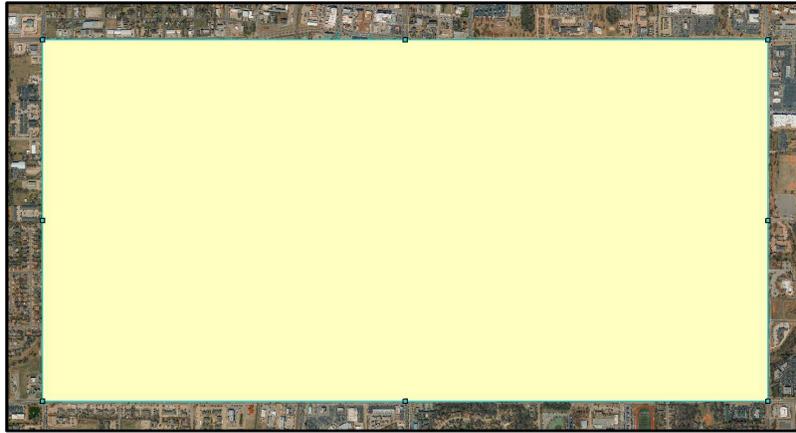
EXPERIENCE

COMMITMENT

Let's examine the first option where in-house basemap information is to be used.

3D GIS Implementation using City Engine

COE Ortho-Imagery with Clipping Polygon for 140335 and 140336



EXPERTISE EXPERIENCE COMMITMENT

Typically a 3D GIS is a subset of the overall 2D GIS. As such, we will want to clip the overall basemap to the desired area of interest. This can be accomplished by defining a rectangle covering the desired area. Note that this rectangle is a graphic that the user interactively creates within ArcMap.

3D GIS Implementation using City Engine

COE Ortho-Imagery Clipped Using the Export Data and Selected Graphic

The screenshot illustrates the process of exporting raster data from a GIS application. On the left, a context menu is open over a layer, with the 'Data' option selected. The 'Export Data...' option is highlighted, and a tooltip explains that it exports raster data from the layer to a specified format, allowing for customization of extent, spatial reference, and cell size. On the right, the 'Export Raster Data - Edmond_MrSID_Mosaic.tiff' dialog box is displayed. It shows the following settings:

- Extent:** Selected Graphics (Clipping) is selected.
- Spatial Reference:** Raster Dataset (Original) is selected.
- Output Raster:** Use Renderer is checked. Cell Size (x, y) is set to 0.25. Raster Size (columns, rows) is 42394 x 21134. NoData as is set to Z96.
- Properties:** Name: Edmond_Clip.tiff; Bands: 3; Pixel Depth: 8 Bit; Uncompressed Size: 2.50 GB; Extent (left, top, right, bottom): (2118228.8975, 238060.9068, 2128827.3527, 232777.4506); Spatial Reference: NAD_1983_StatePlane_Oklahoma_North_FIPS_3501_Feet.
- Location:** D:\City_Engine\Projects\Edmond_Test\images
- Name:** Edmond_Clip.tiff; Format: TIFF
- Compression Type:** NONE; Compression Quality (1-100): 75

Buttons for 'Save' and 'Cancel' are visible at the bottom of the dialog box.

Using the Export Data command the user is able to generate a bitmap representing the area of interest.

3D GIS Implementation using City Engine

COE Ortho-Imagery clipped to extent of 140335 and 140336



EXPERTISE EXPERIENCE COMMITMENT

In this example we have clipped the overall City ortho-imagery to the extents of sections 140335 and 140336.

3D GIS Implementation using City Engine

Option 2

**Use ArcGIS Online from within City Engine
to obtain Ortho-imagery and DSM**

 EXPERTISE EXPERIENCE COMMITMENT

Now, if there is no existing basemap information in-house or if the user desires to use ArcGIS Online, the second option previously mentioned is available.

3D GIS Implementation using City Engine

ArcGIS Online Data

Map Data... command

City Engine 2016 or greater

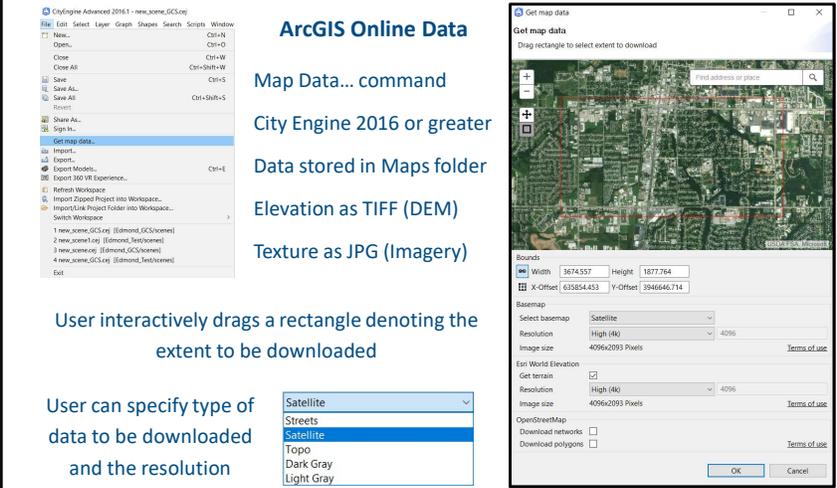
Data stored in Maps folder

Elevation as TIFF (DEM)

Texture as JPG (Imagery)

User interactively drags a rectangle denoting the extent to be downloaded

User can specify type of data to be downloaded and the resolution



The screenshot shows the City Engine 2016.1 software interface. On the left, the 'Get map data...' command is highlighted in the menu. On the right, the 'Get map data' dialog box is open, showing a map with a red rectangle indicating the area to be downloaded. The dialog box includes fields for 'Width' (3674.537) and 'Height' (1877.764), and 'X-Offset' (635854.453) and 'Y-Offset' (3946466.714). Under 'Basemap', 'Satellite' is selected. Under 'Elevation', 'Get terrain' is checked. Under 'OpenStreetMap', 'Download networks' and 'Download polygons' are unchecked. A dropdown menu shows options for 'Satellite', 'Streets', 'Satellite', 'Topo', 'Dark Gray', and 'Light Gray'.

EXPERTISE EXPERIENCE COMMITMENT

In this mode of operation, the basemap can be obtained directly from within City Engine using the Get map data... command. The end result will be the same as with the first option. That is, a clipped rectangular image representing the area of interest. Note that generally speaking, in-house developed imagery will probably be more current and of higher resolution than that provided by Esri. However, if there is no existing basemap, this is an excellent resource for obtaining basemap data.

Basic Feature Layer Datasets

Tree Locations

Street Lights

Street Centerlines

Building Footprints

Depending upon preference, data can be 2D or 3D



As mentioned previously, there are a number of datasets that can be imported into the 3D GIS. During this presentation we will concentrate on trees, street lights, street centerlines, and building footprints. Additionally, the user will want to consider whether 2D or 3D feature data is to be imported into City Engine.

3D GIS Implementation using City Engine

Street Light Layer Considerations

Point features denoting approximate center of street light

Use the Rule File for Rendering Street Lights from the City of Philadelphia Example
[/rules/Street Furniture.cga](#)

Rule Contains a Variety of Street Lights

360Bulb	Globe
LightwFlag	Split and Tall

Attributes for facilitating generation of
Street Lights (case sensitive)

- RULEFILE
- STARTRULE
- TYPE
- SPECIFIC
- SIZE
- ROTATION



EXPERTISE EXPERIENCE COMMITMENT

Like the trees, street lights can be imported into City Engine. The attribute assignment can be done within City Engine or the underlying database can be modified to incorporate the Rule's pertinent attributes.

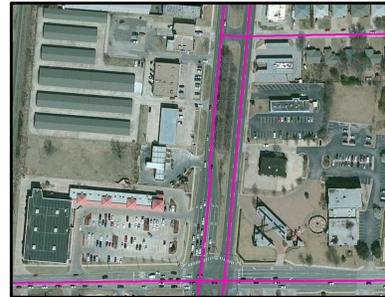
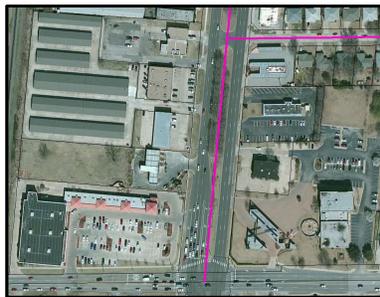
3D GIS Implementation using City Engine

Street Centerline Considerations

One Line denoting the centerline of the ROW

One Line for each Roadway Direction

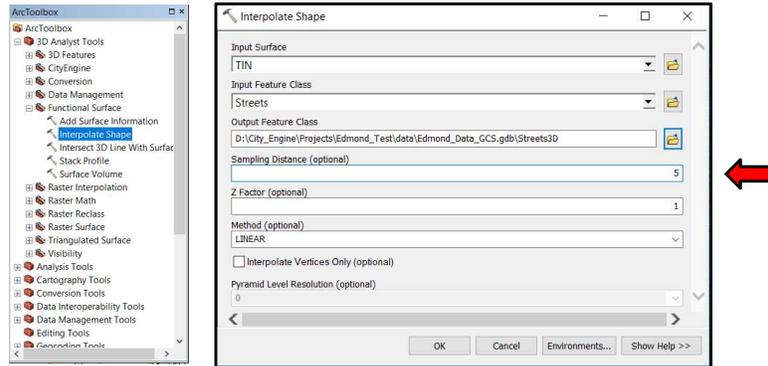
Recommend doing
Street editing in
ArcMap prior to
importing into CE



EXPERTISE EXPERIENCE COMMITMENT

It should be noted that prior to importing street centerline data, consideration should be given as to how the centerlines are to be represented. City Engine does offer the ability to model medians. As such, it may be appropriate to create a centerline that splits the street ROW rather than having two lines representing the road direction. In either case, the editing of the street centerline data will be easier in an ArcMap environment.

3D GIS Implementation using City Engine



If 3D data is desired, 2D datasets should be converted into 3D using the 3D Analyst Interpolate Shape toolbox command

If 3D feature data is desired then the Interpolate Shape command can be used to convert 2D feature data into 3D feature data. Note the input surface, TIN. The TIN dataset was created using the 3D Analyst extension from the DSM that was downloaded from ArcGIS Online. Furthermore, note the Sampling Distance parameter. Since there is no draping functionality in City Engine, it is important that the 3D feature data contains enough intermediate vertices to properly reflect the surface of the terrain.

3D GIS Implementation using City Engine

COE Streets, Building Pads and Parcels within extent of 140335 and 140336



EXPERTISE EXPERIENCE COMMITMENT

So summarizing, we are going to take the above ArcMap datasets, trees, street lights, streets, building footprints, parcels, and import them into City Engine to create a 3D GIS.

3D GIS Implementation using City Engine

3D Development assisted by GIS Featureclasses containing pertinent Data

Streets

- Pavement Width
- Lane Width

Building Footprints

- Total Height
- Number of Floors
- Floor Height

A File GeoDatabase was established containing the various featureclasses.

Depending upon the City Engine Rule the pertinent data will vary.

The attributes shown here are the most basic and pertinent.

City Engine is a Rule Based Modeler comprised of a Rule File and a Start Rule which can be stored as attributes in the featureclass.

A City Engine Rule can be defined to extract a parameter's value from a feature's attribute value. All numeric values are in meters, not feet.


EXPERTISE
EXPERIENCE
COMMITMENT

In developing the 3D GIS, the development process can be assisted by incorporating modeling information into the ArcMap featureclasses. That is to say, by adding certain attributes and populating them with the appropriate values the time to generate the 3D GIS can be greatly reduced.

3D GIS Implementation using City Engine

The Attribute Name must match the Variable Name in the Rule File

Complete_Street.cga Rule File

Lane_Width : double in meters

LOD_Setting : text value equal to High, Moderate, Low

3D_City_Design_Rules/Building Construction.cga Rule File

Floor_Count_Min : integer value > 0

Floor_Count_Max : integer value >= Floor_Count_Min

Rule File and Start Rule Attribute Names

RULEFILE : Rule File name attribute

STARTRULE : Start Rule name attribute

Variable Names in a Rule File are Case Sensitive

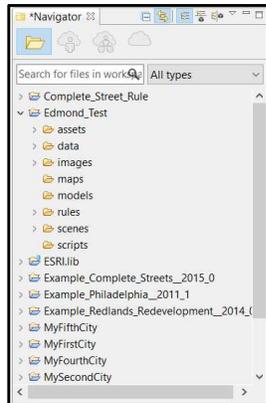


EXPERTISE EXPERIENCE COMMITMENT

Here are just a few attribute names that can appear for the Complete_Street and Building Construction Rule Files. Note the LOD_Setting attribute. This indicates how detail the texturing will be when rendering a feature. The higher the texture level the more memory that is needed. This has a major effect on the performance of City Engine. Finally, it is possible to store the Rule File and Start Rule values under the RULEFILE and STARTRULE attribute names.

3D GIS Implementation using City Engine

City Engine Workspace



- CE Workspace comprised of 1 or more Projects.
- Each Project contains fixed set of folders.
- Project Data stored in specific folders.
- Assets contains the 3D images and textures.
- Data contains Project datasets.
- Images contains ortho-imagery.
- Rules contains the CGA files used to generate the 3D models.
- Scenes contains the 3D views.



EXPERTISE EXPERIENCE COMMITMENT

Those familiar with ArcMap's desktop interface will find City Engine's interface a little different. First of all, City Engine operates in a workspace, which is nothing more than a folder on a hard drive. Within the workspace, there can be one or more projects. Again, projects are nothing more than folders within the workspace folder. Each project is comprised of a fixed set of folders containing specific types of information. Note that it is possible for projects to reference information/data in other projects. For example, rules created in Project A can be accessed in Project B. In so doing duplicate information is avoided.

3D GIS Implementation using City Engine

Getting Started – Importing Data

2D City Engine Model (No DSM imported)

Import 2D Streets, 2D Parcels, 2D Buildings and Ortho-imagery for background

3D City Engine Model (DSM imported)

Import 3D Streets, 3D Parcels, 3D Buildings and Ortho-imagery for background

Import DSM

Changes made to the GIS data are not reflected in the City Engine model, the model will need to be updated manually or by reimporting the GIS data.

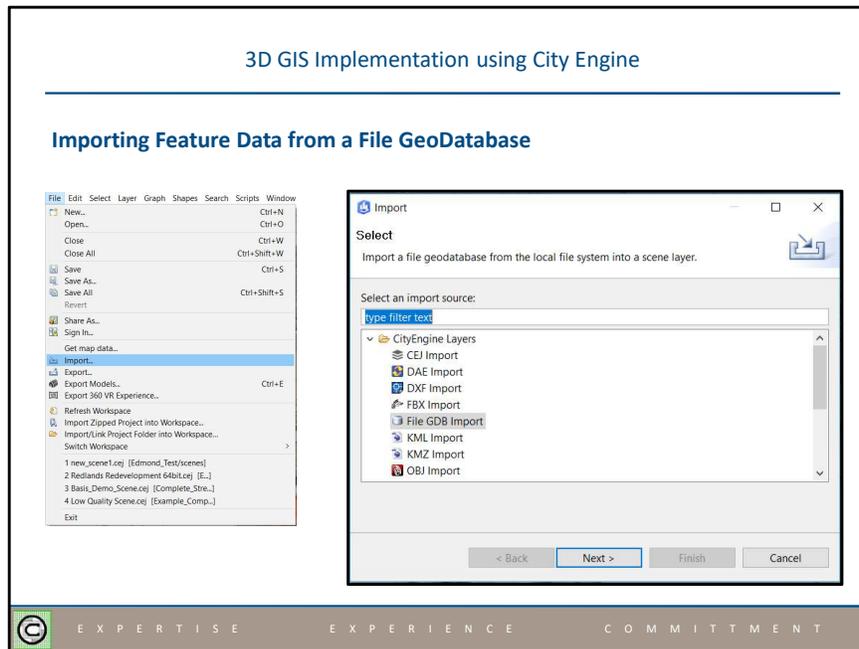


EXPERTISE EXPERIENCE COMMITMENT

How ArcMap data is imported into City Engine will depend on whether 2D or 3D feature data is to be processed and if a DSM is involved. If a DSM is to be used, 2D or 3D feature data can be processed. If a DSM is not to be used, 2D feature data and ortho-imagery can be imported. Note the comment at the bottom of the slide, if the GIS changes, the City Engine model will need to be updated manually or by reimporting the data.

3D GIS Implementation using City Engine

Importing Feature Data from a File GeoDatabase



Regardless if 2D or 3D feature data is to be imported, the Import command will be used to get ArcMap data into City Engine. Depending upon the ArcMap source, the appropriate command should be selected. In our example, we will use the File GDB Import command since our information is stored in a File GeoDatabase.

3D GIS Implementation using City Engine

File GDB Import Dialog Boxes

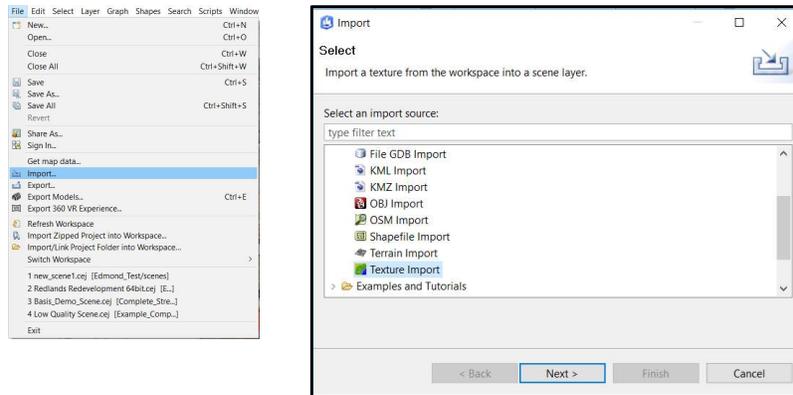
The Threshold Angle controls how sensitive the import will be in generating curves. The higher the number, the more curves.

The image displays three dialog boxes from the City Engine software. The 'File Geodatabase' dialog is on the left, showing a list of datasets with columns for Name, Type, Class, Road, CS Author, and CS ID. The 'Simplify Graph Attributes' dialog is in the center, featuring a 'Threshold Angle' slider set to 10. The 'Cleanup Graph' dialog is on the right, with various checkboxes and input fields for graph cleaning settings. A blue arrow points from the text above to the 'Threshold Angle' slider.

In importing File GeoDatabase featureclasses, the user has the option of importing one or more featureclass at a time. There are also a number of options that can be activated during the import. The settings of these parameters has a tremendous effect on the import process. So be prepared to try a few variations to determine the proper values.

3D GIS Implementation using City Engine

Importing Ortho-imagery Data



To import ortho-imagery, the Texture Import command should be selected.

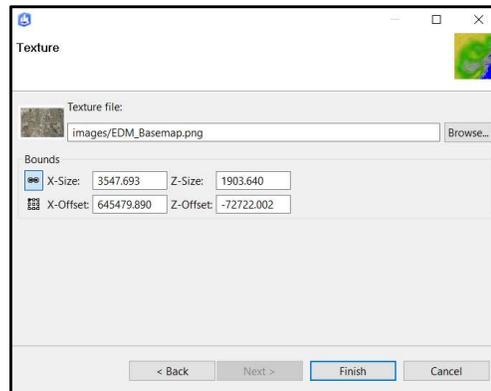
3D GIS Implementation using City Engine

Importing Ortho-imagery Data – Zero Elevation Based

Dialog box for importing the clipped ortho-imagery discussed earlier.

No elevation information is associated with this dataset.

Used solely for visual purposes.



EXPERTISE EXPERIENCE COMMITMENT

In our example we simply specify the clipped ortho-imagery file that was created using the Export Data command in ArcMap.

3D GIS Implementation using City Engine



Initial Display of GIS 2D Data in City Engine

The next tasks are to clean up any street geometry issues followed by assigning the Rule File and Start Rule for the features in the model



EXPERTISE EXPERIENCE COMMITMENT

Once the datasets have been imported this is what the display will look like. In this example there is no DSM so all of the data resides at elevation 0. The next step will be to clean up any geometry issues and there more than likely will be issues. Following this step, the user can then go through the process of assigning rules to the features. It is the rules that make the 3D images appear.

3D GIS Implementation using City Engine

Street Geometry Issues/Post-processing in City Engine

Dead end streets will have cul-de-sacs introduced.
Change Intersection Type for the node from Smart to Freeway.

Intersections that are close to each other result in white area being displayed since road can not be modeled.
Change Intersection Type for the node from Smart to Crossing or Junction depending upon geometry.



EXPERTISE

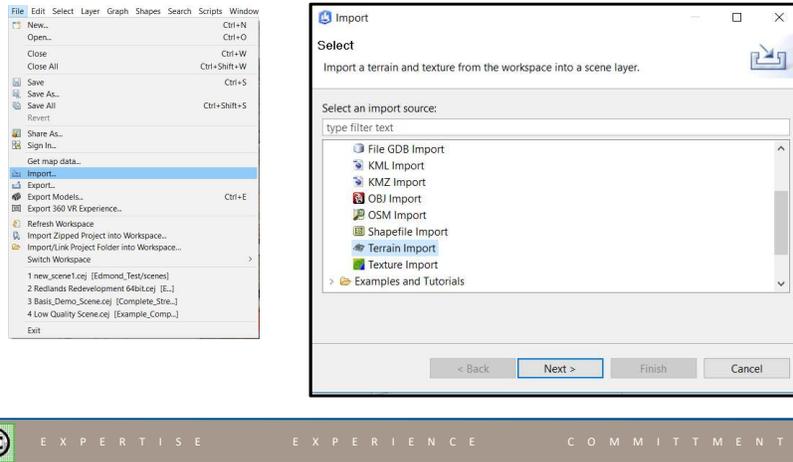
EXPERIENCE

COMMITMENT

The majority of the streets will be imported properly but there will be areas that will need to be addressed manually.

3D GIS Implementation using City Engine

Importing Terrain Data – Elevation Data Present



If a DSM is to be imported, the Terrain Import command can be selected.

3D GIS Implementation using City Engine

Importing Terrain Data

Dialog box for importing the ArcGIS Online data discussed earlier.

Comprised of a DSM and an image or texture as referred to in City Engine.

Used for visual and elevation computation purposes.

Terrain

Heightmap file: /Edmond_GCS/maps/Terrain_Satellite/elevation.tif Browse...

Texture file: /Edmond_GCS/maps/Terrain_Satellite/texture.jpg Browse...

Channel: brightness

Min. elevation: 328.13895

Max. elevation: 369.48297

Bounds

X-Size: 3656.968 Z-Size: 1961.533

X-Offset: 645343.632 Z-Offset: -72778.535

< Back Next > Finish Cancel



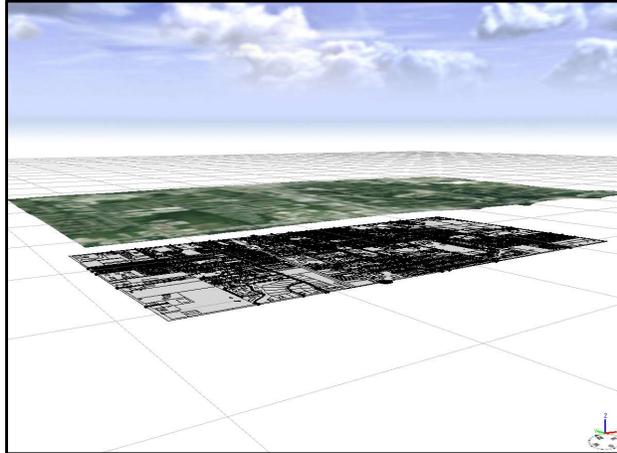
EXPERTISE EXPERIENCE COMMITMENT

In this example we are importing the terrain we downloaded from ArcGIS Online. The DSM and Image files will be stored in the MAPS folder within the City Engine project folder.

3D GIS Implementation using City Engine

2D GIS data imported with the ArcGIS Online DSM and Ortho-imagery

2D GIS data can be projected onto the DSM using the Align Shapes to Terrain and the Align Graph to Terrain commands.

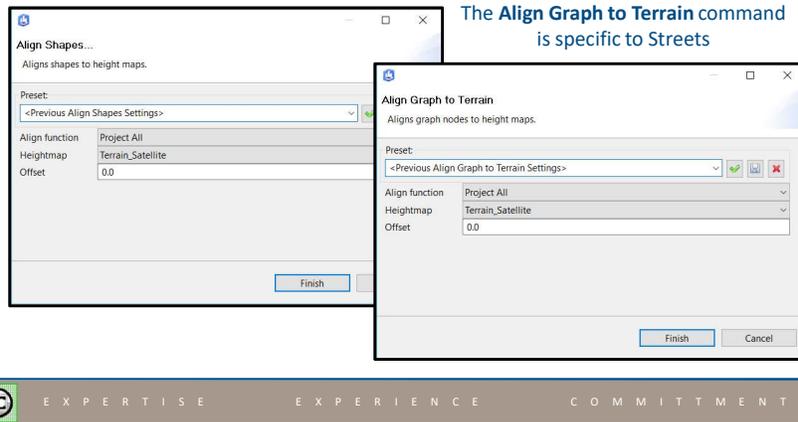


EXPERTISE EXPERIENCE COMMITMENT

If a DSM is imported in conjunction with 2D feature data, you will find a display such as that shown in the slide. The DSM has elevation information while the 2D data is at elevation 0. City Engine has commands that will project the 2D feature data onto the DSM.

3D GIS Implementation using City Engine

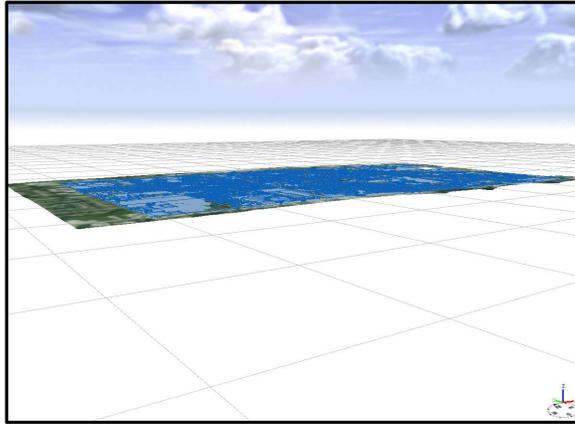
The **Align Shapes to Terrain** and the **Align Graph to Terrain** commands enable the user to specify the Heightmap dataset and an offset from the DSM. The offset is used to eliminate interference with the DSM.



Specifically, the Align Shapes to Terrain and Align Graph to Terrain commands are used to project 2D feature data onto the DSM. The Align Graph to Terrain command is specific to street centerlines. The offset parameter enables the user to offset the features from the DSM by a specific amount in the Z direction.

3D GIS Implementation using City Engine

Result of executing the **Align Shapes to Terrain** and **Align Graph to Terrain** commands.

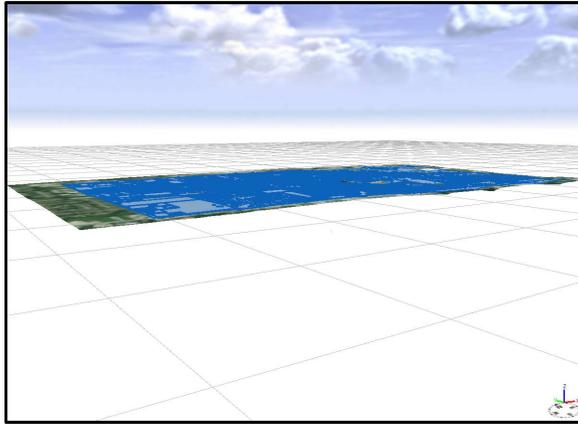


EXPERTISE EXPERIENCE COMMITMENT

Once the two commands have been executed, the 2D feature data will have been converted to 3D data and aligned with the DSM.

3D GIS Implementation using City Engine

Result of importing 3D Feature Layers.



EXPERTISE EXPERIENCE COMMITMENT

In the case of 3D feature data, the features should align with the DSM upon import. No additional processing should be required. However, if need be, the Align Shapes to Terrain and Align Graph to Terrain commands can still be utilized.

3D GIS Implementation using City Engine

Left side image represents the 2D GIS data projected onto the ArcGIS Online DSM.

Right side image represents the 3D GIS data which had interpolated vertices computed every 5 feet superimposed onto the ArcGIS Online DSM.

3D GIS data better fits the DSM due to the interpolated vertices.



EXPERTISE

EXPERIENCE

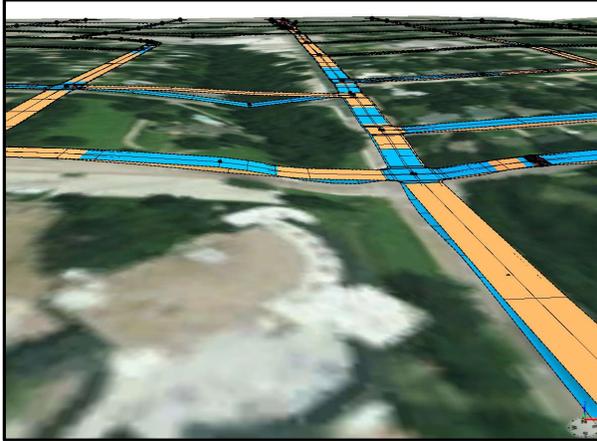
COMMITMENT

Previously we mentioned the Sampling Distance parameter when converting 2D feature data into 3D feature data. This parameter introduces intermediate vertices on a feature. The image on the left side represents 2D data with start and end points, no intermediate vertices. The image on the right side represents 3D feature data with intermediate vertices. Notice how the features in the right side image better follow the terrain of the DSM.

3D GIS Implementation using City Engine

The Gold roadway represents the projected 2D GIS data while the Blue roadway denotes the 3D GIS data with interpolated vertices.

If a DSM is to be used in the 3D Model the GIS Data should be comprised of 3D features with interpolated vertices



EXPERTISE EXPERIENCE COMMITMENT

Just to make the point clearer, the image above shows the 2D feature data in gold, while the 3D feature data is displayed in blue. The take away from this is, if a DSM is to be used, intermediate vertices should be inserted so that features properly align with the DSM. Note that the closer the intermediate vertices are, the better the features will follow the terrain. The downside is, the larger the dataset becomes.

3D GIS Implementation using City Engine

Overview of Rule Files and Rules

- City Engine was designed for procedural modeling (computer graphics terminology for creating 3D models and textures from a set of rules).
- Computer Generated Architecture (CGA) shape grammar is a programming language.
- Rule Files are written in CGA and contain one or more rules.
- Each feature is assigned a Rule File and a Start Rule which denotes how the feature is modeled/displayed.
- If a model contains 100 unique buildings there will be 100 rules.
- Esri has a Web Page containing City Engine Examples. The Rules in these examples can be copied into other project folders for use and/or modification.
- Models created in Maya, Rhino, 3DS or Sketchup can be used in City Engine.



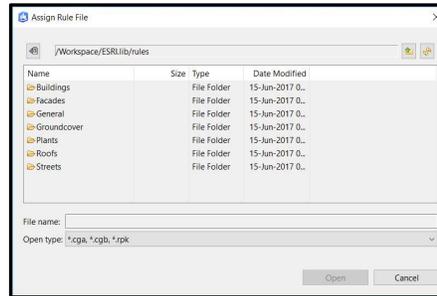
EXPERTISE EXPERIENCE COMMITMENT

With the feature data imported in City Engine, the modeler can now go about the process of creating/assigning the Rule Files and Rules to the features. This assignment can be done on a feature by feature basis or to a group of selected features in one operation. City Engine renders a feature based upon the rule assigned to the feature. Note that it is possible to use 3D models created by another 3D modeler in City Engine.

3D GIS Implementation using City Engine

Esri Default Rule Files and Rules

- Rules for a variety of different feature types.
- Rules generally not associated with feature attributes but rather use random number generation to create different “looks”
- Very useful in generating non-reality scenes
- Need to be modified to represent real world conditions.



Creating Rules can be complicated and one does need some sort of programming, as well as 3D modeling, experience. However, City Engine does come with some pre-defined rules that can be used out of the box. Unfortunately these rules will not generate models that represent real world conditions. The image in the slide illustrates what entities have pre-defined rules. Within each of the folders shown in the image are one or more Rule Files.

3D GIS Implementation using City Engine

Esri Default Rule Files for Buildings Plants Streets

The image displays three overlapping 'Assign Rule File' dialog boxes, each showing a list of files in a specific folder. The top-left dialog is for the Buildings folder, the middle one for Streets, and the bottom-right one for Plants. Each dialog has a 'File name' field and an 'Open type' dropdown menu.

Name	Size	Type	Date Modified
Building_From_Footprint.cga	10 KB	text/x-cga	15-Jun-2017 0...
Building_From_OpenStreetMap.cga	17 KB	text/x-cga	15-Jun-2017 0...
Building_Mass_Texturizer.cga	2 KB	text/x-cga	15-Jun-2017 0...

Name	Size	Type	Date Modified
Street_Modem_Simple.cga	4 KB	text/x-cga	15-Jun-2017 0...
Street_Modem_Standard.cga	14 KB	text/x-cga	15-Jun-2017 0...

Name	Size	Type	Date Modified
3D Plants by e-on software Plant Factory - www.e-onsoftware.com	11 KB	text/x-cga	15-Jun-2017 0...
Plant_Distributor.cga	29 KB	text/x-cga	15-Jun-2017 0...
Plant_Loader.cga			

EXPERTISE EXPERIENCE COMMITMENT

Each of the folders shown in the previous slide contains one or more Rule Files. It is these Rule Files that the user selects.

3D GIS Implementation using City Engine

Example of assigning the Esri Building from Footprint and Street Modern Standard Rules.

A random generator assigns building heights and styles. For streets, different street markings are assigned.



EXPERTISE EXPERIENCE COMMITMENT

Here is a slide illustrating the model that is generated using the Esri Building from Footprint and Street Modern Standard Rule Files.

3D GIS Implementation using City Engine

Example of assigning the Esri Building from Open StreetMap Rule using the GenericMediumTown style.

Default Style
GenericMediumTown
GenericSmallTown
GenericSmallVillage
NewYork
Vancouver
Zurich
Wellington
Monaco
Riomaggiore

Add new style...
Preview & select styles...



EXPERTISE EXPERIENCE COMMITMENT

Here is a slide illustrating the model that is generated using the Esri Building from Open StreetMap and Street Modern Standard Rule Files. With the Building from Open StreetMap Rule File the user has a few options as to how the buildings should be modeled. In this example, the GenericMediumTown option is displayed.

3D GIS Implementation using City Engine

Esri Rule File for Streets – Complete_Street.cga

- Example Complete Streets Features 2015 released by ESRI Redlands.
- Comprised of the following Start Rules:

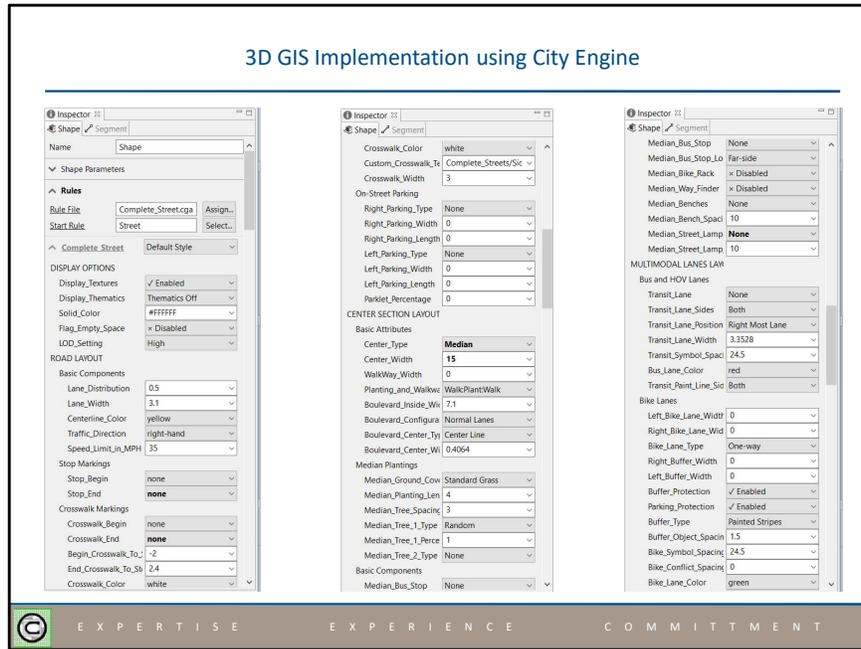
Street	Junction
Bridge Crossing	Median_TreeSplit
Crossing	Roundabout
Freeway	RoundaboutIsland
Freeway Entry	Sidewalk
Joint	
- Based on the Rule different look and different set of parameters.
- The Streets Rule has an extensive set of parameters as shown in the next 2 slides.



EXPERTISE EXPERIENCE COMMITMENT

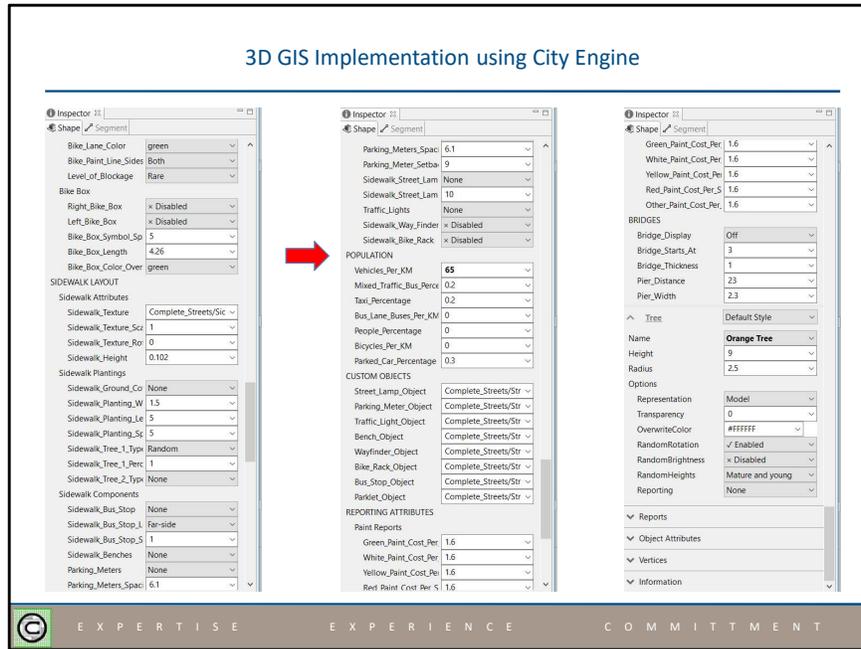
In addition to the default Rule Files, Esri offers the Complete_Street.cga Rule File. This is a robust Rule File that can serve as a starting point for rendering streets.

3D GIS Implementation using City Engine



As can be seen, there are a number of parameters the user can modify to achieve the desired look. Note that distances such as Lane Width are specified in meters, not feet. This applies to all types of distances.

3D GIS Implementation using City Engine



The Population category enables the user to add images of vehicles and people to the model.

3D GIS Implementation using City Engine

Esri Rule File for Streets – Complete_Street.cga – Available Tree Names

Alder Buckthorn	European Larch	Rose	PROXIES	Hophornbeam
Amazon Sword Plant	Ficus	Ruffe Palm		Huckleberry Shrub
American Chestnut	Field Elm	Saguaro Cactus	Algarrobo	Japanese Hemlock
American Sycamore	Flannelbush	Sassafras	American Elderberry	Japanese Nutmeg
Apricot	Flowering Dogwood	Scots Pine	American Pepper	Judas Tree
Australian Pine	Giant Sequoia	Sea Islands Yucca	American Silverberry	Lawson Cypress
Baldcypress	Hedgehog Agave	Shadbush	Athel Tamarisk	Loblolly Bay
Balsam Fir	Japanese Angelica Tree	Snake Plant	Avocado	Mexican Buckeye
Bamboo	Lacy Tree Philodendron	Southern Magnolia	Black Tupelo	Necklacepod
Banana Tree	Leyland Cypress	Spanish Broom	Buttonbush	Northern Bilberry
Basswood	Lily of the Valley	Strawberry Tree	Canada Buffaloberry	Northern White Cedar
Bay Laurel	Lodgepole Pine	Sugar Maple	Chinaberry Tree	Octopus Tree
Black Locust	Mediterranean Buckthorn	Sunflower	Chinese Tallow Tree	Osage Orange
Blue Gum Eucalyptus	Mexican Palmetto	Sweetgum	Common Hackberry	Paper Bark Tree
Boxwood	Mountain Mahogany	Umbrella Acacia	Common Holly	Pawpaw
Cabbage Palm Fern	Northern Red Oak	Western Juniper	Common Persimmon	Persian Silk Tree
California Bay	Norway Maple	White Ash	Desert Bitterbrush	Princess Tree
California Incense Cedar	Norway Spruce	White Oak	European Hornbeam	Smooth Sumac
California Palm	Orange Tree	White Poplar	Giant Chinquapin	Sourwood
California Redwood	Orchid	White Willow	Honey Locust	Southern Wax Myrtle
California Walnut	Oval-leaved Privet	Witch Hazel	Hophornbeam	Tanoak
Coconut Palm	Palm Lily		Huckleberry Shrub	Tree of Heaven
Common Hawthorn	Palo Verde	GENERIC	Japanese Hemlock	Turkish Hazel
Common Whitebeam	Paper Birch	Generic Dead Tree	Japanese Nutmeg	Western Soapberry
Conker Tree	Parlour Palm	Generic Stump	Judas Tree	White Mulberry
Date Palm	Prickly Pear Cactus	Generic Unknown	Lawson Cypress	Yellow Poplar
Desert Willow	Red Alder		Loblolly Bay	Yew
Douglas Fir	Red Hickory		Mexican Buckeye	
European Beech	Rhododendron Azaleas		Necklacepod	o Rule default
European Larch	Rose		Northern Bilberry	-- Connect Attribute...



EXPERTISE EXPERIENCE COMMITMENT

A rather extensive list of tree names are available as well. These can be used when specifying Median and Sidewalk plantings.

3D GIS Implementation using City Engine

Various Design Scenarios

By adjusting the values of the Complete Street parameters the user is able to quickly generate various design scenarios. In this example, (a) no median, (b) median with planting and (c) a median with barrier and shoulder and side street parking are shown.



EXPERTISE

EXPERIENCE

COMMITMENT

Using the parameters in the Complete_Street Rule, various roadway designs can be generated.

3D GIS Implementation using City Engine

Esri Rule File for Buildings – 3D_City_Design_Rules

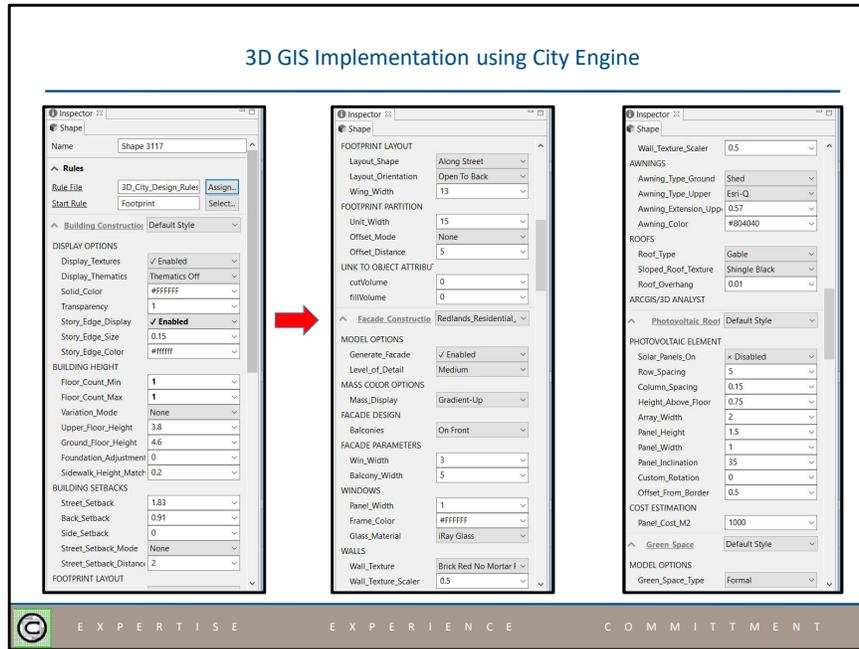
- Example Redlands Redevelopment 2015 released by ESRI Redlands.
- Comprised of a number of Rule Files
- Based on the Rule different look and different set of parameters.
- Included with the Redlands Redevelopment Example.
- Comprised of the following Start Rules:
 - Footprint
 - Parcel
 - Lot
- The Building Construction Rule has an extensive set of parameters as shown in the next 2 slides.



EXPERTISE EXPERIENCE COMMITMENT

Moving on to Buildings, in the Redlands Redevelopment 2015 example there are a number of Rule Files. The Building Construction Rule File pertains to buildings, as the name implies. We decided to use this Rule File because of its robustness in being able to generate various types of building facades.

3D GIS Implementation using City Engine



As can be seen there is an extensive list of parameters. A Key parameter is the Facade Construction parameter which controls the building style that is generated.

3D GIS Implementation using City Engine

The image displays three screenshots of the City Engine Inspector interface, showing various parameters for a 'Common Hackberry' tree object. The parameters are organized into several sections:

- Green_Space_Type:** Formal
- FRONT_YARD:** Front_Yard_Type: Sidewalk, Front_Hardscape_Texture: Concrete Bright, Front_Hardscape_Scale: 1, Front_Hardscape_Rotation: 0
- PATHWAYS:** Unit_Width: 15, Pathway_Type: Paver Grey Asphalt, Rotation: 0, Pervious_Hardscape: Disabled, Pathway_Scale: 2, People_Percentage: 20
- VEGETATION:** Grass_Scale: 1, Grass_Type: Random, Hedge_Type: Hedge Standard, Hedge_Percentage: 70, Tree_Percentage: 5, Max_Trees_Per_Acre: 200, Tree_Height: 0, Tree_Type: Random
- LINK TO OBJECT ATTRIBUTES:** cutVolume: 0, fillVolume: 0
- Greenspace_Rules:** Tree: Default Style, Name: Common Hackberry
- Name:** Common Hackberry
- Height:** 0
- Radius:** 0
- Options:** Representation: Model, Transparency: 1, OverrideColor: #FFFFFF, RandomRotation: Enabled, RandomBrightness: Disabled, RandomHeights: Mature and young, Reporting: None
- Zoning:** Zoning_Display: Building, Envelope_Transparency: 0.4, Story_Edge_Display: Enabled, Story_Edge_Size: 0.15, Story_Edge_Color: #FFFFFF
- USAGE:** Zone_1_Floor_Count: 0, Zone_1_Usage: Commercial, Zone_2_Floor_Count: 0, Zone_2_Usage: None, Zone_3_Floor_Count: 0, Zone_3_Usage: None
- 3D FORM - TRANSECT:** Transect: T4 General Urban
- 3D FORM - HEIGHT LIMIT:** Height_Method: Limit Height to Floor
- Height_Method:** Limit Height to Floor
- Max_Height:** 19
- Floor_Count_Min:** 1
- Floor_Count_Max:** 1
- Ground_Floor_Height:** 4.6
- Upper_Floor_Height:** 3.8
- Roof_Height:** 3
- 3D FORM - SETBACKS:** Street_Setback: 1.83, Street_Height: 4.6, Street_Angle: 50, Back_Setback: 0.91, Back_Height: 4.6, Back_Angle: 50, Side_Setback: 0, Side_Height: 4.6, Side_Angle: 50
- Building_Performance:** Default Style
- TARGET ECO-CRITERIA:** Percent_Reduction_Water: 0, Percent_Reduction_Electricity: 0, Percent_Reduction_Heating: 0, Percent_Reduction_CO2: 0, Percent_Reduction_CO2e: 0, Percent_Greywater_Recovery: 0
- BUILDING COST ESTIMATION:** Cost_Per_M2: 1400
- Reports:**

More parameters for the Building Construction Rule File.

3D GIS Implementation using City Engine

Facade Construction Parameter for the Building Construction Rule

- Offers 14 Building Styles.
- Mixed Use, Residential and Office
- Accounts for Green construction.

Redlands_Mixed_Use_1
Redlands_Mixed_Use_Green_Bu
Redlands_Mixed_Use_2
Redlands_Mixed_Use_3
Redlands_Mixed_Use_Eco_2
Redlands_Residential_1
Redlands_Residential_2
Redlands_Office_Eco_1
Redlands_Office_1
Redlands_Mixed_Use_Green_Bu
Redlands_Residential_3
Redlands_Residential_4
Redlands_Office_2
Redlands_Residential_5

Add new style...
Preview & select styles...



EXPERTISE EXPERIENCE COMMITMENT

Using the Facade Construction parameter the user can generate different building styles.

3D GIS Implementation using City Engine

Facade Building Styles 1 through 5 for the Building Construction Rule



Redlands_
Mixed_Use
_1

Redlands_
Mixed_Use
_Green_Bui
lding_1

Redlands_
Mixed_Use
_2

Redlands_
Mixed_Use
_3

Redlands_
Mixed_Use
_Eco_2



EXPERTISE

EXPERIENCE

COMMITMENT

Here are examples of the types of building facades that can be generated. Note that the user can control the number of floors, window size, balcony size and so forth.

3D GIS Implementation using City Engine

Facade Building Styles 6 through 11 for the Building Construction Rule



Redlands_
Residential
_1

Redlands_
Residential
_2

Redlands
Office
Eco_1

Redlands
Office
1

Redlands_
Mixed_Use
_Green_Bui
lding

Redlands_
Residential
_3



EXPERTISE

EXPERIENCE

COMMITMENT

The second set of building styles.

3D GIS Implementation using City Engine

Facade Building Styles 12 through 14 for the Building Construction Rule



Redlands_Residential_4

Redlands_Office_2

Redlands_Residential_5



EXPERTISE

EXPERIENCE

COMMITMENT

The final set of building styles.

3D GIS Implementation using City Engine



Three Building Rule Files displayed in this view:

Generic Modern Buildings

Building

Building Construction



EXPERTISE EXPERIENCE COMMITMENT

In addition to the Building Construction Rule File within the 3D City Design Rules there are other building rule files that can be downloaded and accessed. The Rule Files mentioned in this slide were obtained from the Philadelphia 2011 Example, the Complete Streets 2015 Example and the Redlands Redevelopment 2015 Example, respectively. Another building rule file that can be considered is the `/ESRI.lib/rules/Buildings/Building_From_Footprint.cga` Rule File. This offers a number of different building styles with a much simpler user interface. This rule file is very applicable in cases where it is desired to assign a common building facade based upon zoning type.

3D GIS Implementation using City Engine

Buildings – Modeling Approaches

- Create a Rule for every *Building*.
 - Yields a very detailed and realistic model
 - Very time consuming and expensive
- Create a Rule for every *Type of Building*.
 - Yields an accurate model based upon use
 - Cost-effective
- Utilize Digital Photos as building facade texturing.
 - Yields a very detailed and realistic model
 - Very time consuming and expensive



EXPERTISE EXPERIENCE COMMITMENT

Summarizing, depending upon how exact you want the model to look you can create a rule for every building or a rule for every type of building. Obviously, the less number of rules you have to create the faster the model building process will be. You can also download rules from the City Engine gallery and use those rules directly or customize them. Furthermore, the ability to utilize a digital photo of a building facade can be employed.

3D GIS Implementation using City Engine



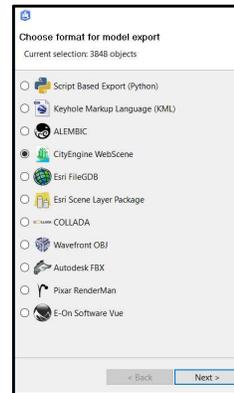
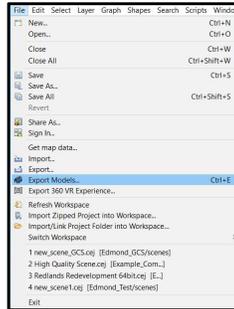
EXPERTISE EXPERIENCE COMMITMENT

One last image illustrating the 3D terrain and ortho-imagery downloaded from ArcGIS Online with the 3D street centerlines and 3D building footprints. The streets have been assigned the Complete Streets Rule and the building footprints the Building from Open StreetMap Rule using the GenericMediumTown style. The most time consuming part of creating this model was the geometry post-processing of the streets in City Engine.

3D GIS Implementation using City Engine

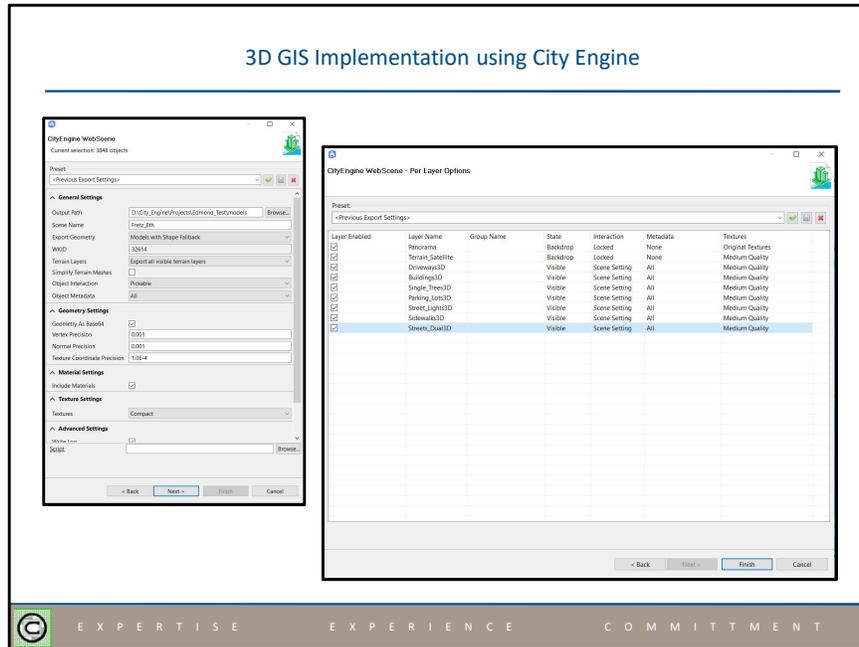
Creating a Web Scene for Viewing in a Browser (Chrome, Firefox, IE, etc.)

- Two Step Process.
- Create a City Engine Web Scene (3ws). Note, must select objects to be exported first.
- Once the 3ws File is created upload the file to ArcGIS Online or to an internal Server for hosting.



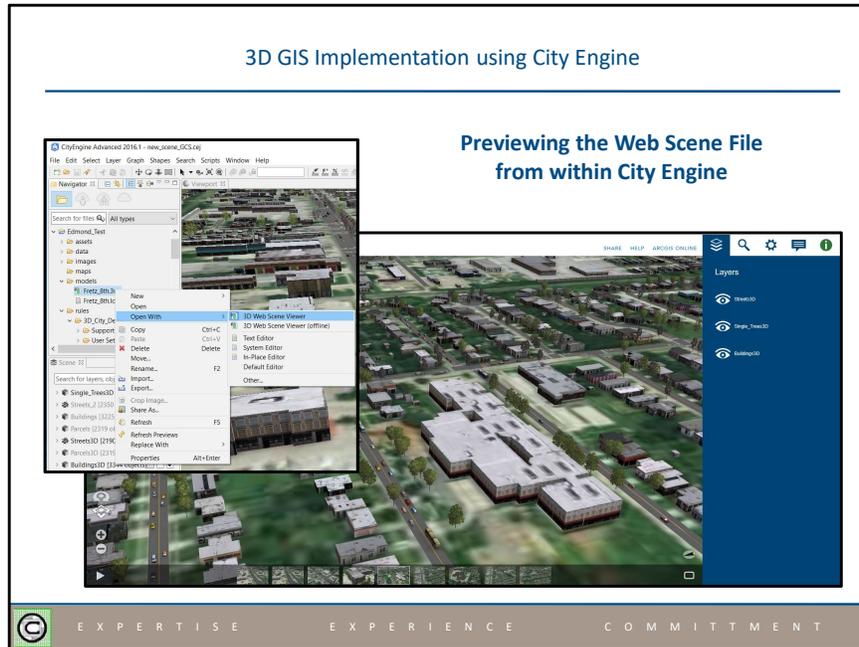
Once the model has been built there will probably be a need to make the model available for public or internal consumption via the Internet. To accomplish this, a City Engine Web Scene must first be created. This is an optimized web file carrying the 3ws extension. Once the file has been created it can be uploaded to ArcGIS Online or to an internal Server for hosting.

3D GIS Implementation using City Engine



Once the user confirms that a City Engine Web Scene is to be created, there are a couple dialog boxes that are presented to the user. An important hint to remember is to disable the Simplify Terrain Meshes option. This will keep the Web Scene file small. Additionally, the Textures parameter enables the user to control the level of detail that appears in the Web Scene, while the Metadata parameter controls what attributes will be displayed when a feature is identified.

3D GIS Implementation using City Engine



Prior to uploading the City Engine Web Scene file, it is possible to preview the Web Scene by navigating to the file from within City Engine and right-clicking on the file and opening the file using the 3D Web Scene Viewer. Note at the bottom of the Web Scene image, thumbnails of the bookmarks in the City Engine project will appear. To publish the Web Scene to ArcGIS Online, the Share As... menu item in the pop-up menu list can be selected.

**End of
Formal Presentation**

**We'd be pleased to
Respond to any Questions or Comments**



At the beginning of the presentation we had four questions we wanted to answer. We've addressed two of the four. That is how do we build the model and the level of difficulty. The remaining two, how long and cost really depends upon the level of detail that is desired in the model. The other factor to consider is the maintenance of the model. Like any other database there will be a maintenance effort to ensure the 3D GIS is current. As a rough estimate we would say that, for a two to six square mile area, a simple model can be generated in one to two weeks. A moderately realistic model is probably in the 1 to 3 month neighborhood, and a highly realistic model will be considerably longer. This assumes that the street centerline data will be used as is. If the centerline data has to be "cleaned up", the amount of time goes up.



The CEDRA Corporation

Implementers of ArcGIS, ArcGIS Online and City Engine Technology

1600 Moseley Road – Suite 500

Victor, NY 14564

Ph. 585.414.6541

Fx. 585.924.8614

Visit us at: www.cedra.com