FME® Desktop KML Pathway Tutorial

FME Desktop 2012 Edition

SAFE SOFTWARE

Safe Software Inc. makes no warranty either expressed or implied, including, but not limited to, any implied warranties of merchantability or fitness for a particular purpose regarding these materials, and makes such materials available solely on an “as-is” basis.

In no event shall Safe Software Inc. be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising out of purchase or use of these materials. The sole and exclusive liability of Safe Software Inc., regardless of the form or action, shall not exceed the purchase price of the materials described herein.

This manual describes the functionality and use of the software at the time of publication. The software described herein, and the descriptions themselves, are subject to change without notice.

Copyright

Revisions

Every effort has been made to ensure the accuracy of this document. Safe Software Inc. regrets any errors and omissions that may occur and would appreciate being informed of any errors found. Safe Software Inc. will correct any such errors and omissions in a subsequent version, as feasible. Please contact us at:

Safe Software Inc.
Suite 2017, 7445 – 132nd Street
Surrey, BC
Canada
V3W1J8

www.safe.com

Safe Software Inc. assumes no responsibility for any errors in this document or their consequences, and reserves the right to make improvements and changes to this document without notice.

Trademarks

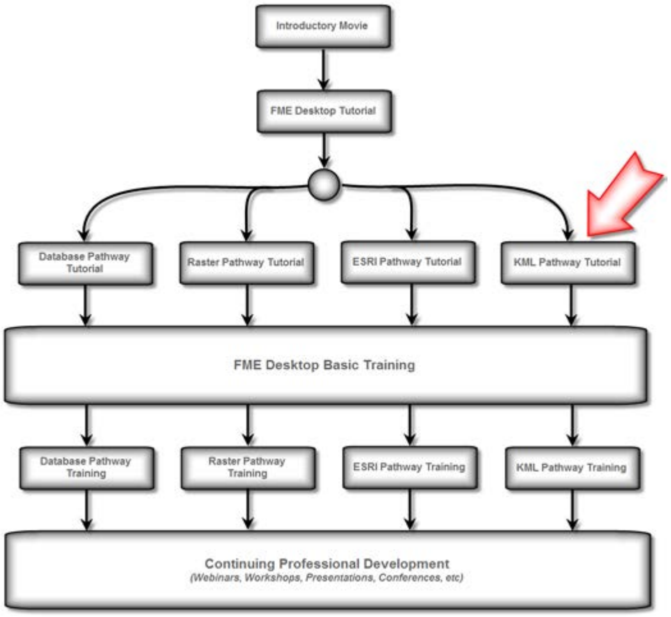
Documentation Information

Document Name: FME Desktop KML Pathway Tutorial
Version: FME 2012
Updated: October 2011

Using KML

Introduction

Welcome to the FME Desktop KML Pathway Tutorial



KML Pathway

This tutorial is an introduction to using KML data with FME. It is the first part of the FME Training KML Pathway.

It is assumed that you will already be familiar with the concepts and techniques described in the FME Desktop Tutorial.

NB: *You can find the FME Desktop Tutorial either online or on the FME Desktop installation DVD. It includes both PDF documents and a set of movies which cover each chapter of the tutorial.*

FME Version

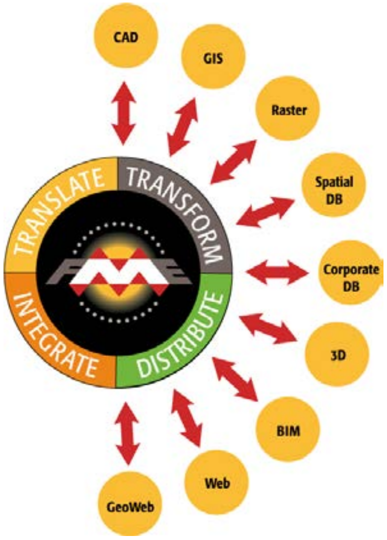
This tutorial specifically covers the use of FME Desktop® 2012 edition. Older versions of FME may not have some of the functionality described in this tutorial.

Sample Data

The sample data required to carry out the examples in this document can be obtained from:
http://www.safe.com/fmedata

Introduction to FME

Here's a quick one-page reminder on what FME is

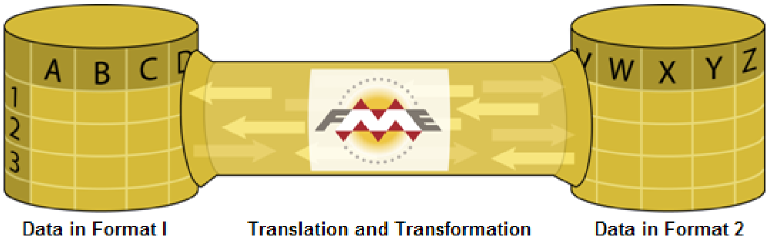


What is FME?
FME is a spatial data transformation platform that helps organizations more easily overcome a range of spatial data interoperability challenges. It is available in both desktop and server solutions.

FME is classified as a Spatial ETL (Extract-Transform-Load) tool, designed to help users master more spatial data transformation challenges than any other technology.

- *Extract* is the ability to read any format of spatial data.
- *Transform* is the ability to manipulate data during the translation process.
- *Load* is the ability to write the data in any other format.

With Data Transformation, the output from an FME process can be tailored to match a required structure, and can even be greater than the sum of the inputs.



The key FME Desktop application is **FME Workbench**, an intuitive point and click interface for graphically defining translations and transformations as a flow of data.

FME Quick Translator is an application for carrying out basic, non-customized translations.

FME Universal Viewer is an application for visually inspecting spatial data.

KML Basics

If you are new to KML data, then here are some basic facts and information

What is KML?

KML stands for Keyhole Markup Language. It is an XML-based format (or language if you prefer) intended to store data for use within the Google Earth™ and Google Maps™ applications.

The name “Keyhole” comes from the name of the original developers of the KML format and Google Earth product.

KMZ is an alternate form for a KML format dataset. A KMZ dataset is simply a KML dataset compressed by a ZIP type program and renamed with a new file extension. KMZ is most frequently used as a means to store a set of raster images; the KMZ (zip) folder stores the raster files (as JPEG or GeoTIFF) plus a KML file that references them.

What do KML Datasets Look Like?

A KML dataset looks similar to an XML or HTML document. In fact the analogy that Google uses is that Google Earth is to KML data what Internet Explorer is to an HTML document: simply a browser that permits a user to visualize the content of the dataset.

Google Maps is also a KML browser, but – at the time of writing – only supports a subset of KML. There are other KML viewers, but for this module we will stick to using Google Earth.

Like HTML, KML has *tags* that affect how specific features are displayed. Since KML stores data of a spatial nature – whereas HTML tends to be non-spatial information – the tags are those that relate to spatial data symbology; for example, line styles, point symbols and area fill colors.

After cleaning up the data (by removing various style tags), a sample KML dataset looks like this:

```
<Placemark id="kml_1">
      <name>Safe Software HQ</name>
      <description><![CDATA[Safe Software's HQ]]></description>
      <Style>
            <IconStyle>
                  <scale>0.8</scale>
                  <Icon>
                        <href>images/C2.png</href>
                  </Icon>
            </IconStyle>
      </Style>
      <Point>
            <coordinates>-132,45,0 </coordinates>
      </Point>
</Placemark>
```

KML and FME

As with any FME supported format it is important to be aware of how the format's structure relates to FME, and how FME defines that structure as a schema.

Feature Types and Datasets

KML is a file-based dataset, meaning each KML file is counted as an FME dataset. Each file can contain "Folder id" sections representing FME feature types.

For example, if the FME dataset name is *planning* and the feature type name is ROADS, then the output would be a KML file called *planning.kml* containing a section beginning:

```
<Folder id="kml_ft_ROADS">
```

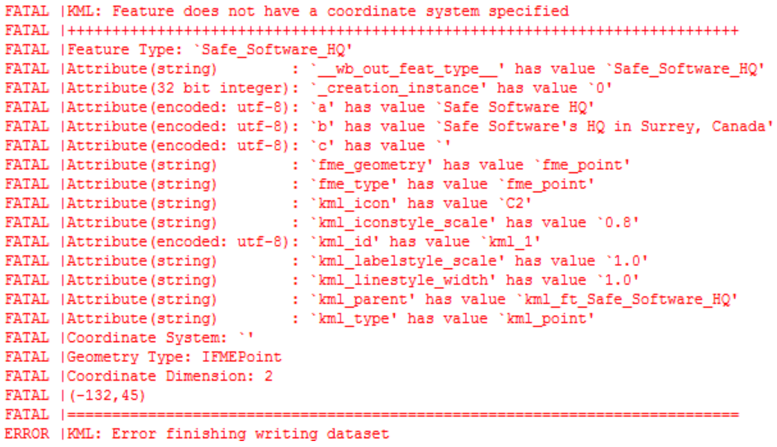
KML and Coordinate Systems

KML stores coordinates as latitude and longitude values based on the WGS84 datum. This is the only coordinate system that KML supports.

The FME equivalent coordinate systems are ***LL84*** and ***EPSG:4326***

Data sent to the KML writer must be tagged with a coordinate system – either from within the source data or using a *CoordinateSystemSetter* transformer. Then FME will automatically convert the data to LL84.

When the data is untagged, and FME is unable to ascertain the source coordinate system, the translation will be terminated.



```
FATAL |KML: Feature does not have a coordinate system specified
FATAL |+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
FATAL |Feature Type: `Safe_Software_HQ'
FATAL |Attribute(string)        : `__wb_out_feat_type__' has value `Safe_Software_HQ'
FATAL |Attribute(32 bit integer): `_creation_instance' has value `0'
FATAL |Attribute(encoded: utf-8): `a' has value `Safe Software HQ'
FATAL |Attribute(encoded: utf-8): `b' has value `Safe Software's HQ in Surrey, Canada'
FATAL |Attribute(encoded: utf-8): `c' has value `'
FATAL |Attribute(string)        : `fme_geometry' has value `fme_point'
FATAL |Attribute(string)        : `fme_type' has value `fme_point'
FATAL |Attribute(string)        : `kml_icon' has value `C2'
FATAL |Attribute(string)        : `kml_iconstyle_scale' has value `0.8'
FATAL |Attribute(encoded: utf-8): `kml_id' has value `kml_1'
FATAL |Attribute(string)        : `kml_labelstyle_scale' has value `1.0'
FATAL |Attribute(string)        : `kml_linestyle_width' has value `1.0'
FATAL |Attribute(string)        : `kml_parent' has value `kml_ft_Safe_Software_HQ'
FATAL |Attribute(string)        : `kml_type' has value `kml_point'
FATAL |Coordinate System: `'
FATAL |Geometry Type: IFMEPoint
FATAL |Coordinate Dimension: 2
FATAL |(-132,45)
FATAL |===========================================================================
ERROR |KML: Error finishing writing dataset
```

Reading KML Data

FME's support for KML includes the ability to read KML datasets and to convert them into other GIS data formats

FME's ability to read KML datasets includes all of the spatial components, but also items like document properties, folders, and timestamp attributes.

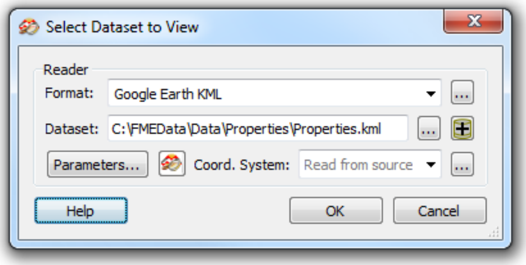
Exercise 1: Read KML Data
Follow these steps to open a KML dataset for inspection.

1. Start FME Universal Viewer. Select Start > All Programs > FME Desktop 2012 > FME Universal Viewer from the Windows Start menu.

2. Select **File** > **Open Dataset** from the menu bar to open the dataset selection dialog.

   Fill in these fields:

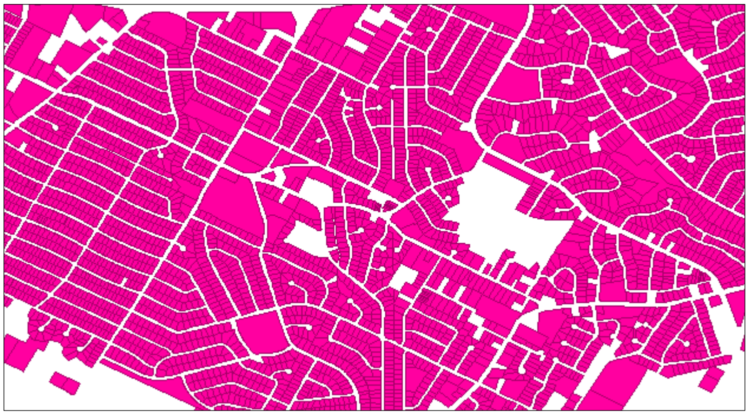
   Reader Format: Google Earth KML
   Reader Dataset: C:\FMEData\Data\Properties\Properties.kml



Select Dataset to View

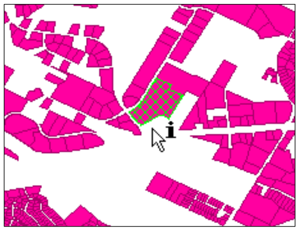
Reader
Format: Google Earth KML
Dataset: C:\FMEData\Data\Properties\Properties.kml
Coord. System: Read from source

3. Click **OK** to accept the selection and open the dataset:

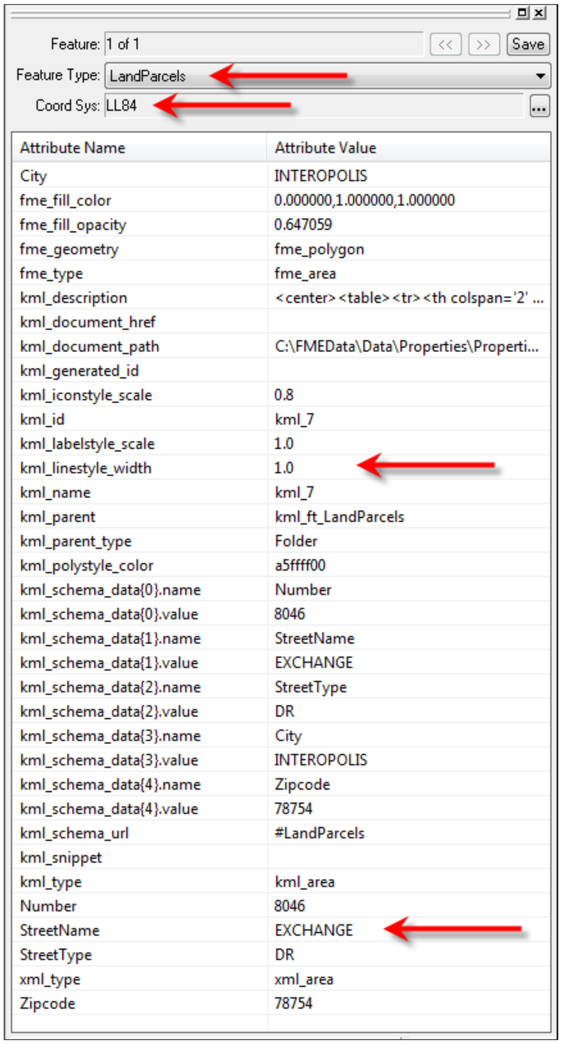


Now that there is some data open in the FME Universal Viewer, it can be queried.

4. Click the Select Features tool to make it active (denoted by a small "i" character on the cursor). Click on any parcel in the dataset to query it.



The feature is queried and the information window of the FME Universal Viewer shows the results of the query:



Feature: 1 of 1

Feature Type: LandParcels

Coord Sys: LL84

Attribute Name	Attribute Value
City	INTEROPOLIS
fme_fill_color	0.000000,1.000000,1.000000
fme_fill_opacity	0.647059
fme_geometry	fme_polygon
fme_type	fme_area
kml_description	<center><table><tr><th colspan='2' ...
kml_document_href	
kml_document_path	C:\FMEData\Data\Properties\Properti...
kml_generated_id	
kml_iconstyle_scale	0.8
kml_id	kml_7
kml_labelstyle_scale	1.0
kml_linestyle_width	1.0
kml_name	kml_7
kml_parent	kml_ft_LandParcels
kml_parent_type	Folder
kml_polystyle_color	a5ffff00
kml_schema_data{0}.name	Number
kml_schema_data{0}.value	8046
kml_schema_data{1}.name	StreetName
kml_schema_data{1}.value	EXCHANGE
kml_schema_data{2}.name	StreetType
kml_schema_data{2}.value	DR
kml_schema_data{3}.name	City
kml_schema_data{3}.value	INTEROPOLIS
kml_schema_data{4}.name	Zipcode
kml_schema_data{4}.value	78754
kml_schema_url	#LandParcels
kml_snippet	
kml_type	kml_area
Number	8046
StreetName	EXCHANGE
StreetType	DR
xml_type	xml_area
Zipcode	78754

Notice how the window shows:

- The data type/folder (LandParcels)
- The coordinate system of the data (LL84)
- KML information such as scale and style
- Attributes like Street Name and Type

Use the other tools in this part of the toolbar to measure distances, pan, and zoom.

Exercise 2: Translate KML with FME Workbench

Follow these steps to set up a conversion from KML format data to a different format. In this example the chosen output format is MapInfo TAB.

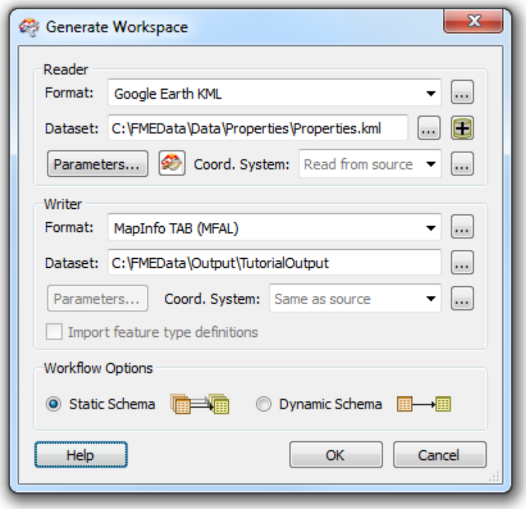
1. Start FME Workbench. In the Start tab, choose the option to **Generate workspace**.

2. When the New Workspace dialog opens, fill in these fields:

   Reader
   Format: Google Earth KML
   Dataset: C:\FMEData\Data\Properties\Properties.kml

   Writer
   Format: MapInfo TAB (MFAL)
   Dataset: C:\FMEData\Output\TutorialOutput

   Do not click **OK** yet.



Generate Workspace

Reader
Format: Google Earth KML
Dataset: C:\FMEData\Data\Properties\Properties.kml
Parameters... Coord. System: Read from source

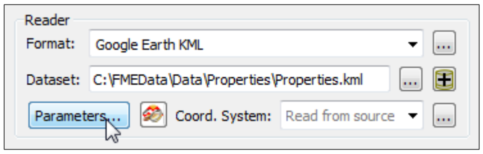
Writer
Format: MapInfo TAB (MFAL)
Dataset: C:\FMEData\Output\TutorialOutput
Parameters... Coord. System: Same as source
□ Import feature type definitions

Workflow Options
● Static Schema ○ Dynamic Schema

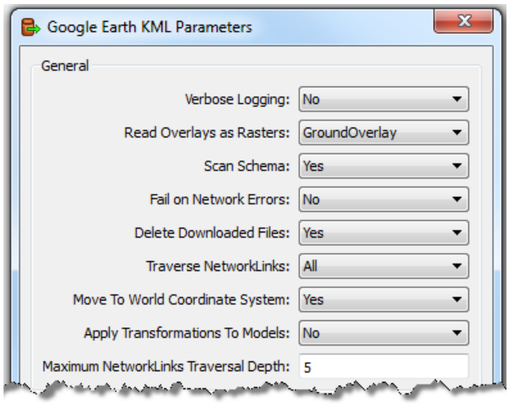
***NB:** for MapInfo TAB, the writer dataset requires a folder to be selected, not a file or filename.*

3. Click the **Parameters** button in the Reader section.

   A new dialog will open displaying all of the parameters that can be used to control reading of the KML data.



Reader
Format: Google Earth KML
Dataset: C:\FMEData\Data\Properties\Properties.kml
Parameters... Coord. System: Read from source



Google Earth KML Parameters

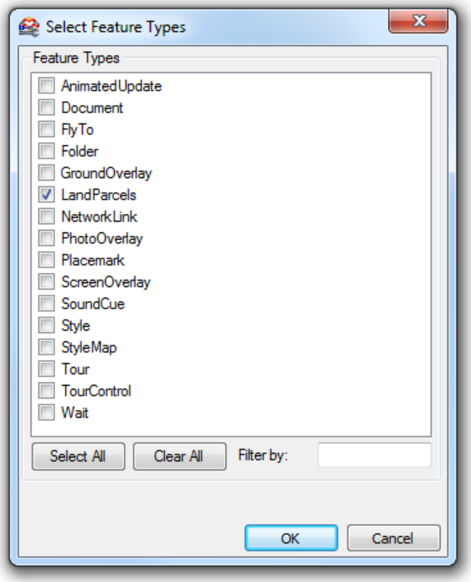
General
Verbose Logging: No
Read Overlays as Rasters: GroundOverlay
Scan Schema: Yes
Fail on Network Errors: No
Delete Downloaded Files: Yes
Traverse NetworkLinks: All
Move To World Coordinate System: Yes
Apply Transformations To Models: No
Maximum NetworkLinks Traversal Depth: 5

Notice how there are parameters to control aspects such as the reading of raster data, the traversal of network links, and the transformation of KML models (3D objects stored in an associated Collada file).

Now click **OK** to close the dialog and again to accept the selection.

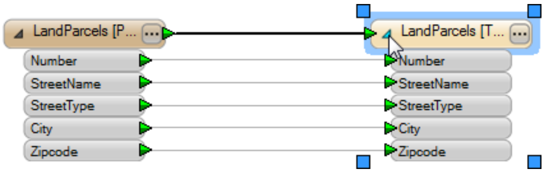
4. The Select Feature Types dialog appears.

   Press the **Clear All** button to deselect all types. Place a check mark next to the *LandParcels* feature type in order to select it for translation. Click **OK**.

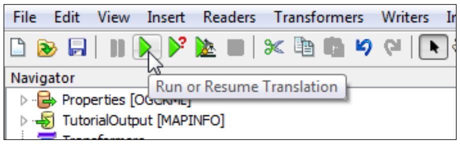


□ AnimatedUpdate
□ Document
□ FlyTo
□ Folder
□ GroundOverlay
☑ LandParcels
□ NetworkLink
□ PhotoOverlay
□ Placemark
□ ScreenOverlay
□ SoundCue
□ Style
□ StyleMap
□ Tour
□ TourControl
□ Wait

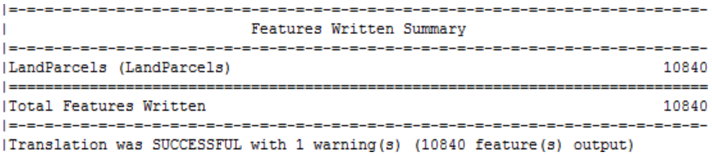
5. A workspace will now be created. Click the expand icon on each feature type object, in order to reveal the attributes being translated. The workspace will now look like this:



6. Run the translation. Click the green play button to start the translation.

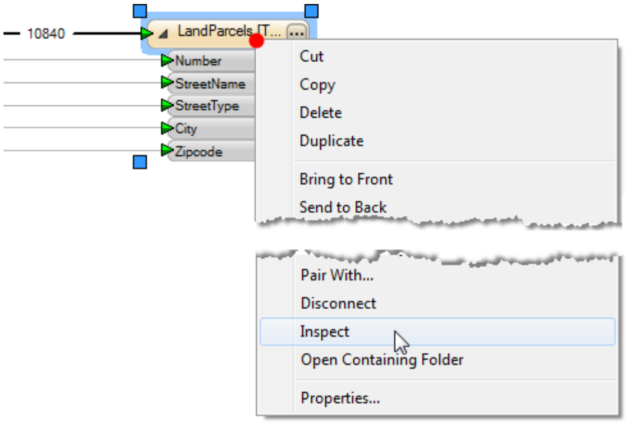


The translation will now run. It may take one or two minutes to complete. A dialog and log message may warn of “Unexpected Input”, but it is not really unexpected, since we turned off several feature types in step 4. Therefore the warning can be ignored.

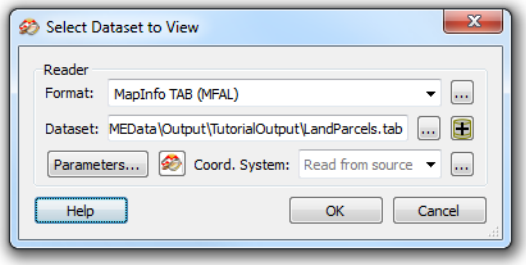


```
|=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
|                            Features Written Summary
|=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
|LandParcels (LandParcels)                                                  10840
|==================================================================================
|Total Features Written                                                     10840
|=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
|Translation was SUCCESSFUL with 1 warning(s) (10840 feature(s) output)
```

7. To inspect the output, right-click on the writer feature type and choose the option **Inspect**.



The FME Universal Viewer will start up, and a pre-filled dialog will be displayed:



Format: MapInfo TAB (MFAL)
Dataset: MEData\Output\TutorialOutput\LandParcels.tab
Coord. System: Read from source

Click **OK**. The newly created MapInfo dataset will be opened and can be inspected to prove the translation functioned correctly.

Writing KML Data

FME allows users to translate other spatial datasets into the KML format widely used in today's society

Writing a basic KML dataset, with no concern about complex node types or feature styling, is as simple as choosing KML as the output format and running the translation. Where there is styling (symbology) present on the source data, FME will attempt to preserve it when writing KML output.

KML Writing Requirements

For the most part if there are any peculiarities about the KML format, FME takes care of them automatically.

- KML requires all features to be three-dimensional; if necessary FME will force compliance to this rule by setting a Z value (third dimension) of zero on all two-dimensional features.
- All nodes must have a unique ID. By default FME uses the format attribute kml_id, but if this is unset then FME will automatically create an id number in order to comply with this rule.
- As noted, KML requires all features to be held in the LL84 coordinate system. FME will automatically convert your data to LL84 provided that it knows the source coordinate system used. If it cannot deduce this information, and you do not provide it in the dataset parameters, then the translation will be stopped with an error.

KML or KMZ?

FME's KML writer provides the capability to write the output data as either a KML or a KMZ dataset. The type of dataset created depends upon the file extension you provide within the output dataset name; for example, name your output myData.kml to create an uncompressed dataset, or myData.kmz to create it in compressed form.

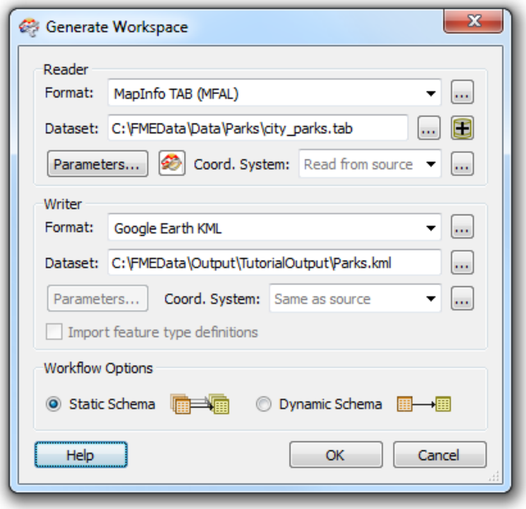
Exercise 3: Translate to KML with FME Workbench
Follow these steps to set up a conversion of GIS data to KML format. In this example the source data format is again MapInfo TAB.

1. Start FME Workbench. In the Start tab choose the option to **Generate workspace**.

2. When the New Workspace dialog opens, fill in these fields:

   Reader
   Format: MapInfo TAB (MFAL)
   Dataset: C:\FMEData\Data\Parks\city_parks.tab

   Writer
   Format: Google Earth KML
   Dataset: C:\FMEData\Output\TutorialOutput\Parks.kml



Generate Workspace

Reader
Format: MapInfo TAB (MFAL)
Dataset: C:\FMEData\Data\Parks\city_parks.tab
Coord. System: Read from source

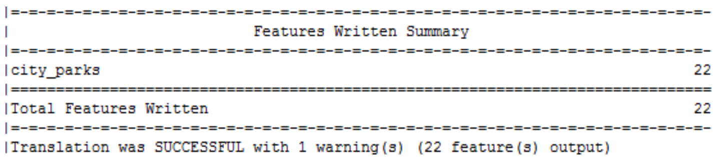
Writer
Format: Google Earth KML
Dataset: C:\FMEData\Output\TutorialOutput\Parks.kml
Coord. System: Same as source
□ Import feature type definitions

Workflow Options
● Static Schema ○ Dynamic Schema

   There are no available parameters for the KML writer at this point, so go ahead and click **OK**.

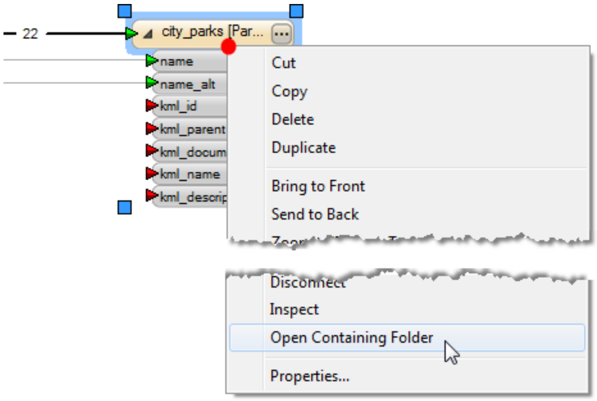
3. Run the translation. Click the green play button to start the translation.

   The translation will now run and will be completed in a few seconds.



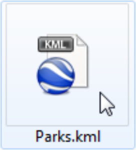
```
|=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
|                           Features Written Summary
|=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
|city_parks                                                                  22
|==============================================================================
|Total Features Written                                                      22
|=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
|Translation was SUCCESSFUL with 1 warning(s) (22 feature(s) output)
```

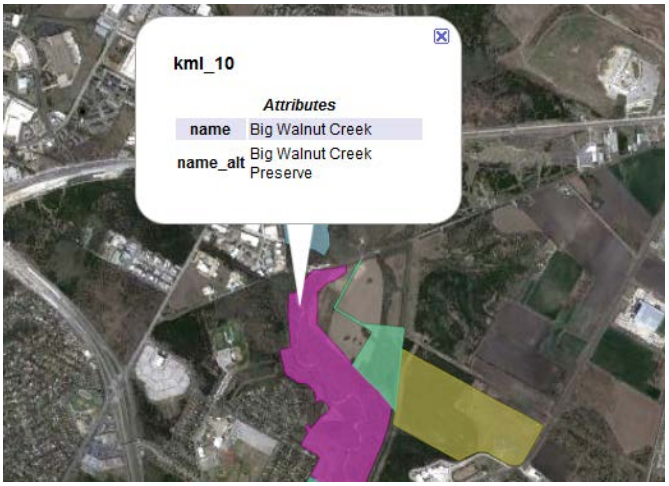
4. Inspect the output. To locate the output folder, right-click on the writer feature type. Choose the option **Open Containing Folder**.



5. Windows Explorer will now open and display the correct location for the output dataset.

   Right-click the file *Parks.kml* and choose **Open with** > **Google Earth**.





Attributes

name	Big Walnut Creek
name_alt	Big Walnut Creek Preserve

Notice how the parks data retains the same color as it did in the source data (you can open the source MapInfo in the FME Universal Viewer to check) and that querying a feature returns a list of attribute values.

KML Data Transformation

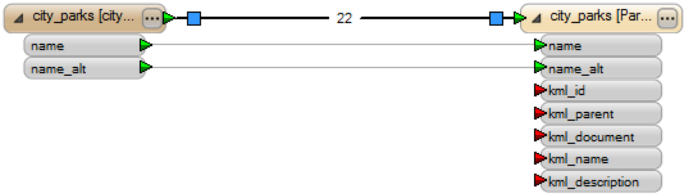
Much of the KML-specific transformation in FME is related to feature symbology and styling

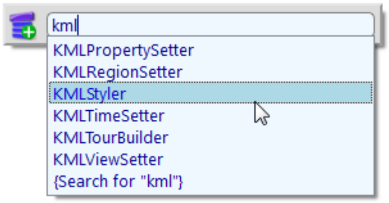
KML format supports tags that define the style and symbology of the features within a dataset. In FME these tags can be set using a series of writer parameters and KML-specific transformers.

Exercise 4: Style KML Data with FME Workbench
Follow these steps to transform and style data as it is converted to KML.

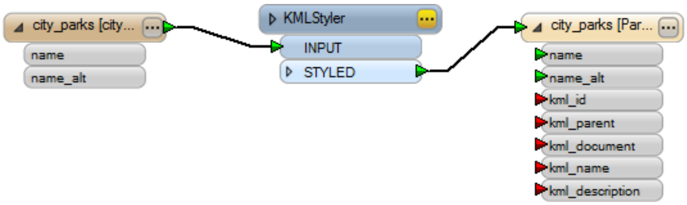
1. Start FME Workbench. Open, or re-create, the workspace from the previous exercise.

2. Place the KMLStyler transformer. Click on the connection between the reader and writer feature types. Type the characters KML. A list of KML-related transformers will appear.





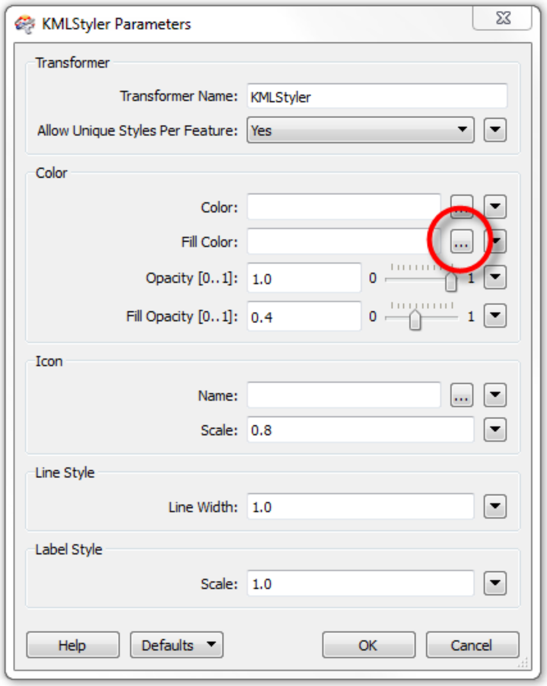
Select KMLStyler from the list. A KMLStyler transformer will be added between the reader and writer.



3. Click on the yellow parameters button on the KMLStyler to open up the parameters dialog for this transformer.

   To set a fill color for the parks, click on the browse button to the right of the Fill Color field:

   In the color-picker dialog, choose a shade of green and click **OK**.



KMLStyler Parameters

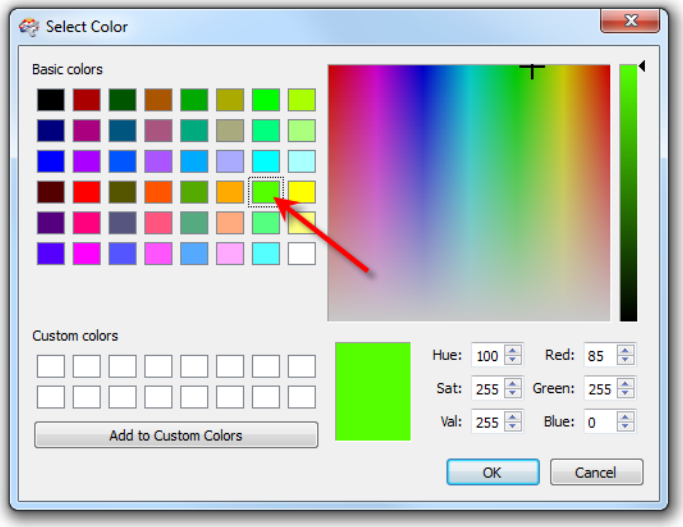
Transformer
Transformer Name: KMLStyler
Allow Unique Styles Per Feature: Yes

Color
Color:
Fill Color:
Opacity [0..1]: 1.0
Fill Opacity [0..1]: 0.4

Icon
Name:
Scale: 0.8

Line Style
Line Width: 1.0

Label Style
Scale: 1.0

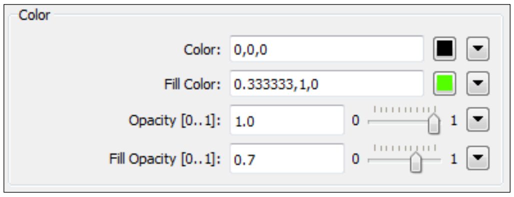


Select Color

Hue: 100
Sat: 255
Val: 255
Red: 85
Green: 255
Blue: 0

4. Back in the KMLStyler parameters dialog, click on the browse button to the right of the Color field, and set the Color value for the polygon outline to black or to the same green as the fill color.

   Set the value for Fill Opacity to 0.7



Color
Color: 0,0,0
Fill Color: 0.333333,1,0
Opacity [0..1]: 1.0
Fill Opacity [0..1]: 0.7

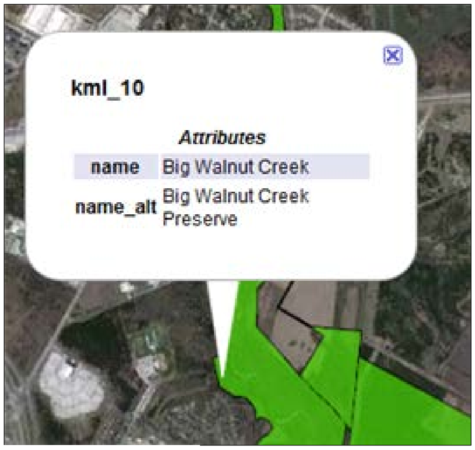
   Click **OK**.

5. Run the translation and inspect the output.

6. Back in Google Earth, choose the **File** > **Revert** menu item (or right-click the object in the “Places” window and choose Revert) to reload the data.

   The park features will now all be the same shade of semi-transparent green, with a common border color.

Notice that, in Google Earth, features are given a default name, such as kml_10

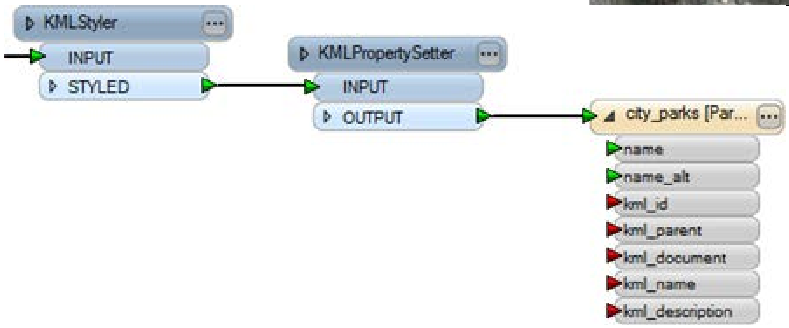


kml_10

Attributes

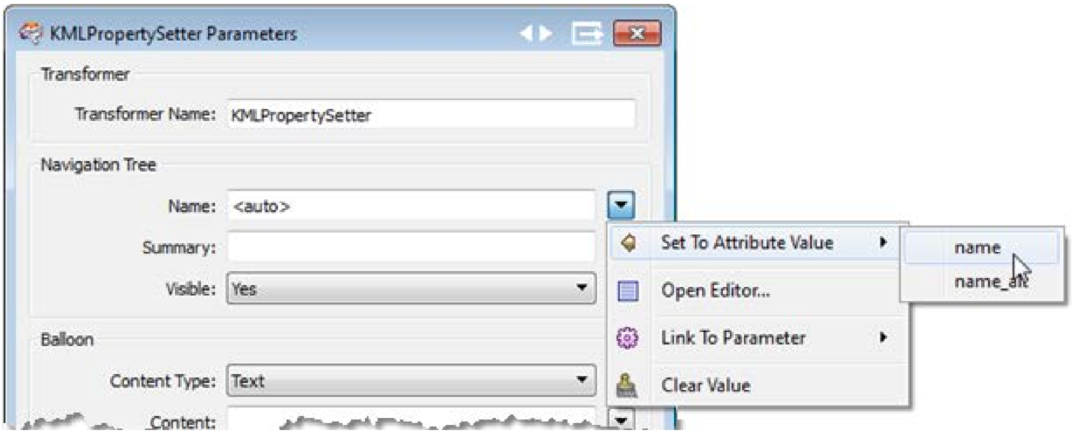
name	Big Walnut Creek
name_alt	Big Walnut Creek Preserve

7. To use the park name as the name for the feature, click on the connection between KMLStyler and writer feature type, then type KMLPropertySetter to place this transformer.



8. Open the parameters dialog for the *KMLPropertySetter*. Fill in these fields:

Name:	click the button to select Set To Attribute Value > name
Summary:	A park in the city of Interopolis



KMLPropertySetter Parameters

Transformer

Transformer Name: KMLPropertySetter

Navigation Tree

Name: <auto>

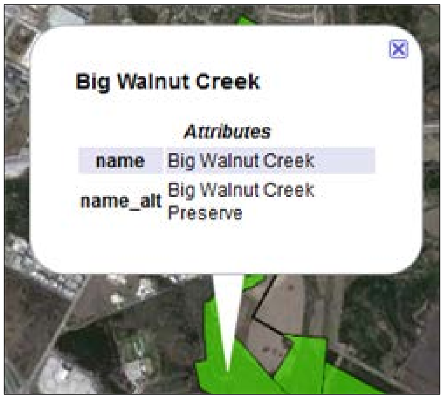
Summary:

Visible: Yes

Balloon

Content Type: Text

Content:



Big Walnut Creek

Attributes

name	Big Walnut Creek
name_alt	Big Walnut Creek Preserve

9. Run the translation and back in Google Earth use the **File** > **Revert** menu item to reload the data.

Now querying a feature will show the feature name, and the Places dialog will show both the name and the description of the feature.

Exercise 5: Add a Point Dataset

Point features are points of interest that have a location but no size, such as mountain peaks or historical markers. Point features in KML are assigned an icon as a map symbol. FME includes a set of built-in icons specifically designed for KML format translations and viewing. The KMLStyler transformer is used to set point feature parameters including icon type, color, size and opacity.

1. In FME Workbench, open or continue with the workspace from the previous exercise.

2. Click **Readers** > **Add Reader** on the menu bar. When the Add Reader dialog opens, fill in these fields:

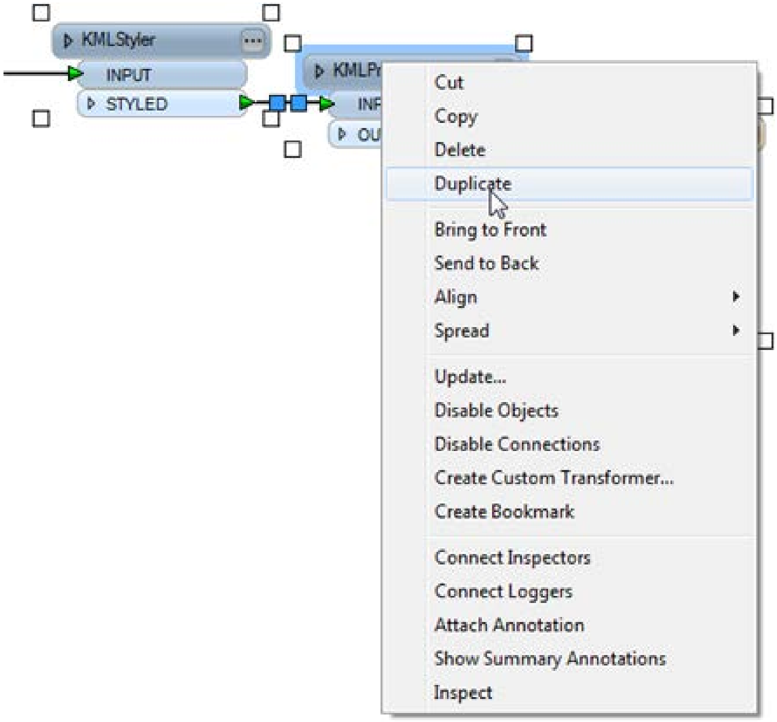
   Reader
   Format: MapInfo MIF/MID
   Dataset: C:\FMEData\Data\BirdSociety\BirdNestPoints.mif

   Click **OK** to add the new reader.

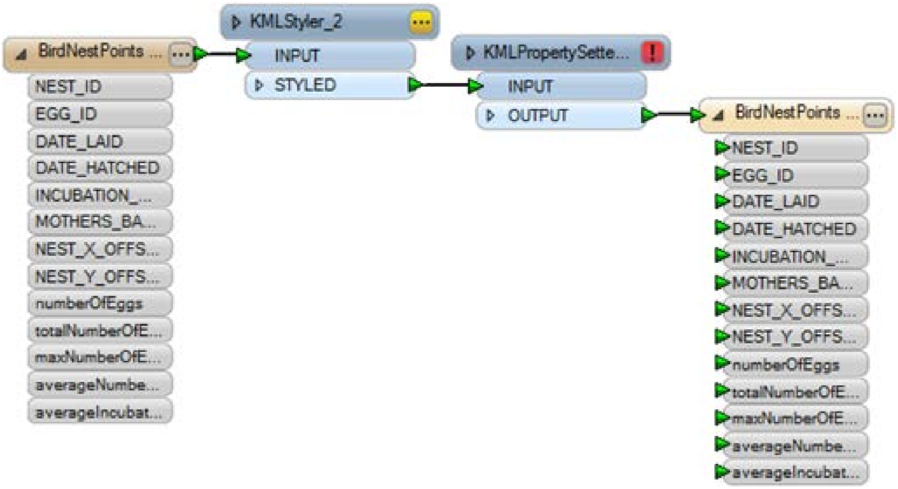
3. Select the previously placed *KMLStyler* and *KMLPropertySetter* transformers.

   Right-click and choose the **Duplicate** option.

   Connect the *BirdNestPoints* feature type to the new transformers.



4. Right-click the *BirdNestPoints* reader feature type, and choose **Duplicate (On Writer)**. Delete the connection that will automatically be created, and connect the *KMLPropertySetter* transformer to the new writer feature type.



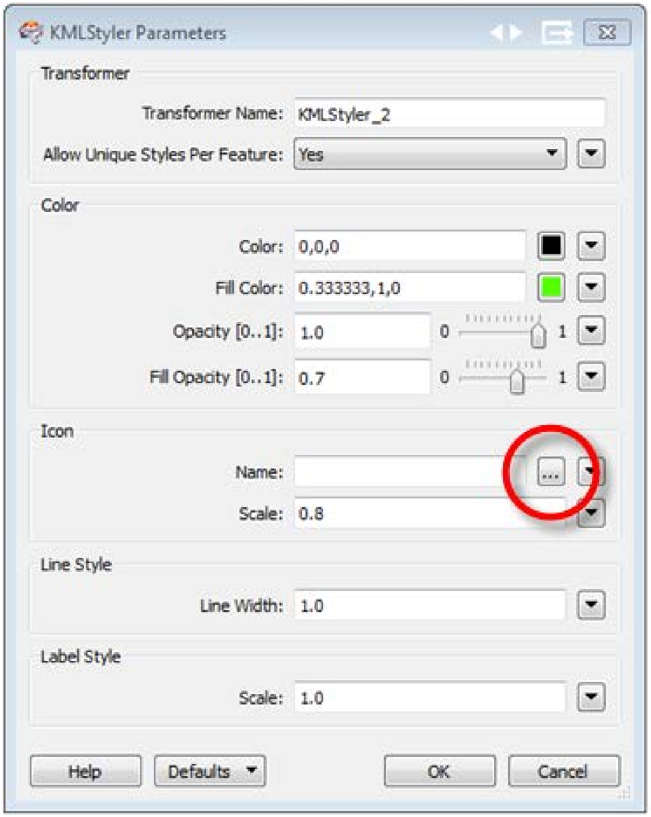
5. Click on the yellow parameters button on *KMLStyler* to open up the parameters dialog.

   To set an icon for the bird nests, click on the browse button to the right of the Name field. In the icon-picker dialog, choose icon C2 and click **OK**.

   Set icon scale to 0.4

   Click **OK** again to close the KMLStyler parameters dialog.

   NB: *An image URL can be entered into the name field instead of picking an FME icon. For example, try http://fme.ly/nest with an icon scale of 1.0*



KMLStyler Parameters

Transformer

Transformer Name: KMLStyler_2

Allow Unique Styles Per Feature: Yes

Color

Color: 0,0,0

Fill Color: 0.333333,1,0

Opacity [0..1]: 1.0 (0 – 1)

Fill Opacity [0..1]: 0.7 (0 – 1)

Icon

Name:

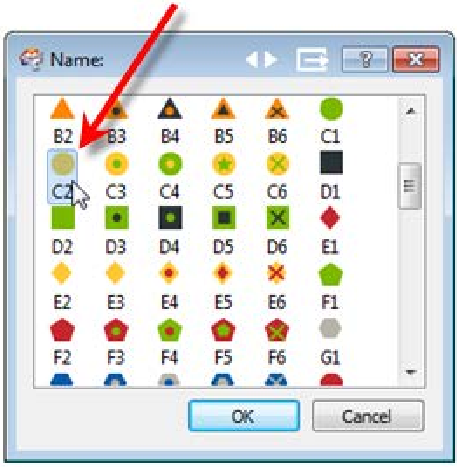
Scale: 0.8

Line Style

Line Width: 1.0

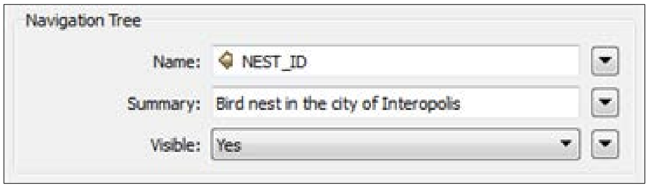
Label Style

Scale: 1.0



6. Set the properties for these bird nest features. Click on the red parameters button on the *KMLPropertySetter* to open up its parameters dialog. Fill in these fields:

Name:	set to the attribute value NEST_ID
Summary:	Bird nest in the city of Interopolis



Navigation Tree

Name: NEST_ID

Summary: Bird nest in the city of Interopolis

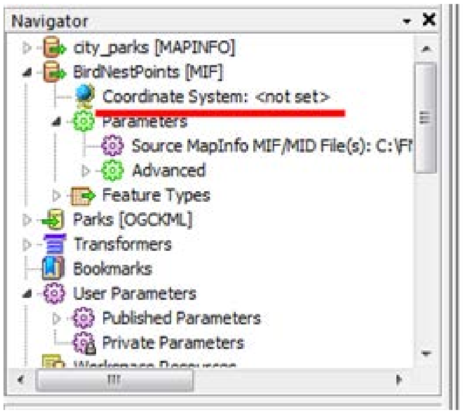
Visible: Yes

7. Run the translation.

   Back in Google Earth, reload the data.

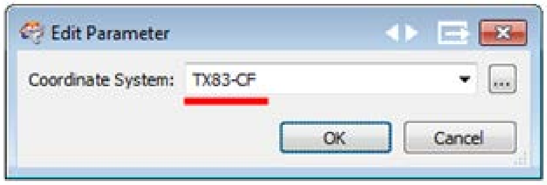
8. Problem! Google opens the data in the center of the Atlantic Ocean. There must be a problem with the bird nest coordinate system.

   In the Navigator windows, locate the Coordinate System parameter for the *BirdNestPoints* reader. Double-click the parameter. When prompted enter a coordinate system of TX83-CF and click **OK**.



Navigator

- city_parks [MAPINFO]
- BirdNestPoints [MIF]
  - Coordinate System: <not set>
  - Parameters
    - Source MapInfo MIF/MID File(s): C:\FI
    - Advanced
  - Feature Types
- Parks [OGCKML]
- Transformers
- Bookmarks
- User Parameters
  - Published Parameters
  - Private Parameters
- Workspace Resources



Edit Parameter

Coordinate System: TX83-CF

9. Re-run the translation.

   Back in Google Earth, reload the data.

   You will now see the bird nests represented by icons.

220
27
272
121
37
161
29
106
64
128
175
112
224
259
264
291
43
180
172
88
310
186
267
329
227
50
215
153
124
266
265
233
257
104
221
249
209
76
60

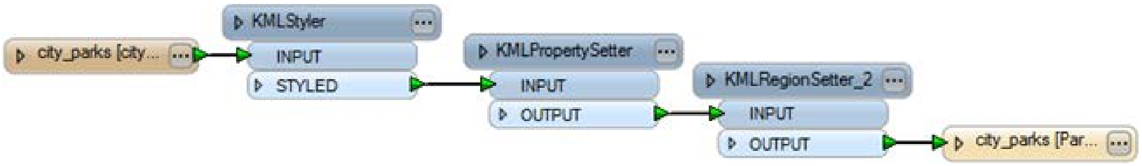
Exercise 6: Regionate KML Data with FME Workbench

Follow these steps to create regions that set Level Of Detail bounds within KML.

1. Start FME Workbench. Open or recreate the workspace from the previous exercise.

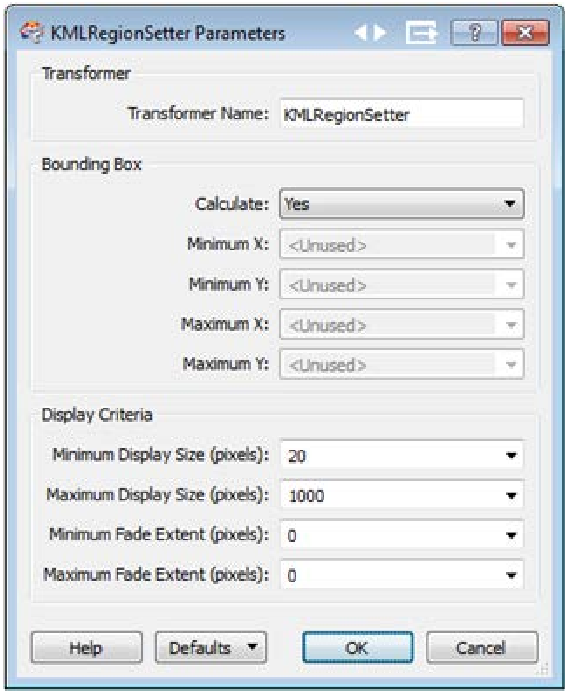
2. Place a *KMLRegionSetter* transformer. Click on the connection between the *KMLPropertySetter* transformer and the writer feature type.

   Type the characters KML. Select *KMLRegionSetter* from the list.



3. Click on the red parameters button on the *KMLRegionSetter*. Fill in these fields:

Bounding Box	**Calculate:**	Yes
Display Criteria	**Minimum Display Size:**	20
Display Criteria	**Maximum Display Size:**	1000



KMLRegionSetter Parameters

Transformer

Transformer Name: KMLRegionSetter

Bounding Box

Calculate: Yes

Minimum X: <Unused>

Minimum Y: <Unused>

Maximum X: <Unused>

Maximum Y: <Unused>

Display Criteria

Minimum Display Size (pixels): 20

Maximum Display Size (pixels): 1000

Minimum Fade Extent (pixels): 0

Maximum Fade Extent (pixels): 0

4. Run the translation.

   Back in Google Earth, reload the data.

   You will now see that city parks appear and disappear as you zoom in and out.

5. Place a second *KMLRegionSetter* transformer in the connection between the *KMLPropertySetter* transformer and the birds nests writer feature type.

Bounding Box	**Calculate:**	No
	Minimum X:	-97.7
	Minimum Y:	30.2
	Maximum X:	-97.5
	Maximum Y:	30.4
Display Criteria	**Minimum Display Size:**	5000
	Maximum Display Size:	-1

NB: *It's better to set an explicit bounding box for point features and not use the option to calculate it. That's because point features produce an infinitely small bounding box and would never show in the output. The numbers above cover a good area of the data.*

6. Run the translation and reload Google Earth.

You will now see that bird nests also appear as you zoom in and out, but because Maximum Display Size = -1, will never disappear because you have zoomed in too close.

Exercise 7: Create 3D KML Data with FME Workbench

Follow these steps to create 3D building features within KML.

1. Start FME Workbench. In the Start tab choose the option to **Generate workspace**.

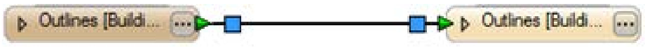
2. When the New Workspace dialog opens, fill in these fields:

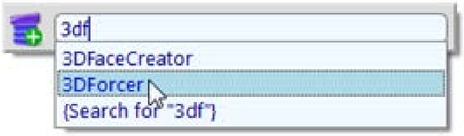
   Reader
   Format: Esri Geodatabase (File Geodatabase API)
   Dataset: C:\FMEData\Data\Properties\Buildings.gdb

   Writer
   Format: Google Earth KML
   Dataset: C:\FMEData\Output\TutorialOutput\Buildings.kml

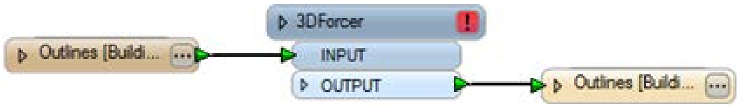
   Click **OK** to create the workspace.

3. Click on the connection between the reader and writer feature types.
   Type the characters 3DF. Select *3DForcer* from the list of transformers.



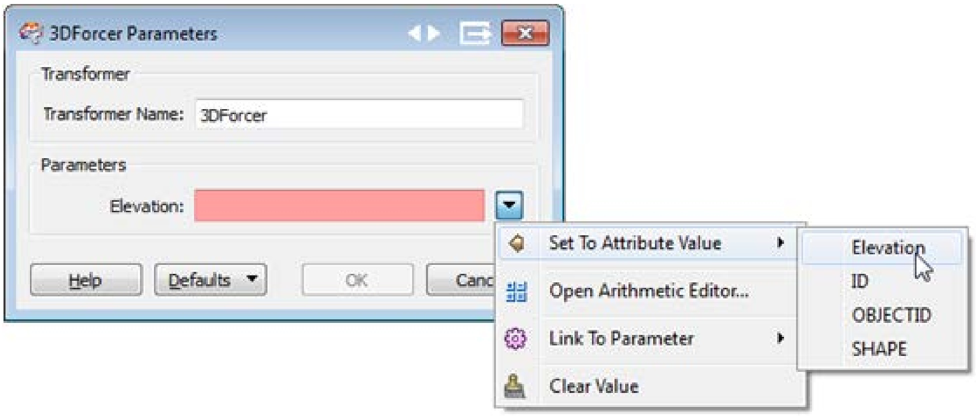


A *3DForcer* transformer will be added between the reader and writer.



4. Open up the parameters dialog for the *3DForcer* transformer. This transformer gives a Z value to the coordinates of 2D features.

   Select the attribute Elevation as the source for the Z values. This attribute contains a building height for each feature.



3DForcer Parameters

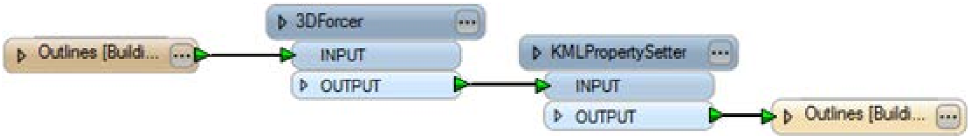
Transformer

Transformer Name: 3DForcer

Parameters

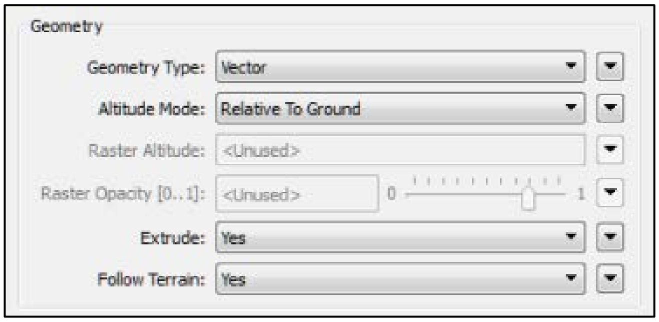
Elevation:

5. Place a *KMLPropertySetter* transformer between the *3DForcer* and the writer feature type.



   Open the properties dialog. Fill in these fields:

Altitude Mode:	Relative to Ground
Extrude:	Yes



Geometry

Geometry Type: Vector

Altitude Mode: Relative To Ground

Raster Altitude: <Unused>

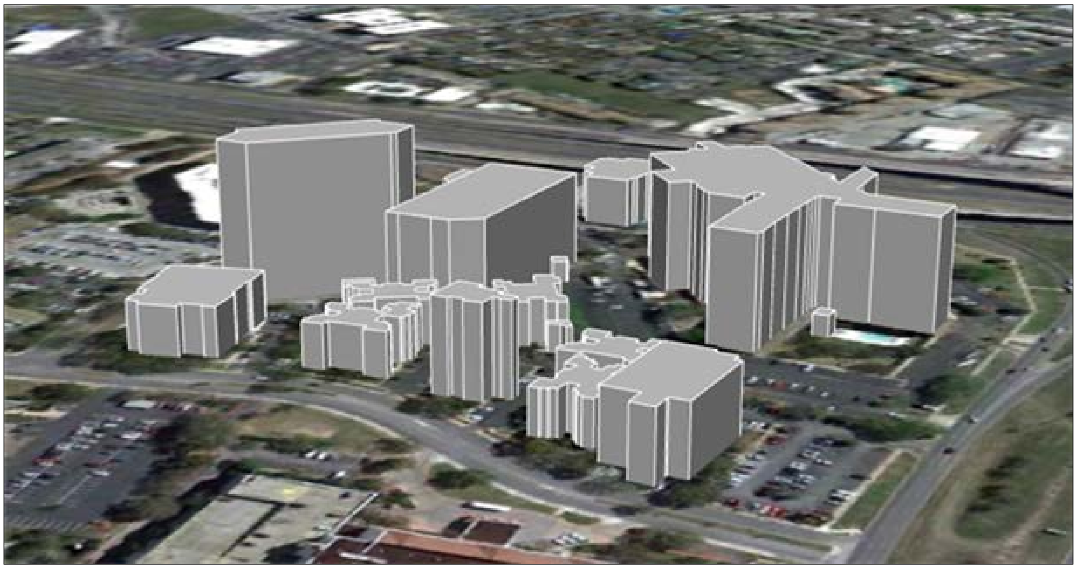
Raster Opacity [0..1]: <Unused> 0 1

Extrude: Yes

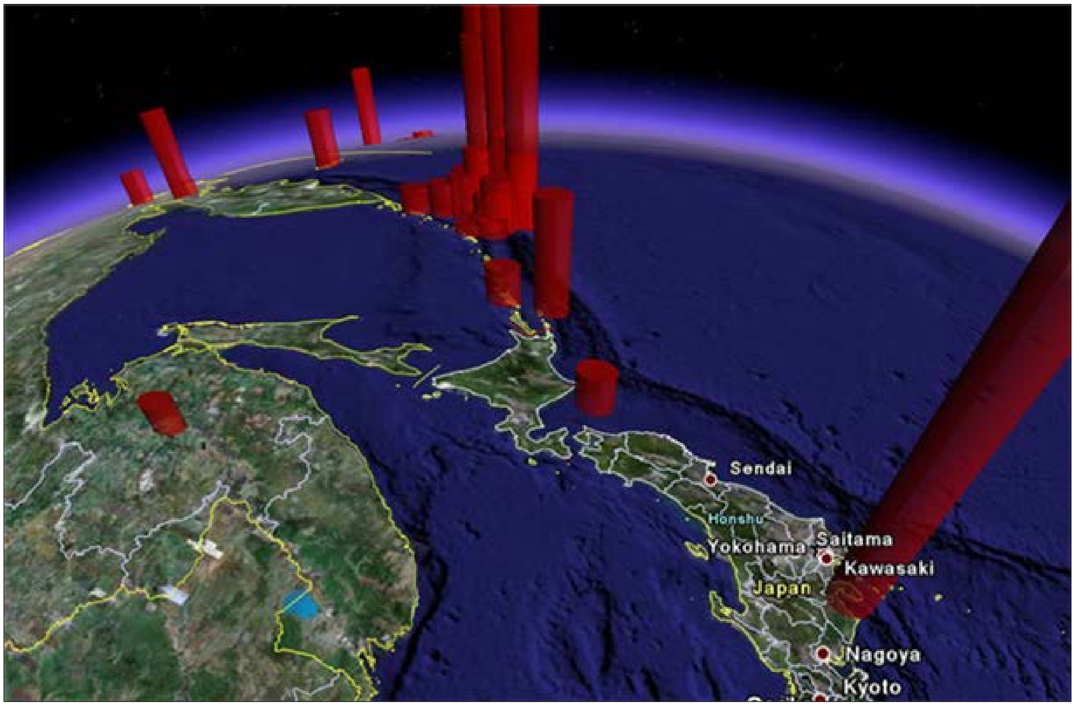
Follow Terrain: Yes

6. Run the translation and inspect the output.

   Change the view aspect and you will now see three-dimensional building features.



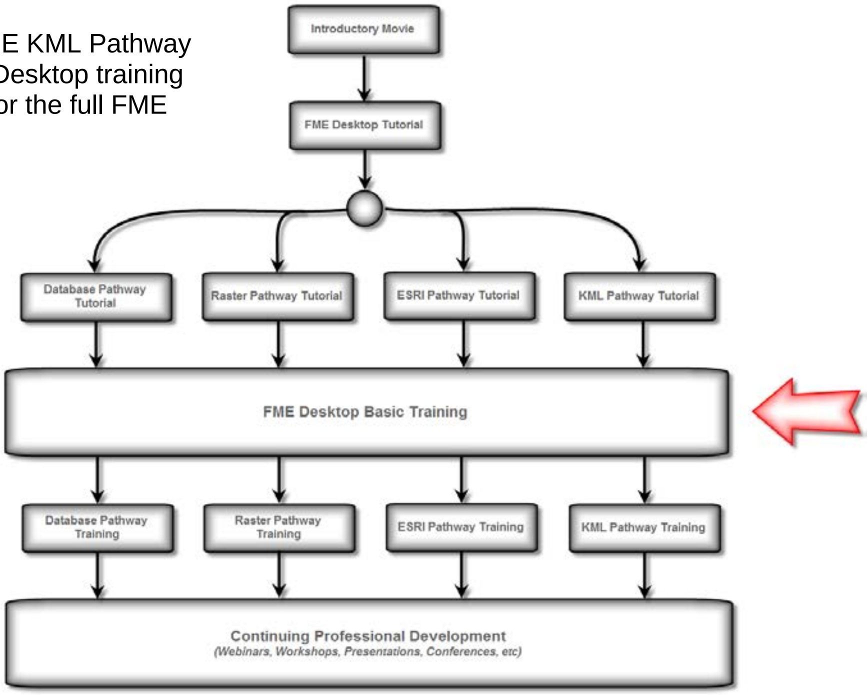
NB: *This 3D forcing technique can be used in many scenarios, even for thematic mapping data such as earthquake intensity:*



What's Next?

This document is a basic introduction to using KML data with FME

Next Step
The next step in the FME KML Pathway is to take a basic FME Desktop training course, in preparation for the full FME KML Training Course.



Further information on all training options is available on the Safe Software web site at www.safe.com/training

Many other resources for FME Desktop technical information can be located through the FMEpedia knowledgebase at http://fmepedia.safe.com