Using FME to create 3D Tiles for Cesium from OpenStreetMap data

Overview

This tutorial explains how to create 3D Tiles for the Cesium virtual globe from OpenStreetMap data and show them on the globe. For data processing and tiling, FME is used which is one of the few softwares which can export 3D Tiles. In our example, we extrude buildings footprints. Once imported to Cesium, the buildings are colored by height.

Contents

Preconditions

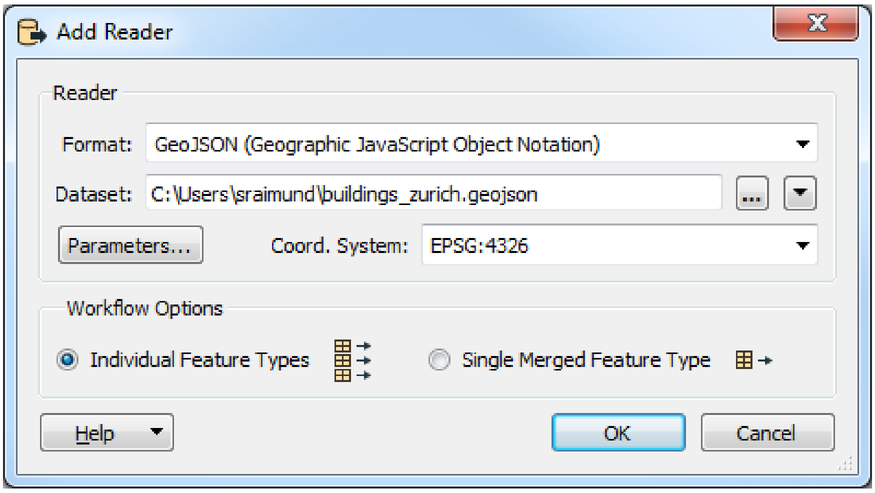
You should have obtained and stored OpenStreetMap data in a GIS format by following the OSM QGIS tutorial (http://osgl.ethz.ch/training/Tutorial_OSM_QGIS.pdf) or the JOSM tutorial (http://osgl.ethz.ch/training/Tutorial_JOSM.pdf). When you want to skip these tutorials, you can use the file “buildings_zurich.geojson” provided in the data package for this tutorial. It contains building footprints of the city of Zurich from OpenStreetMap.

Tools

- ✓ FME 2017.1 or later (https://www.safe.com/): Extract-transform-load GIS software, available at many universities or as a 30 days test version
- ✓ Cesium (https://cesiumjs.org/downloads/): Virtual globe engine based on JavaScript WebGL, free for non-commercial and commercial use
- ✓ Python (https://www.python.org/): A programming language which offers an easy way to create an HTTP file server

A) Exporting 3D Tiles with FME

1. Start FME Workbench.
2. Click on the “Add Reader” icon in the tool bar.
3. Click on the “…” button and select the GeoJSON containing the OSM data.
4. Select “EPSG:4326” in the drop down menu as a coordinate system.
5. Press the “OK” button:



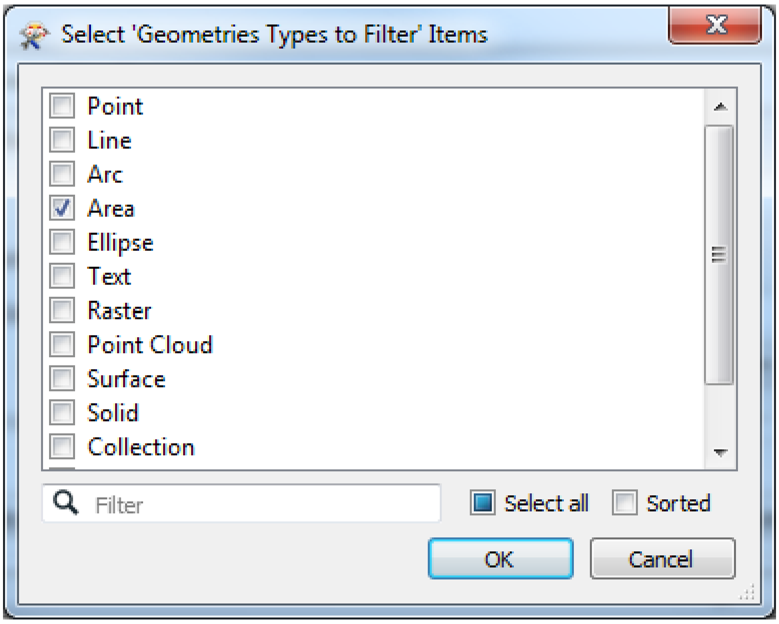
Format: GeoJSON (Geographic JavaScript Object Notation)

Dataset: C:\Users\sraimund\buildings_zurich.geojson

Coord. System: EPSG:4326

● Individual Feature Types ○ Single Merged Feature Type

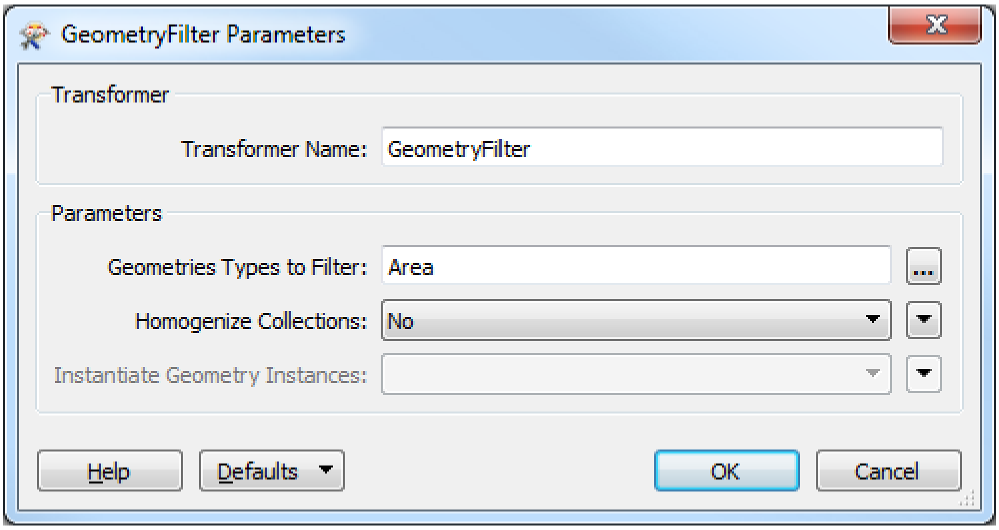
6. Click on the “Add Transformer” icon in the tool bar and select the “GeometryFilter” from the list. With this transformer, we would like to filter all geometry types except polygons.
7. Click on the settings icon of the “GeometryFilter” transformer.
8. Click on the “…” button for the “Geometry Types to Filter” and tick only “Area” in the appearing dialog:



☐ Point
☐ Line
☐ Arc
☑ Area
☐ Ellipse
☐ Text
☐ Raster
☐ Point Cloud
☐ Surface
☐ Solid
☐ Collection

■ Select all ☐ Sorted

9. Press “OK” in the “GeometryFilter” transformer to apply the changes:



GeometryFilter Parameters

Transformer

Transformer Name: GeometryFilter

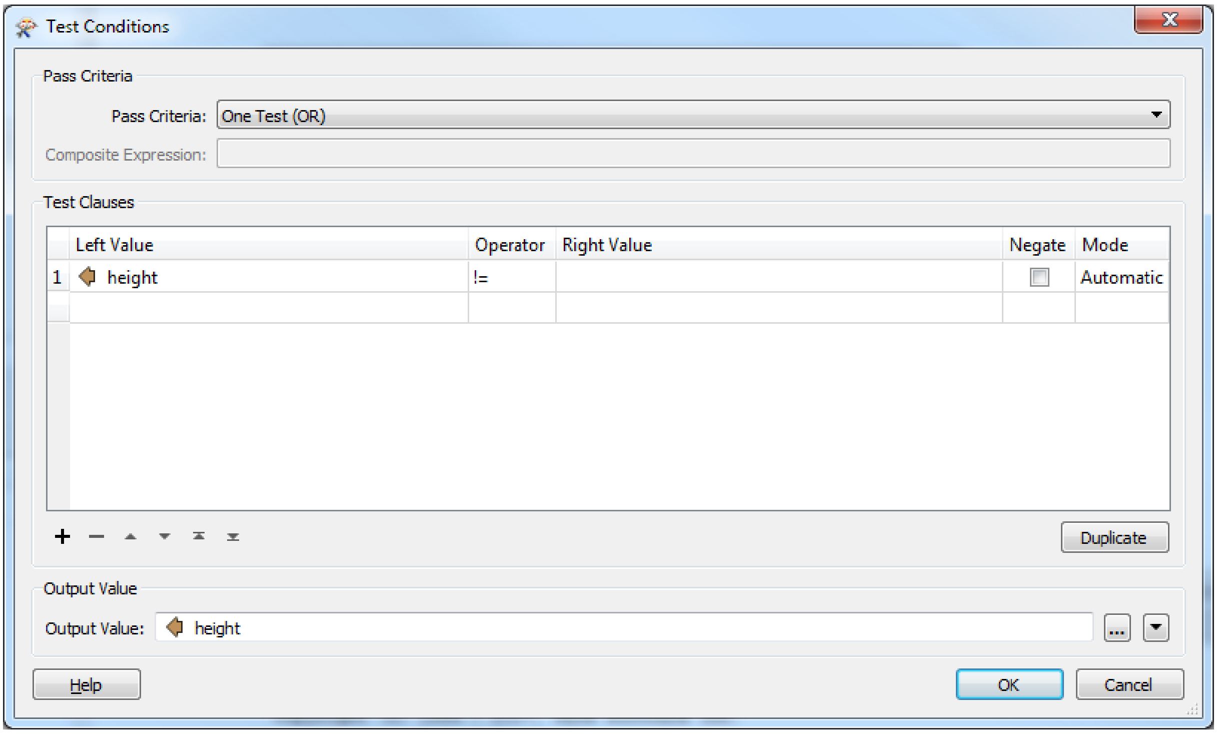
Parameters

Geometries Types to Filter: Area

Homogenize Collections: No

Instantiate Geometry Instances:

10. Click on the “Add Transformer” icon in the tool bar and select the “AttributeCreator” from the list. With this transformer, we would like to merge two attribute columns representing the building height.
11. Click on the settings icon of the “AttributeCreator” transformer.
12. Enter “building_height” as a “New Attribute” and select “Conditional Value…” for the “Attribute Value” using the arrow icon.
13. For the “Conditional value…”, click on the cell for “If” and “Test Condition”.
14. This “Test Condition” uses the attribute value “height” as “Left Value”, the “!=” operator and an empty “Right Value” (do not enter anything). Press “OK” to use this condition to test against null values:



Test Conditions

Pass Criteria

Pass Criteria: One Test (OR)

Composite Expression:

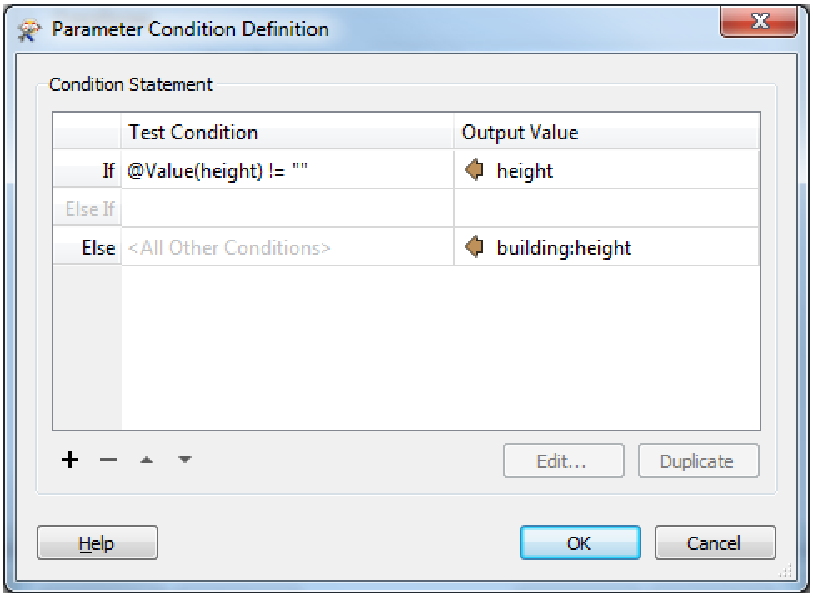
Test Clauses

	Left Value	Operator	Right Value	Negate	Mode
1	height	!=		□	Automatic

Output Value

Output Value: height

15. Select the “height” attribute as “Output Value” when the “If” condition is met and the “building:height” attribute as “Output Value” for “Else”, and press the “OK” button:

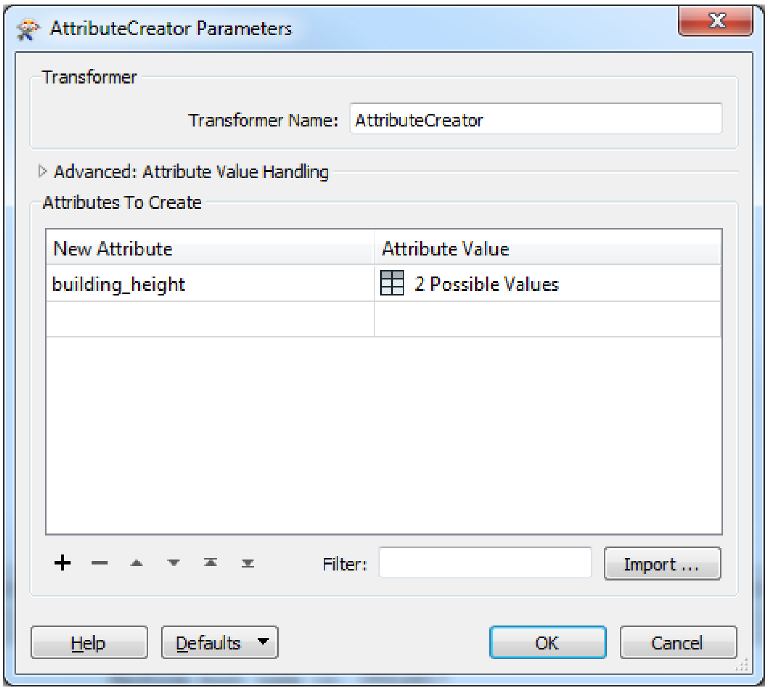


Parameter Condition Definition

Condition Statement

	Test Condition	Output Value
If	@Value(height) != ""	height
Else If		
Else	<All Other Conditions>	building:height

16. Press the “OK” button in the “AttributeCreator” transformer to apply the changes:



AttributeCreator Parameters

Transformer

Transformer Name: AttributeCreator

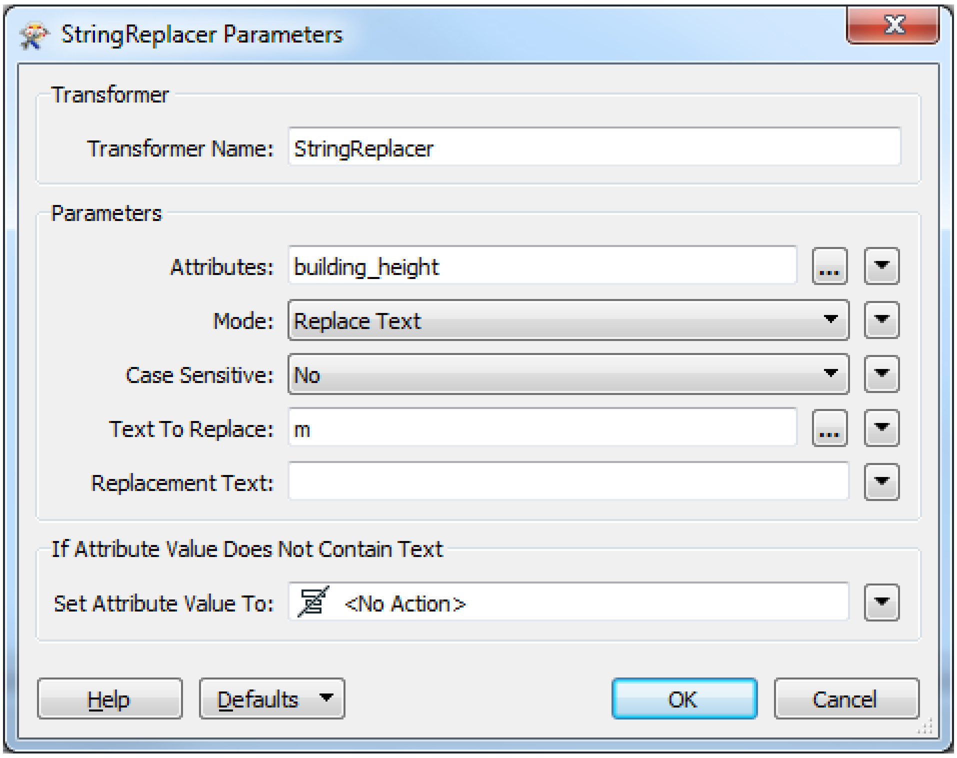
Advanced: Attribute Value Handling

Attributes To Create

New Attribute	Attribute Value
building_height	2 Possible Values

17. Click on the “Add Transformer” icon in the tool bar and select the “StringReplacer” from the list. With this transformer, we would like to replace the unit “m” (meters) in the height column.
18. Click on the settings icon of the “StringReplacer” transformer.
19. Select “building_height” for “Attributes” using the arrow icon and type in “m” for “Text to Replace”.

20. Press “OK” in the “StringReplacer” transformer to apply the changes:



StringReplacer Parameters

Transformer

Transformer Name: StringReplacer

Parameters

Attributes: building_height

Mode: Replace Text

Case Sensitive: No

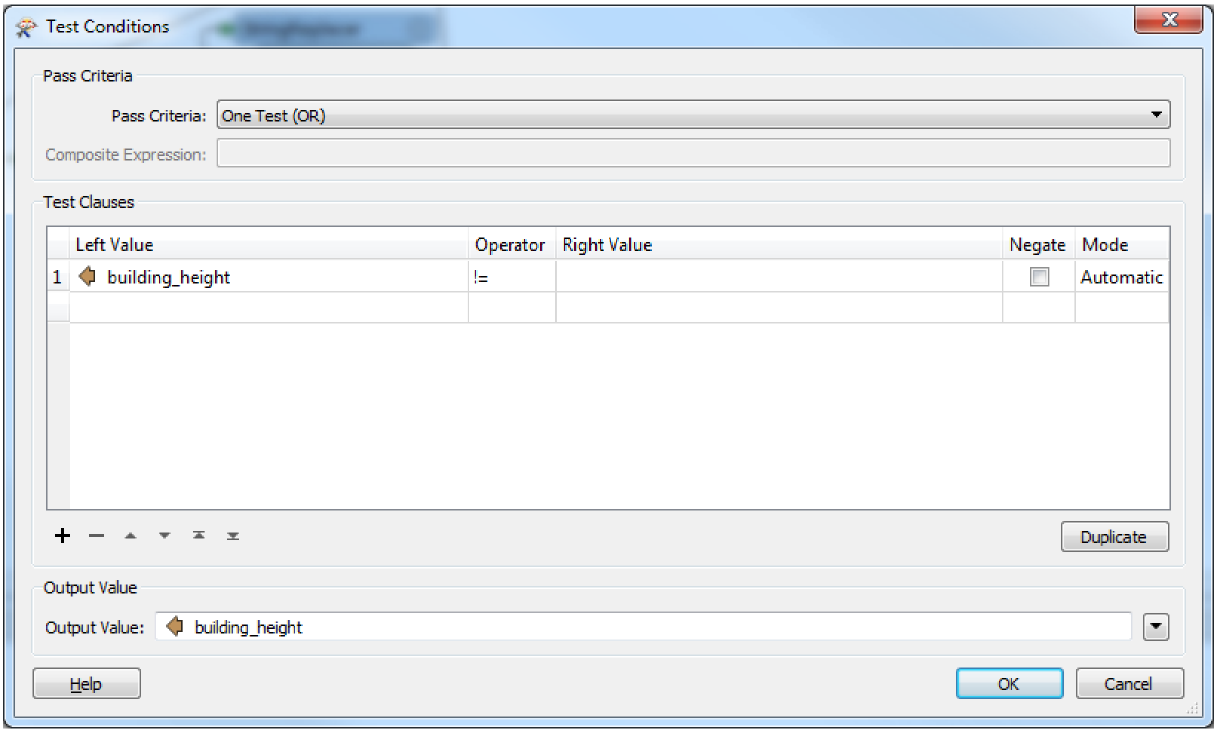
Text To Replace: m

Replacement Text:

If Attribute Value Does Not Contain Text

Set Attribute Value To: <No Action>

21. Click on the “Add Transformer” icon in the tool bar and select the “Extruder” from the list. With this transformer, we would like to extrude polygons based on the height attribute.
22. Click on the settings icon of the “Extruder” transformer.
23. Choose “Vertical” as “Direction” and enter a “Conditional value…” for the “Distance.
24. For the “Conditional value…”, click on the cell for “If” and “Test Condition”.
25. This “Test Condition” uses the attribute value “building_height” as “Left Value”, the “!=” operator and an empty “Right Value” (do not enter anything). Press “OK” to use this condition to test against null values:



Test Conditions

Pass Criteria

Pass Criteria: One Test (OR)

Composite Expression:

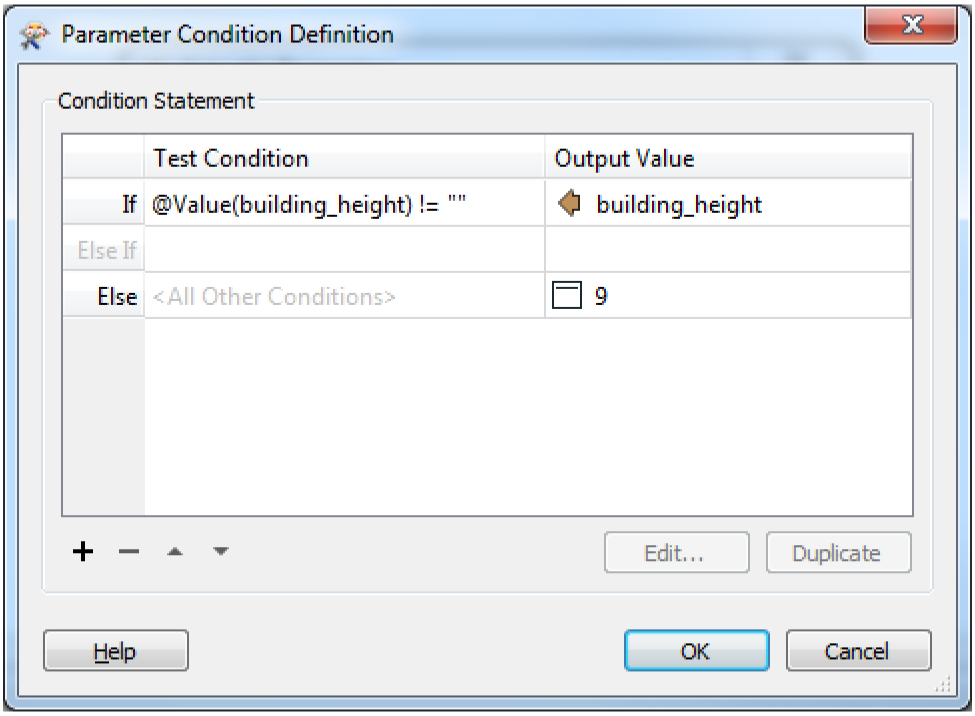
Test Clauses

	Left Value	Operator	Right Value	Negate	Mode
1	building_height	!=		□	Automatic

Output Value

Output Value: building_height

26. Select the “building_height” attribute as “Output Value” when the “If” condition is met, enter “9” as “Output Value” for “Else” and press the “OK” button:

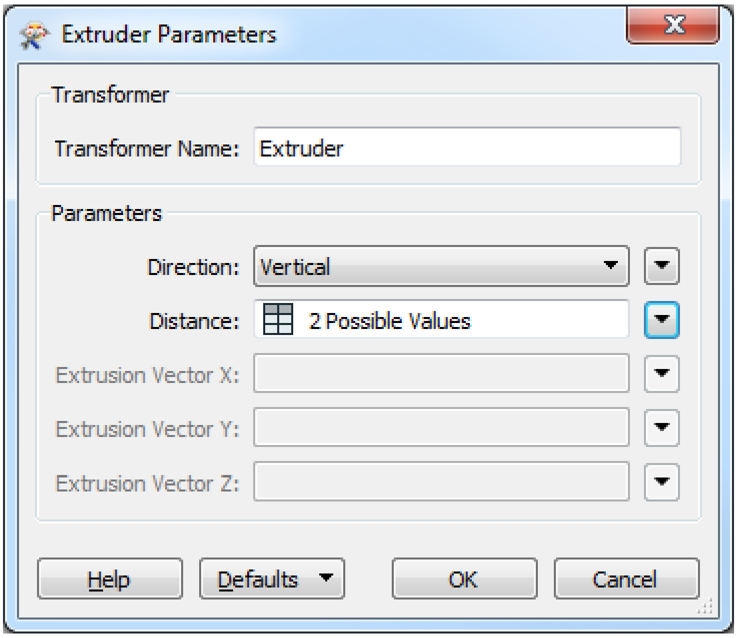


Parameter Condition Definition

Condition Statement

	Test Condition	Output Value
If	@Value(building_height) != ""	building_height
Else If		
Else	<All Other Conditions>	9

27. Press “OK” in the “Extruder” transformer to apply the changes:



Extruder Parameters

Transformer

Transformer Name: Extruder

Parameters

Direction: Vertical

Distance: 2 Possible Values

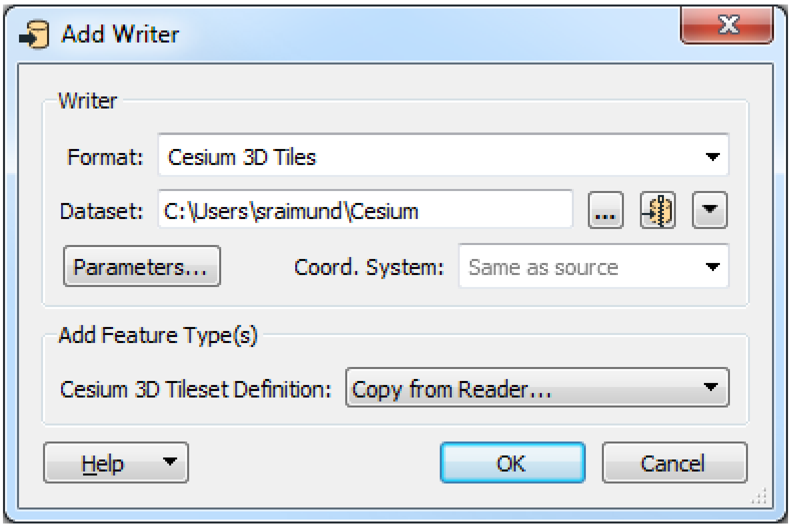
Extrusion Vector X:

Extrusion Vector Y:

Extrusion Vector Z:

28. Click on the “Add Writer” icon in the tool bar. With the writer, we would like to export Cesium 3D Tiles.
29. Select “Cesium 3D Tiles” as “Format” and click on the “…” button to select a folder to output the 3D tileset.

30. Press on the “OK” button to apply the changes:



Add Writer

Writer

Format: Cesium 3D Tiles

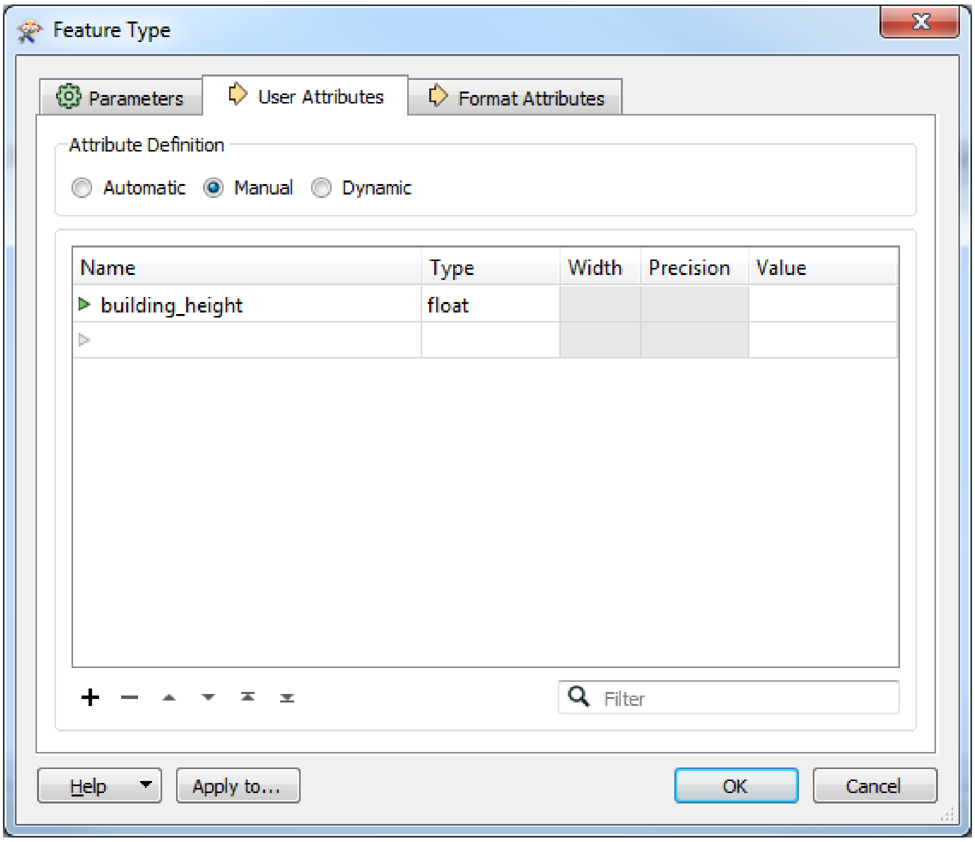
Dataset: C:\Users\sraimund\Cesium

Coord. System: Same as source

Add Feature Type(s)

Cesium 3D Tileset Definition: Copy from Reader...

31. Click on the settings icon of the “Cesium 3D Tiles” writer and navigate to the tab “User Attributes”.
32. Click on the “Manual” radio button for “Attribute Definition” and select all feature attributes you would like to use in Cesium later on as query or styling attribute. In this example, we leave only the “building_height” attribute and cast it to the “Type” “float”
33. Press on the “OK” button to apply the changes:



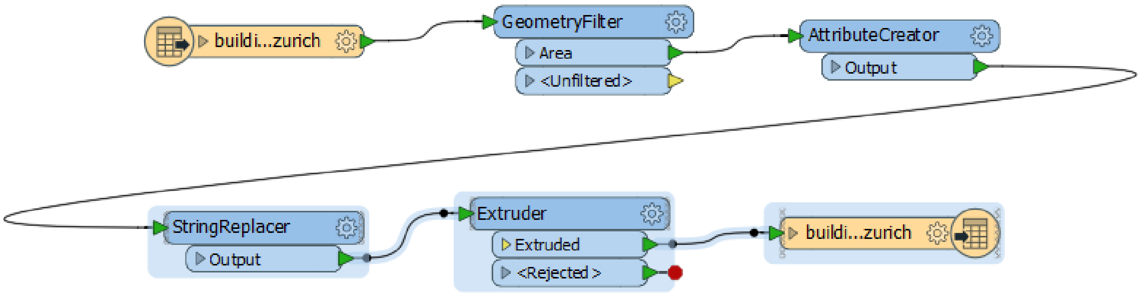
Feature Type

Attribute Definition

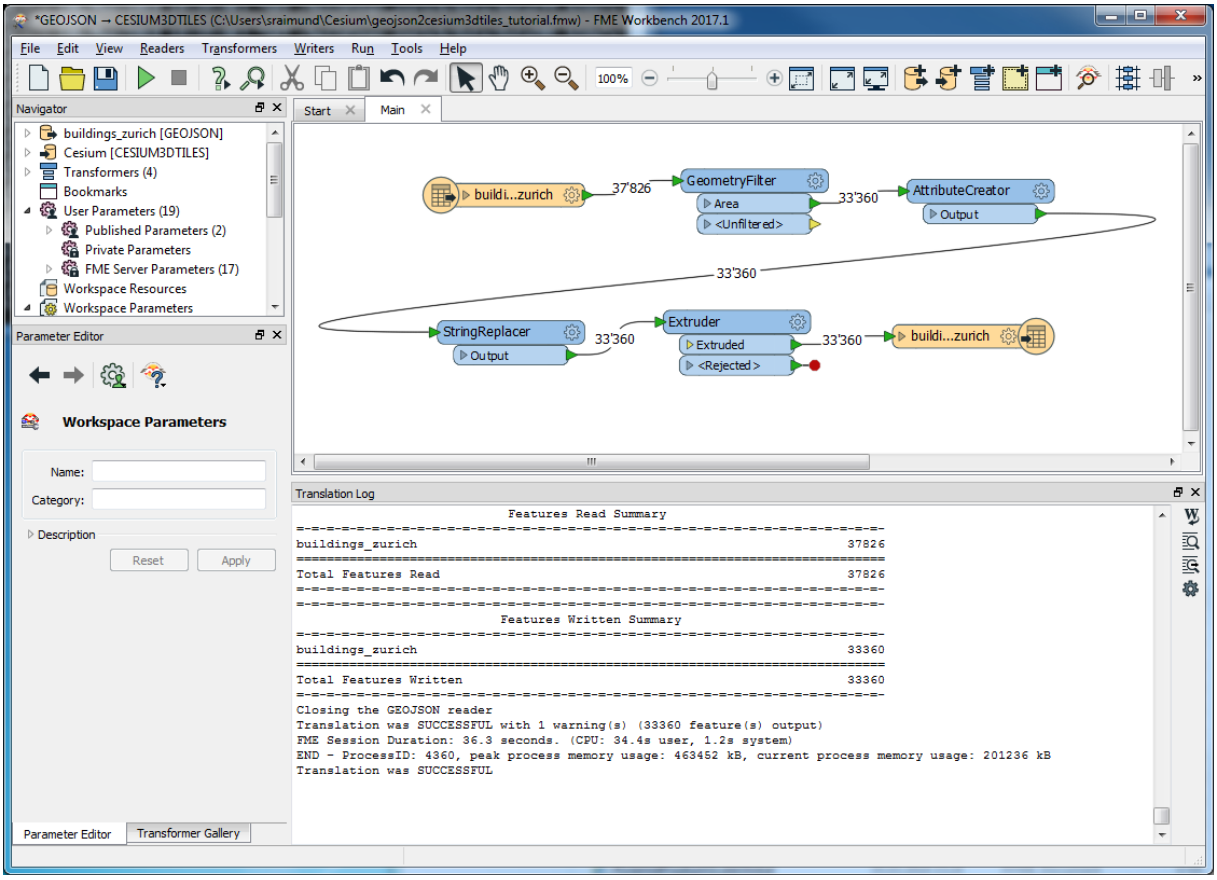
○ Automatic ● Manual ○ Dynamic

Name	Type	Width	Precision	Value
building_height	float			

34. Connect the “Extruder” transformer with the “Cesium 3D Tiles” writer in the canvas. Your processing pipeline should look like this:



35. Press on the “Run Translation” icon in the tool bar to start the conversion. If everything works fine, then there are not any errors in the “Translation Log”:

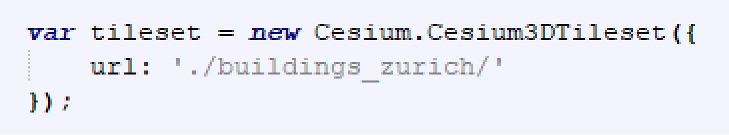


```
                           Features Read Summary
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
buildings_zurich                                                          37826
===============================================================================
Total Features Read                                                       37826
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
                          Features Written Summary
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
buildings_zurich                                                          33360
===============================================================================
Total Features Written                                                    33360
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Closing the GEOJSON reader
Translation was SUCCESSFUL with 1 warning(s) (33360 feature(s) output)
FME Session Duration: 36.3 seconds. (CPU: 34.4s user, 1.2s system)
END - ProcessID: 4360, peak process memory usage: 463452 kB, current process memory usage: 201236 kB
Translation was SUCCESSFUL
```

36. Note: The configuration file “geojson2cesium3dtiles.fmw” of this project is included in the materials package for this tutorial.

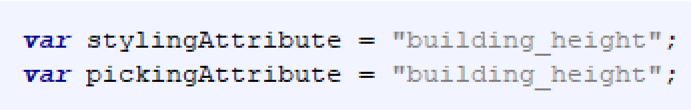
B) Visualizing 3D Tiles in Cesium

1. Create a new folder.
2. Download and extract the zip folder with the current Cesium release
3. Copy the “Cesium” folder from the “Cesium-[version]” > “Build” folder to your newly created folder. Alternatively copy the “CesiumUnminified” folder and rename it to “Cesium” when you want to get better debug messages while testing your code.
4. Copy the 3D tiles output folder from part A) into your newly created folder.
5. Copy the “OSMBuildingsExtruded.html” document from the materials or the Appendix into your newly created folder.
6. Change the URL (here: “buildings_zurich”) of the Cesium3DTileset in “OSMBuildingsExtruded.html” so that it points to the copied 3D tiles folder:



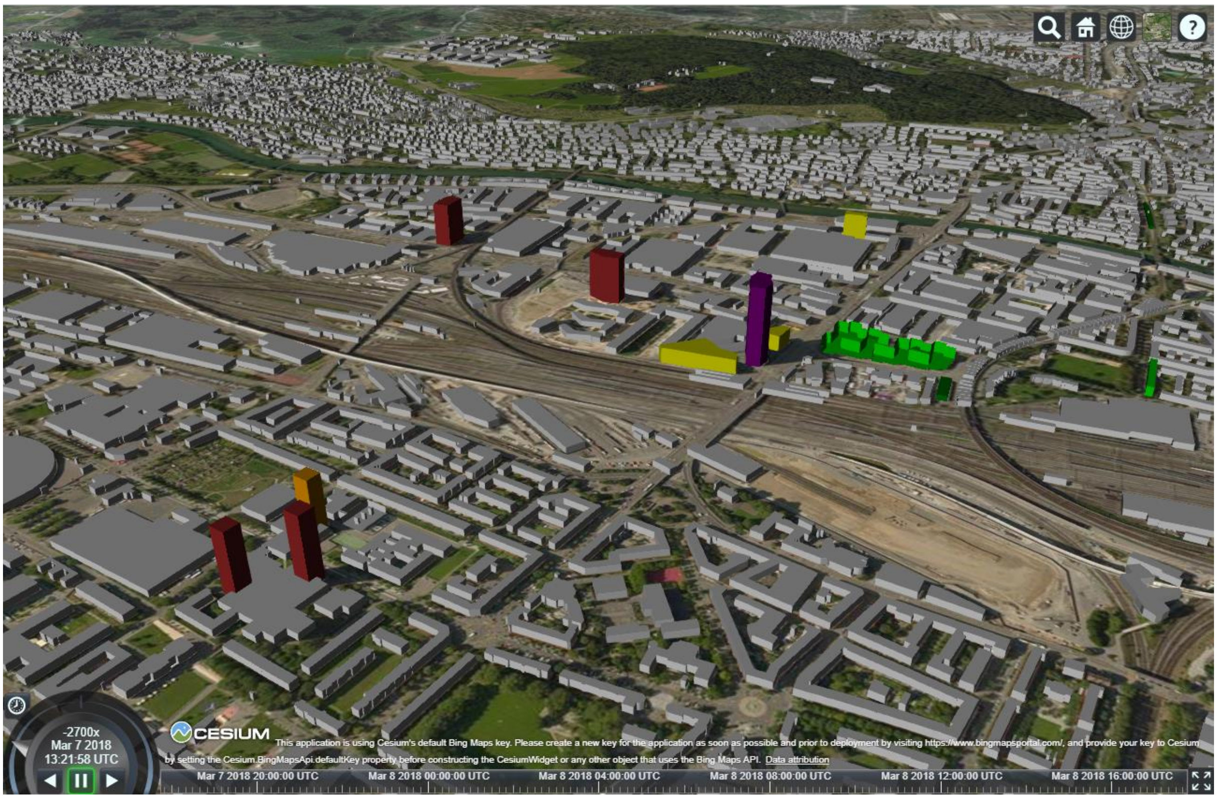
```
var tileset = new Cesium.Cesium3DTileset({
    url: './buildings_zurich/'
});
```

7. Change all attribute names for styling and picking (here: “building_height”) in “OSMBuildingsExtruded.html” to your exported attributes:



```
var stylingAttribute = "building_height";
var pickingAttribute = "building_height";
```

8. Open the Windows command line (cmd.exe), navigate to your newly created folder and start a local file server with the command “python -m SimpleHTTPServer”. This command works only when the folder where python.exe is located (e.g. C:\Python27) has been added to your PATH environment variable. Eventually you have to set PYTHONHOME pointing to the same folder (e.g. C:\Python27) and PYTHONPATH pointing to python packages (e.g. C:\Python27\Lib\site-packages;C:\Python27\DLLs). You can use the “set” command to inspect all environment variables and change them (set variable=value) as long the command line window is open.
9. In your web browser open http://localhost:8000/OSMBuildingsExtruded.html to see your 3D features on the Cesium globe:



This application is using Cesium's default Bing Maps key. Please create a new key for the application as soon as possible and prior to deployment by visiting https://www.bingmapsportal.com/, and provide your key to Cesium by setting the Cesium.BingMapsApi.defaultKey property before constructing the CesiumWidget or any other object that uses the Bing Maps API. Data attribution

10. Click with the left mouse button on a feature to inspect its height.

Raimund Schnürer, 12.03.2018

Appendix

Complete code of “OSMBuildingsExtruded.html”:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no">
    <title>Extruded OSM Buildings as 3D Tiles</title>
    <script src="./Cesium/Cesium.js"></script>
    <style>
      @import url(./Cesium/Widgets/widgets.css);
      html, body, #cesiumContainer {
          width: 100%; height: 100%; margin: 0; padding: 0; overflow:
hidden;
      }
    </style>
  </head>
  <body>
    <div id="cesiumContainer"></div>
    <script>
      //Code adapted from Cesium Sandcastle example "3D Tiles format"
      var viewer = new Cesium.Viewer('cesiumContainer');
      var scene = viewer.scene;

      var tileset = new Cesium.Cesium3DTileset({
            url: 'data/osm_buildings_extruded/'
      });

      var stylingAttribute = "building_height";
      var pickingAttribute = "building_height";

      scene.primitives.add(tileset);

      tileset.readyPromise.then(function (tileset) {
            //zoom to 3D tiles
            viewer.zoomTo(tileset, new Cesium.HeadingPitchRange(0.0, -0.5,
tileset.boundingSphere.radius));

            //color based on building height
            var properties = tileset.properties;
            if (Cesium.defined(properties) &&
Cesium.defined(properties[stylingAttribute])) {
                  tileset.style = new Cesium.Cesium3DTileStyle({
                              color: {
                                    conditions: [
                                          ["${" + stylingAttribute + "} ===
null", "color('white')"],
                                          ["${" + stylingAttribute + "} >=
120", "color('purple')"],
                                          ["${" + stylingAttribute + "} >=
80", "color('brown')"],
                                          ["${" + stylingAttribute + "} >=
50", "color('orange')"],
                                          ["${" + stylingAttribute + "} >=
20", "color('yellow')"],
                                          ["${" + stylingAttribute + "} >=
10", "color('lime')"],
                                          ["${" + stylingAttribute + "} >= 
0", "color('green')"],
                                          ["true", "color('red')"]
                                    ]
                              }
                        });
            }
      }).otherwise(function (error) {
            throw (error);
      });

      //Picking with left mouse button
      var handler = new Cesium.ScreenSpaceEventHandler(viewer.canvas);

      handler.setInputAction(function (movement) {
            var feature = viewer.scene.pick(movement.position);
            if (!Cesium.defined(feature)) {
                  return;
            }
            var building_height = feature.getProperty(pickingAttribute);

            if (building_height)
                  alert("Height: " + building_height + "m");
            else
                  alert("Height: n/a");

      }, Cesium.ScreenSpaceEventType.LEFT_CLICK);
    </script>
  </body>
</html>
```