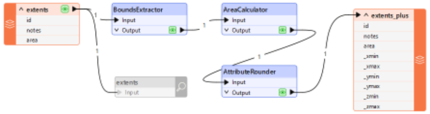
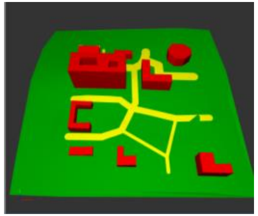
FME Form for beginners





Giorgio Agugiaro

Last update: 7 September 2024

License

Brief overview of FME Form

FME in a nutshell
Readers and writers
Transformers
Data types
Geometry model
Other stuff

- In a nutshell: FME Form performs spatial data **ETL** (**E**xtract, **T**ransform, **L**oad)
  - https://en.wikipedia.org/wiki/Spatial_ETL
  - In FME jargon: **Reader**(s) → **Transformer**(s) → **Writer**(s)
- Interface areas (see next slides)
  - Workspace
  - Navigator
  - Transformer gallery
  - Translation log (and other views)
  - Menu and toolbar
- Data exploration
  - Visual Preview (in FME Workbench)
  - FME Data inspector (external application)
- Some advanced functionalities (not treated in these slides)
  - Customisation: Create your own transformers
  - Programmability: Python-based transformers can be created
  - Automation: Parent transformers can launch children transformers

FME Form Workbench

FME in a nutshell

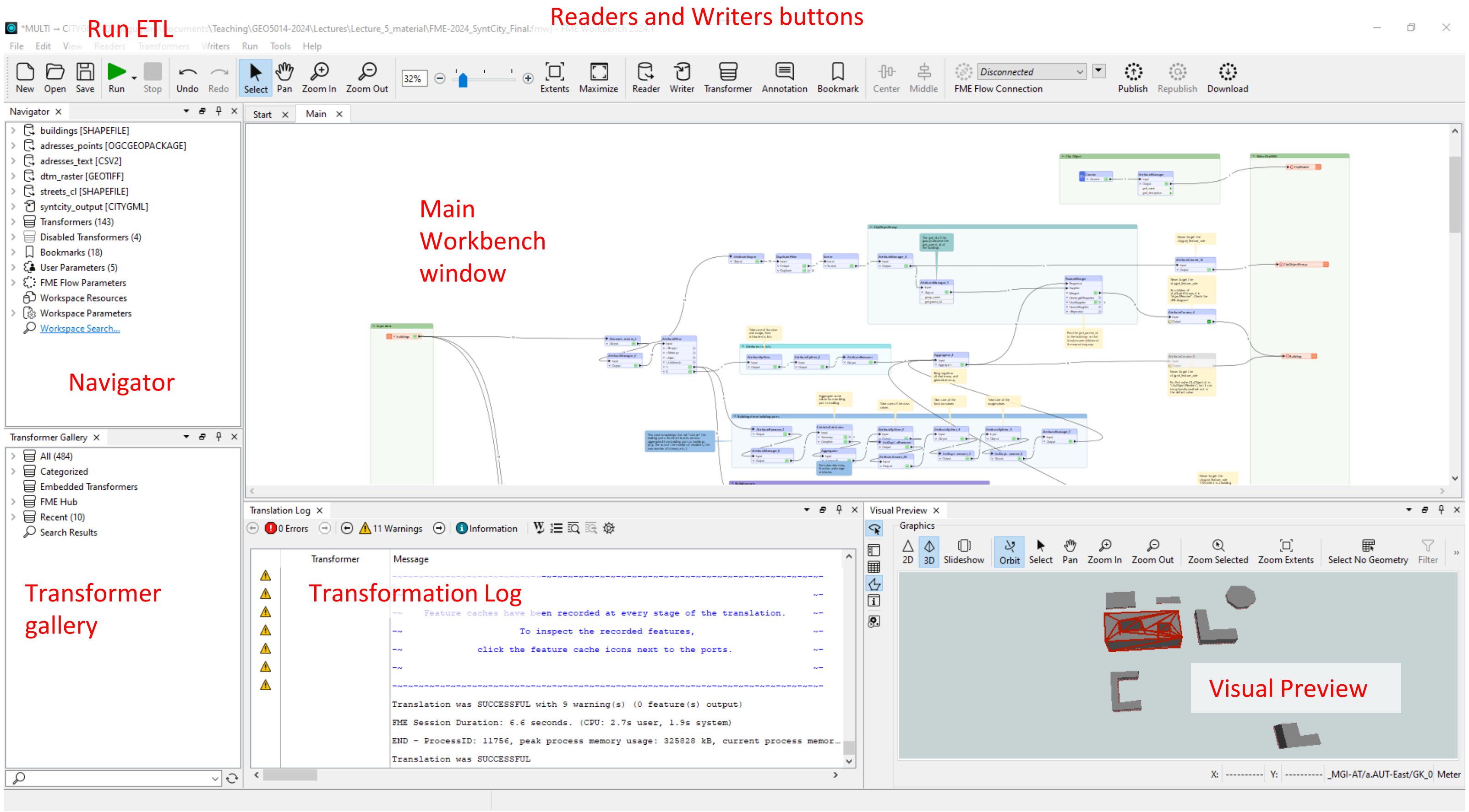
Readers and writers

Transformers

Data types

Geometry model

Other stuff



FME Data Inspector

FME in a nutshell

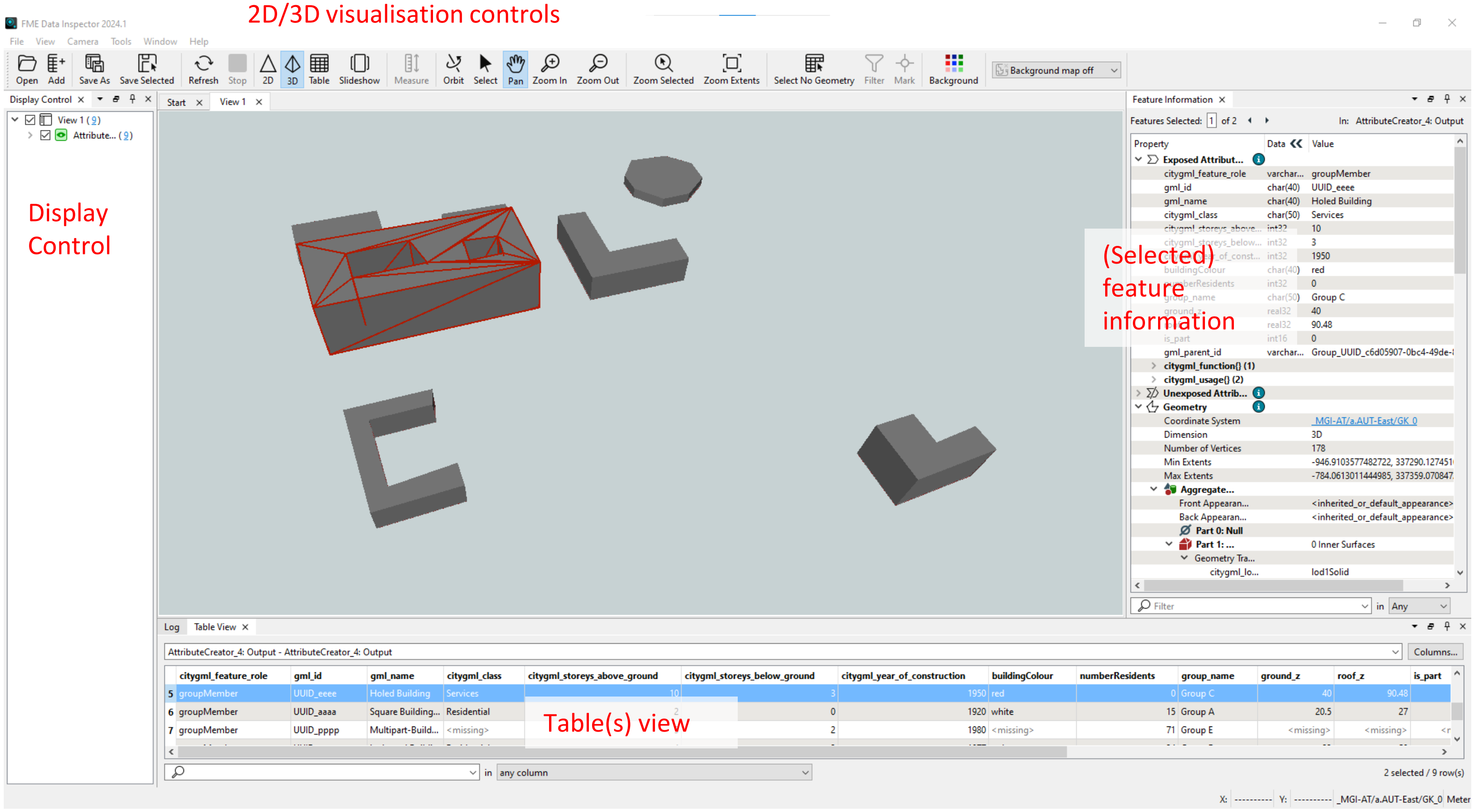
Readers and writers

Transformers

Data types

Geometry model

Other stuff



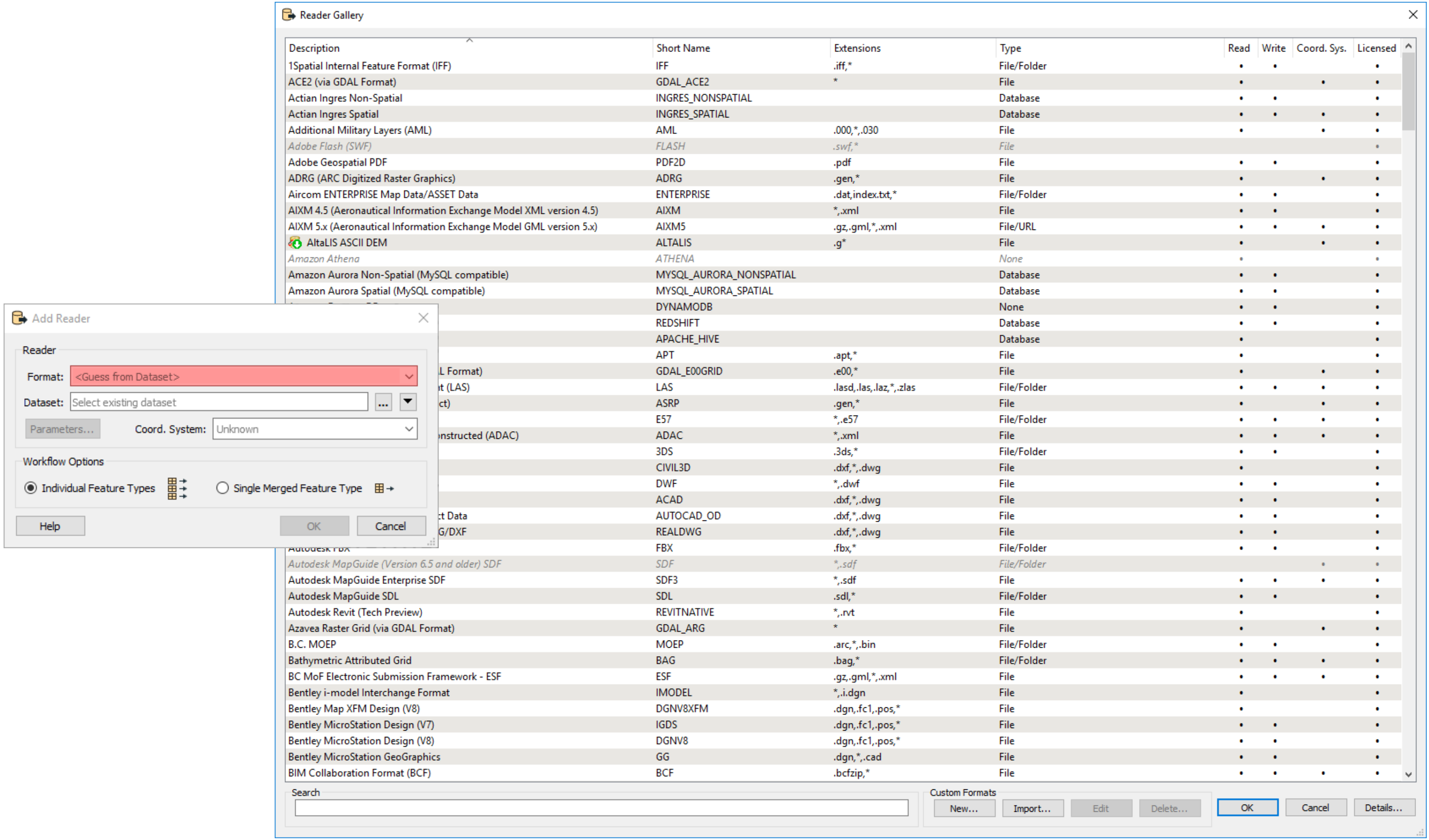
Readers & writers

- Lots of spatial and non-spatial formats
  - https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_ReadersWriters/Format-List-All.htm
  - E.g. vector: DWG/DXG, shp, txt, csv, Esri Geodatabase, GeoJSON, OBJ, PostgreSQL/PostGIS, R, WMS, WFS, etc…
  - Raster: JPG, GeoTiff, …
  - CityGML + Application Domain Extensions (ADEs)
  - CityJSON

- **WARNING: There is NO MAGIC**! An FME reader/writer does not “save” you from the need to know the characteristics of a specific format.
  But it simplifies life a lot! 😋

Readers & writers

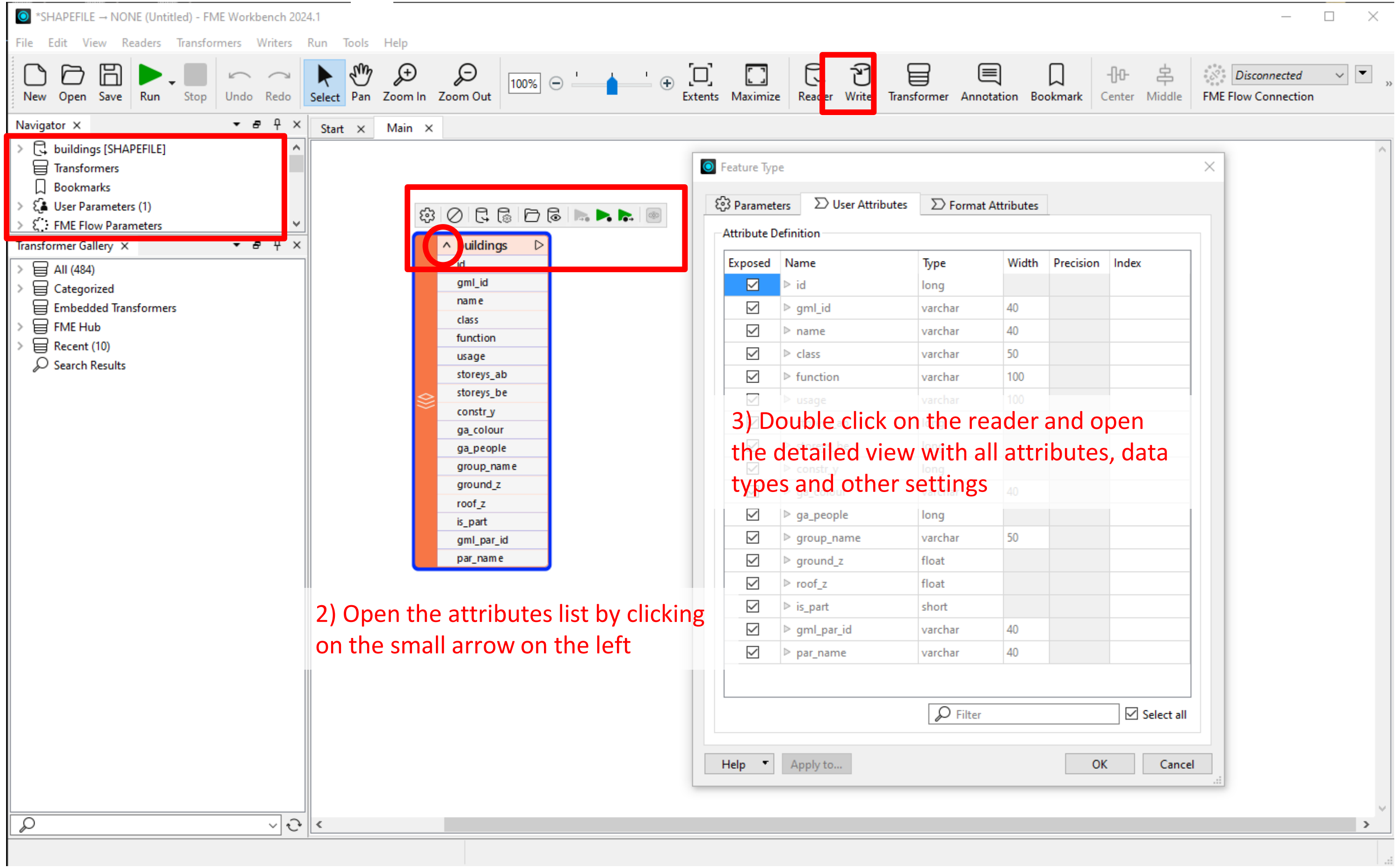
FME in a nutshell
Readers and writers
Transformers
Data types
Geometry model
Other stuff



Reader(s)

1a) Click the reader button and select the reader depending on format, OR
1b) Type the reader format (e.g. SHAPEFILE) in the workbench area, OR
1c) Drag and drop the file in the workbench area

FME in a nutshell
Readers and writers
Transformers
Data types
Geometry model
Other stuff

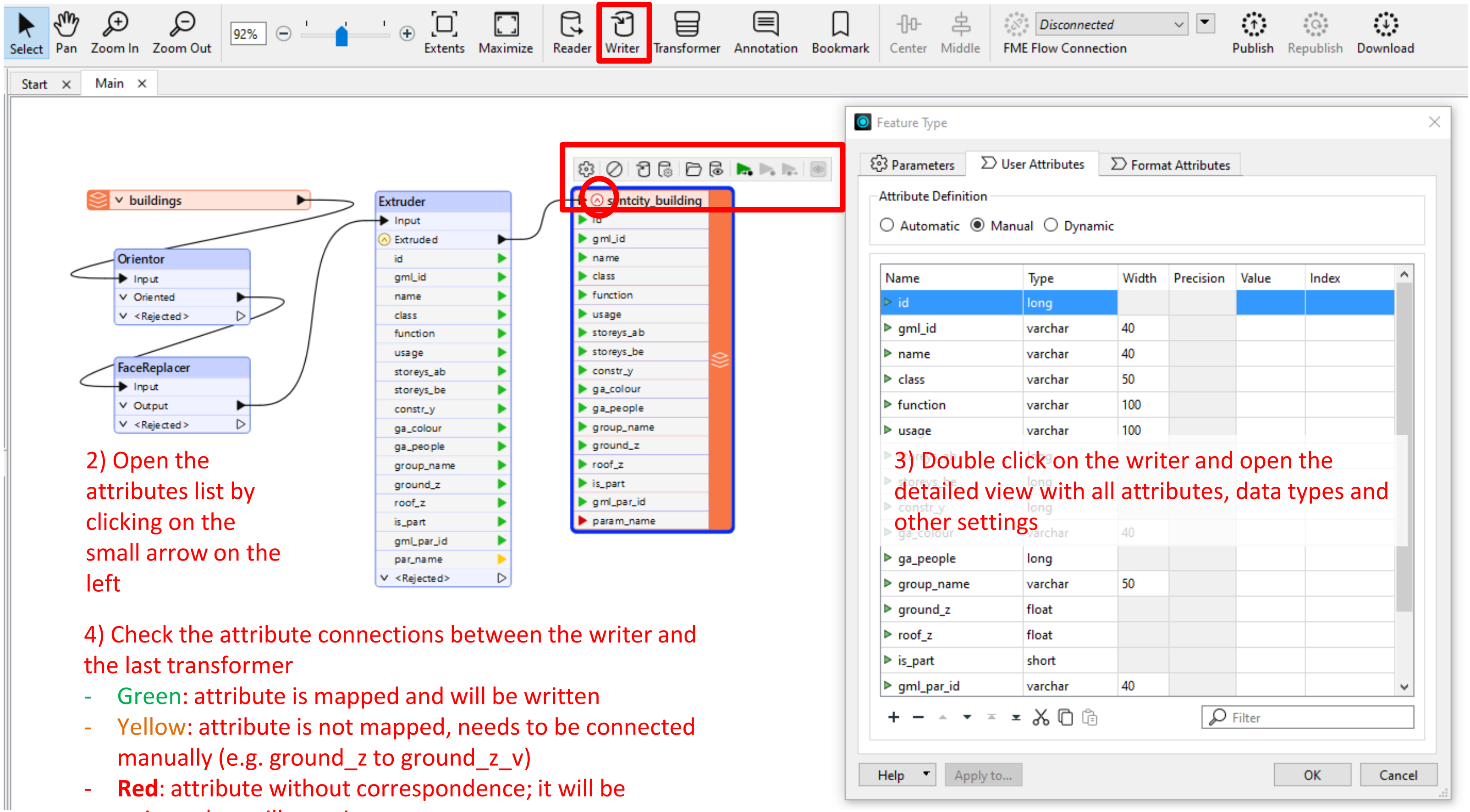


3) Double click on the reader and open the detailed view with all attributes, data types and other settings

2) Open the attributes list by clicking on the small arrow on the left

Writer(s)

1a) Click the writer button and select the reader depending on format, OR
1b) Type the writer format (e.g. SHAPEFILE) in the workbench area



2) Open the attributes list by clicking on the small arrow on the left

3) Double click on the writer and open the detailed view with all attributes, data types and other settings

4) Check the attribute connections between the writer and the last transformer

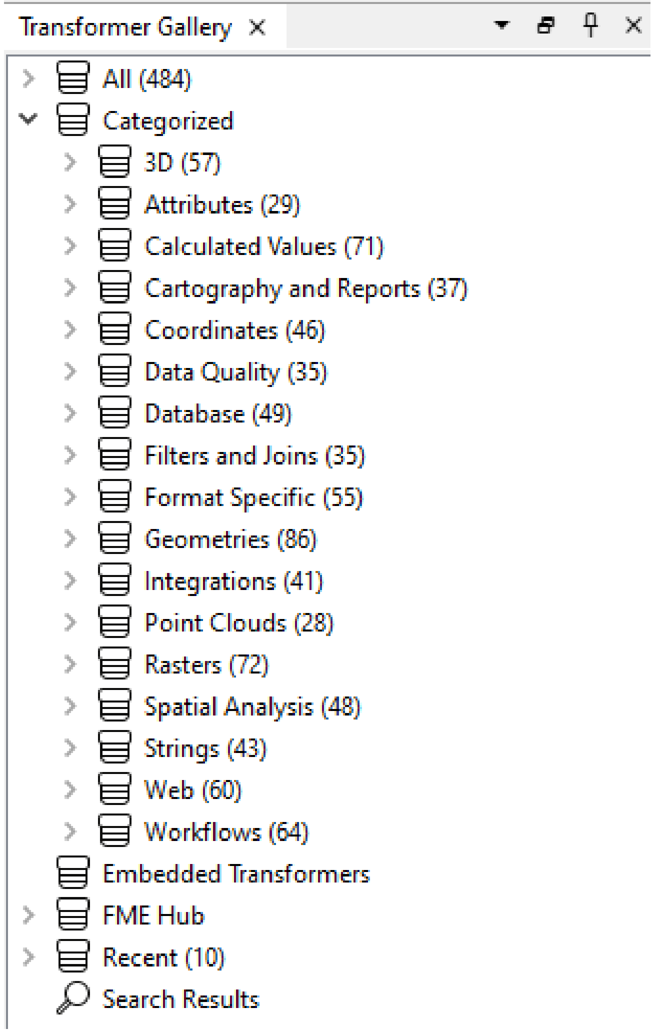
- Green: attribute is mapped and will be written
- Yellow: attribute is not mapped, needs to be connected manually (e.g. ground_z to ground_z_v)
- **Red**: attribute without correspondence; it will be written, but will remain empty

TUDelft 3Dgeoinfo

FME in a nutshell
Readers and writers
Transformers
Data types
Geometry model
Other stuff

Transformer(s)

- A transformer… *transforms* data! ☺
- Circa 500 transformers, thematically classified
- In general, transformers work sequentially (FIFO = First In, First out)
- Possibility to interface to e.g. Python
- Possibility to define/create your own transformer
- If you feel lost: use the help!

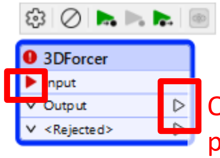


Transformer(s)

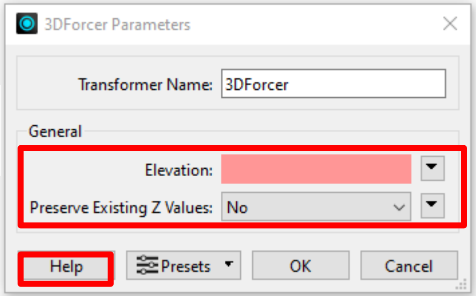
1a) Double click on the transformer name in the Transformer gallery, OR
1b) Drag and drop the transformer from the Transformer gallery into the workbench area OR
1c) Directly type the transformer name (e.g. 3Dforcer) in the workbench area

FME in a nutshell
Readers and writers
Transformers
Data types
Geometry model
Other stuff

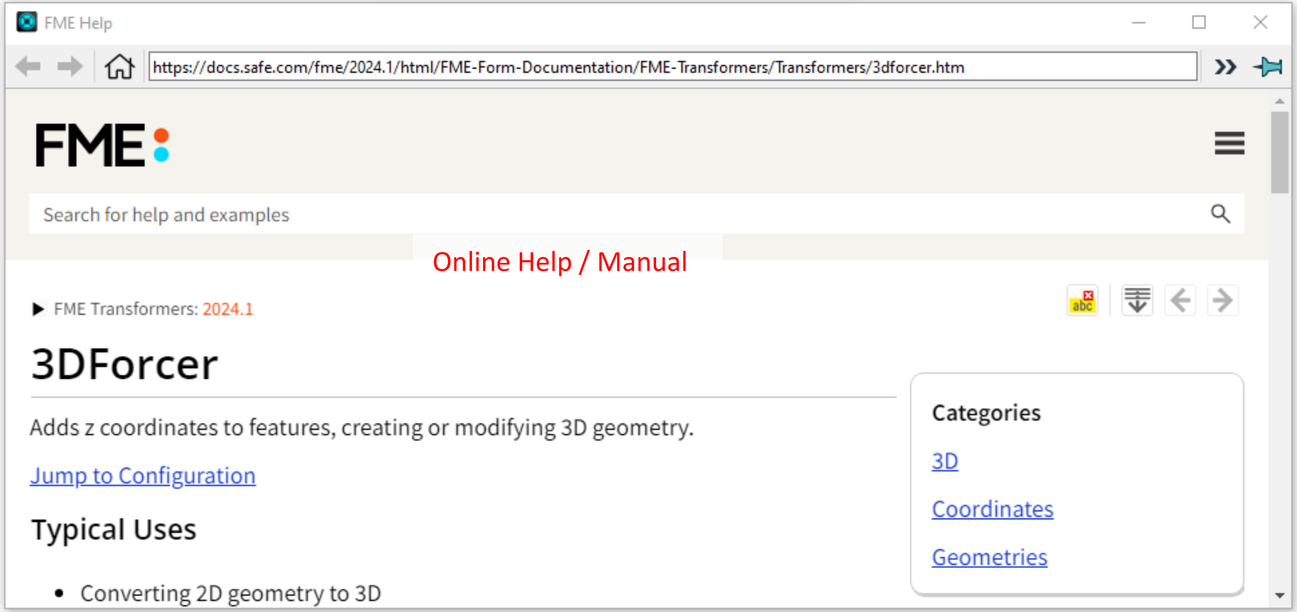
Input port



Output port(s)

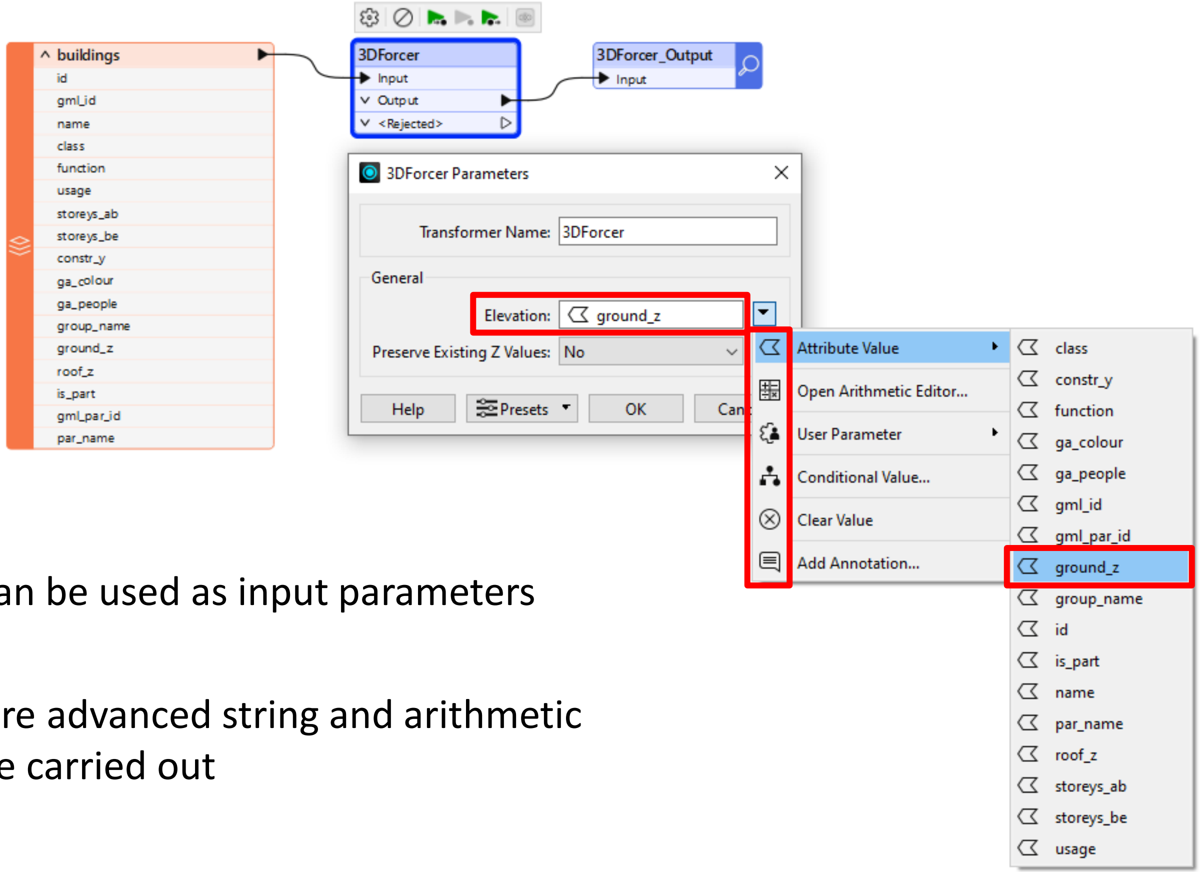


Transformer Parameters



Online Help / Manual

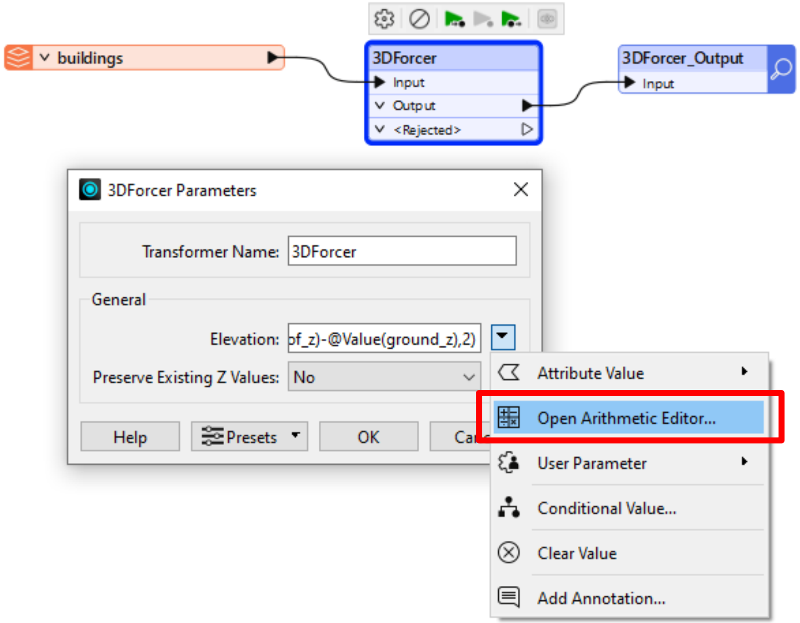
Transformer(s)

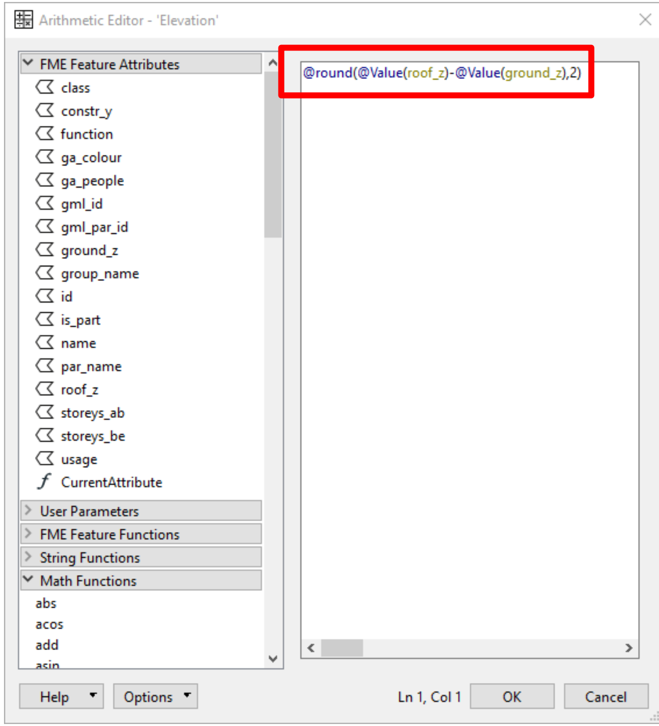


- Attribute fields can be used as input parameters
- Alternatively, more advanced string and arithmetic operations can be carried out

Transformer(s)

FME in a nutshell
Readers and writers
Transformers
Data types
Geometry model
Other stuff





- Alternatively, more advanced string and arithmetic operations can be carried out

Data types

- For **non-spatial data** FME Form *generally* takes care of automatically choosing the correct type (integer, string, date, Boolean, etc.)
- Data is dealt with as tabular data, generally associated to geometries
- Transformers are available for most typical alphanumeric data operations
- For time/date formats, you may have to use transformers to adapt them (e.g. **DateTimeConverter**)

Data types: lists

- Lists are denoted by a pair of curly brackets (“{}”)
  - E.g. *name_of_list*{}

- Lists can contain attributes associated to each item. A period (“.”) denotes them
  - E.g. trees{}.*name*, trees{}.*species*, etc.

- It is good habit to prefix lists with an underscore (“_”)
  - E.g. _trees{}

- Further info:
  - https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Desktop/!List_Attributes/Understanding-List-Attributes.htm

Data types: spatial data

FME in a nutshell
Readers and writers
Transformers
Data types
Geometry model
Other stuff

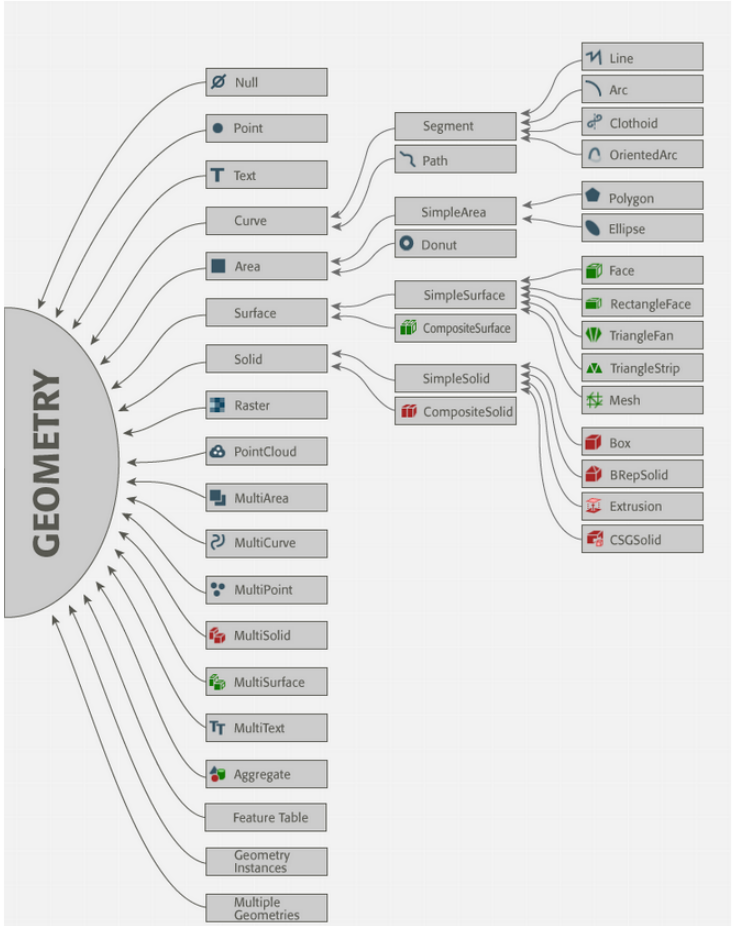
- For **spatial data**, FME has its own internal geometry model. All geometries are mapped to/from it
  - https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Desktop/!FME_Geometry/FME_Geometry_Model.htm
- It is rather intuitive, however with some peculiar semantics
  - **Area**: a 2D polygon with or without holes is called a "Donut" or a "SimpleArea", respectively
  - **Surface**: a 3D geometry (simple or composite) that is made of 3D entities (Face, RectangleFace, … Mesh) and has a surface normal
- Examples of series of geometrical transformations could be:
  - donut -> face -> multisurface
  - polygon -> face -> composite surface -> Brep-solid
- **WATCH OUT:** A "classical" 3D polygon can be many things in FME!
- Geometries can have "internal" attributes called *geometry traits*
  - E.g. they are needed for CityGML

FME in a nutshell
Readers and writers
Transformers
Data types
Geometry model
Other stuff

FME Geometry Model

FME provides a comprehensive geometry model that includes everything from the simplest geometry to the most complex.

Click on a geometry class for more information.

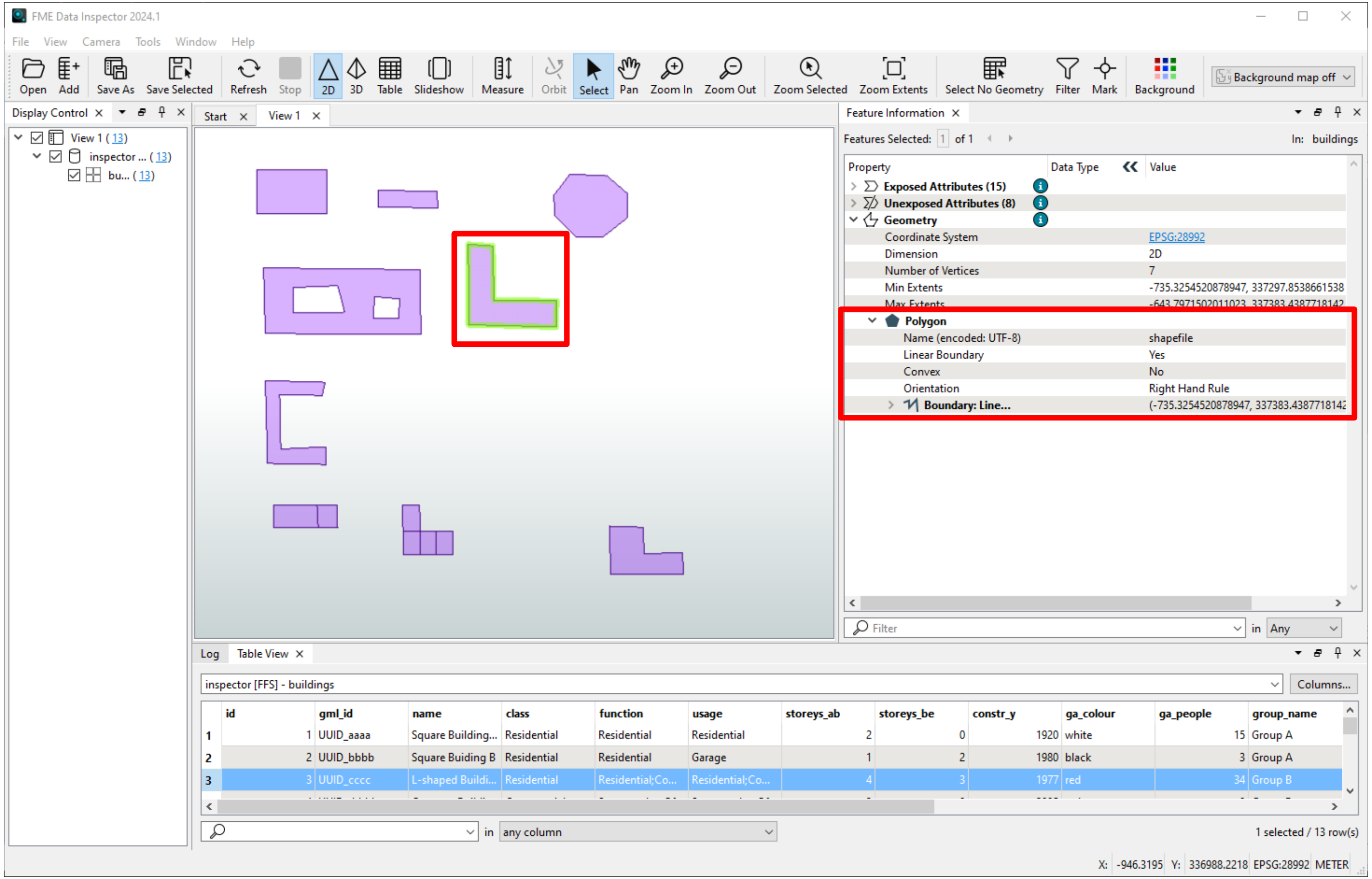


Geometry model

- How to check which geometry data type is used?
- One possibility is to open the **Data Inspector**, select the geometry and check in the Feature Information tab. At the end, you will see the geometry type and the associated geometry traits (if any)
- The hierarchy and the nested geometries will be shown, too.
- See next slides for examples

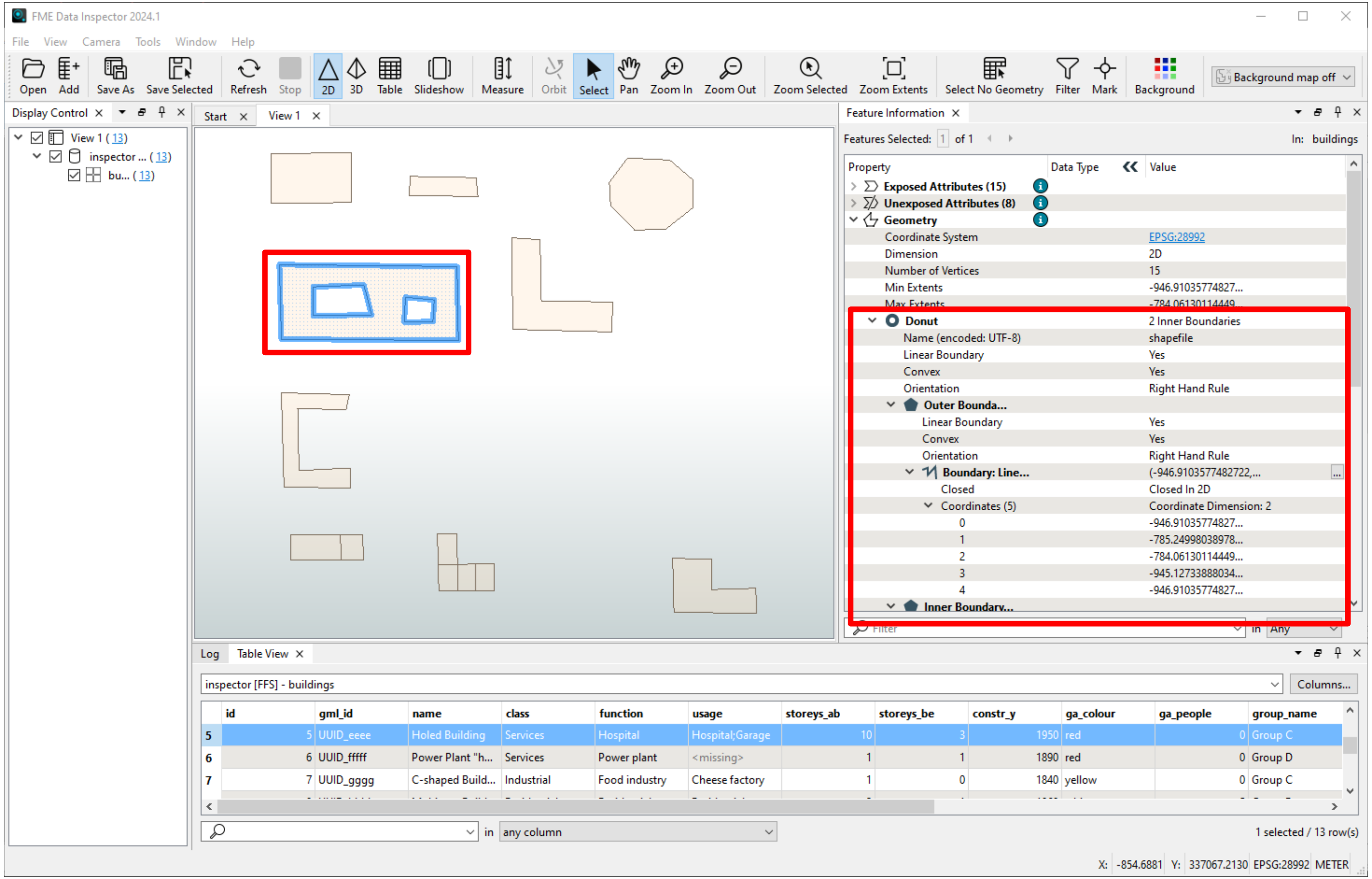
Geometry model

FME in a nutshell
Readers and writers
Transformers
Data types
Geometry model
Other stuff



Geometry model

FME in a nutshell
Readers and writers
Transformers
Data types
Geometry model
Other stuff



Geometry model

- *Polygons* and *donuts* are 2D geometry types

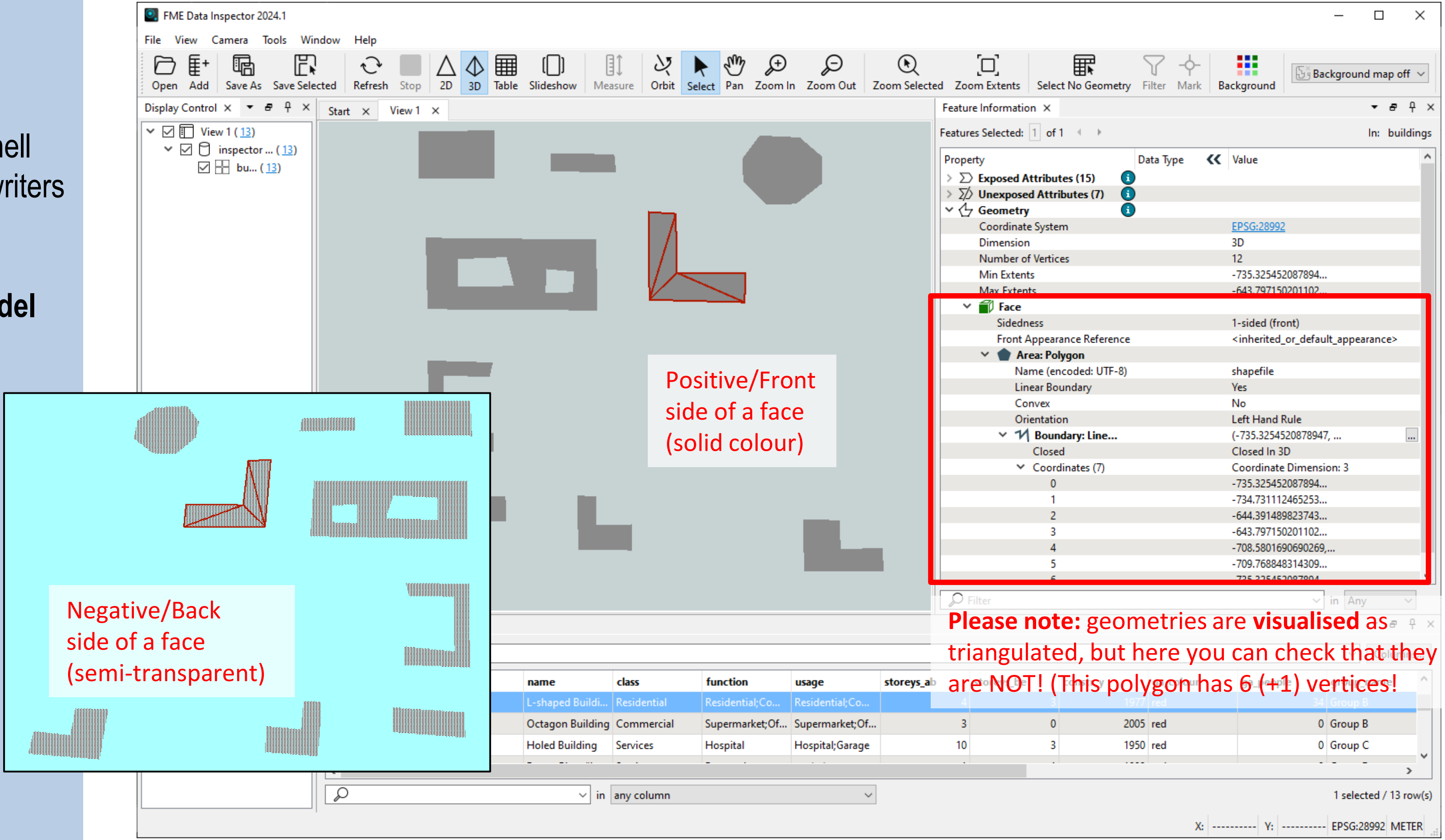
- To transform them into 3D entities, in general they must become (3D) *faces*
  - You can use the **FaceReplacer** transformer, after (optionally) forcing them to 3D using a **3DForcer** transformer

- Faces have the additional property of having a positive and negative *side* (wrt. the surface normal vector)
  - You may have to check the orientation with the **Orientor** transformer

- The two sides of a face are visualised differently in FME:
  - Solid-colour for the positive/front side of the surface
  - Semi-transparent surface for the negative/back side
  - **Watch out!** When visualised, the triangulation does **not** affect the real geometry. It is only for visualisation purposes (unless you are working with triangular geometries!)

Geometry model

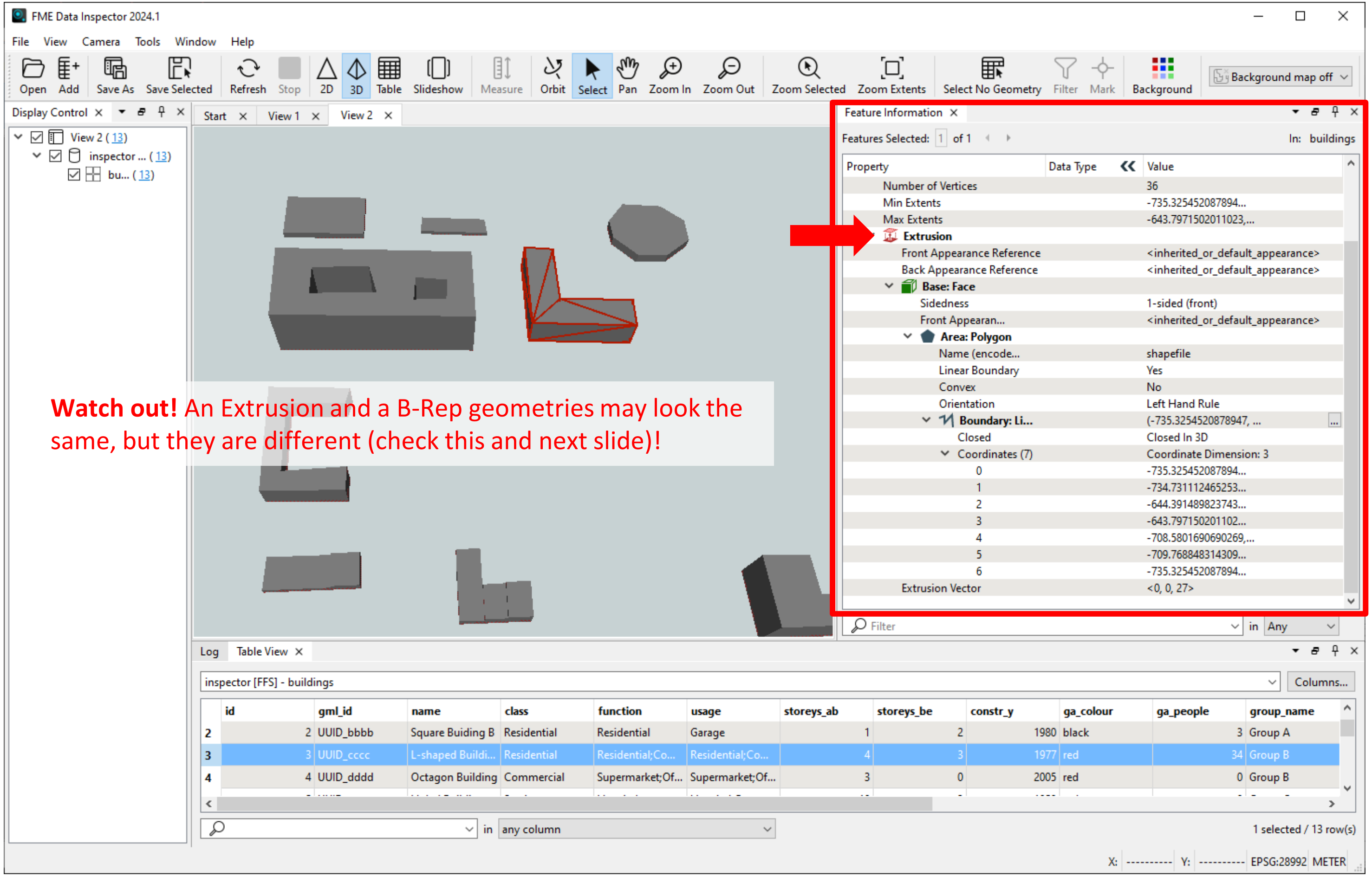
- FME in a nutshell
- Readers and writers
- Transformers
- Data types
- **Geometry model**
- Other stuff



TUDelft 3Dgeoinfo

Geometry model

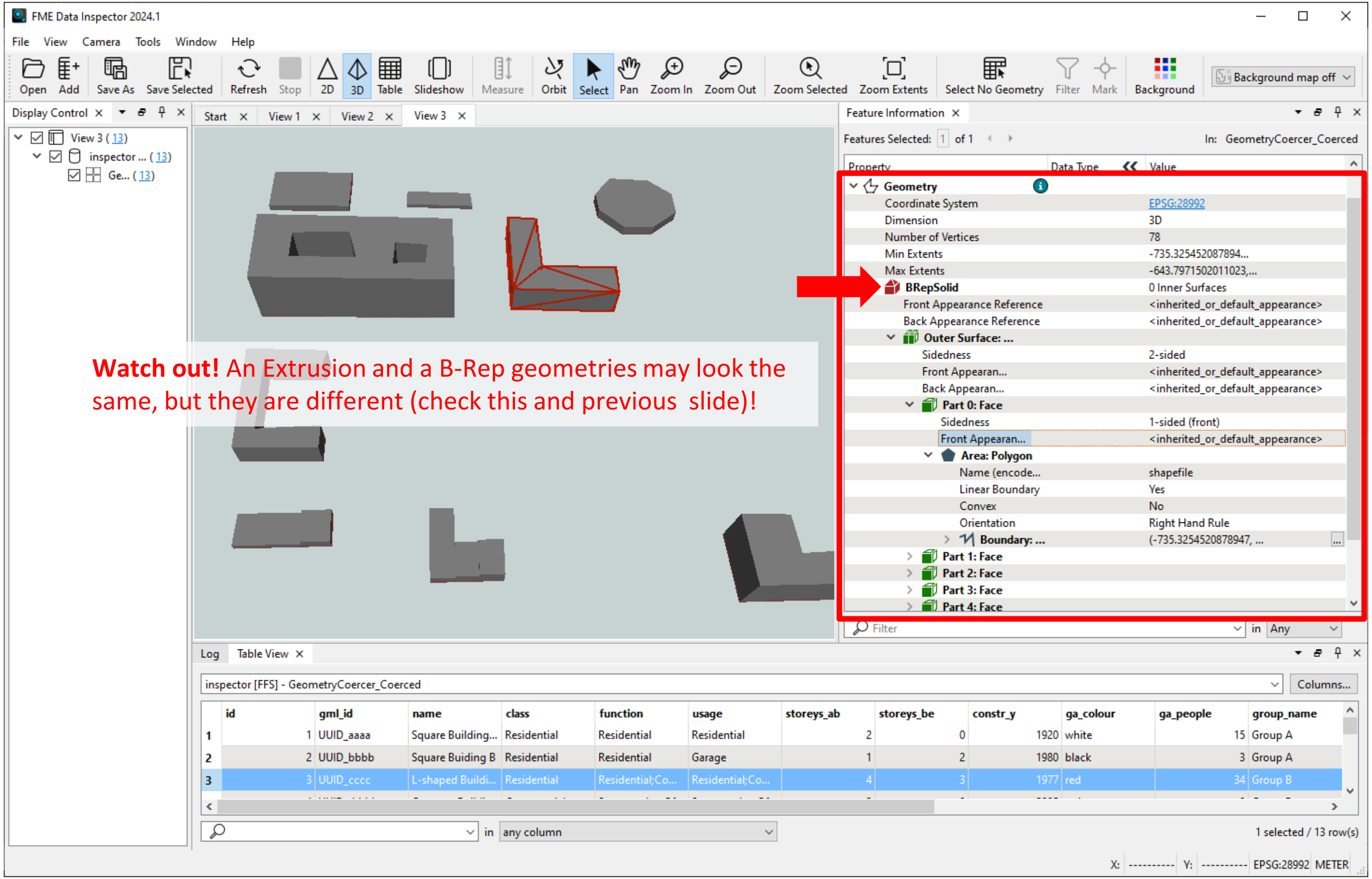
- FME in a nutshell
- Readers and writers
- Transformers
- Data types
- **Geometry model**
- Other stuff



Watch out! An Extrusion and a B-Rep geometries may look the same, but they are different (check this and next slide)!

Geometry model

- FME in a nutshell
- Readers and writers
- Transformers
- Data types
- **Geometry model**
- Other stuff



Watch out! An Extrusion and a B-Rep geometries may look the same, but they are different (check this and previous slide)!

Geometry model

- You can transform geometries from one type to another using, for example, the **GeometryCoercer** transformer
  - Beware: not all transformation combinations are possible
  - Beware: geometries might *look* the same, but they are implicitly different (see previous slides)
    - Both look like 3D extrusions, but they can be modelled as MultiSurface objects, or BRepSolids, or Extrusions

- The **Aggregator/Deaggregator** transformers allow to group/ungroup geometries (and, more in general, features)
  - Handy transformer to go from "simple" geometries to **multi**-geometries.

- The **GeometryFilter** allows to select specific geometry types

- The **GeometryValidator** allows to check the validity and automatically repair (some) errors of geometries

Other stuff

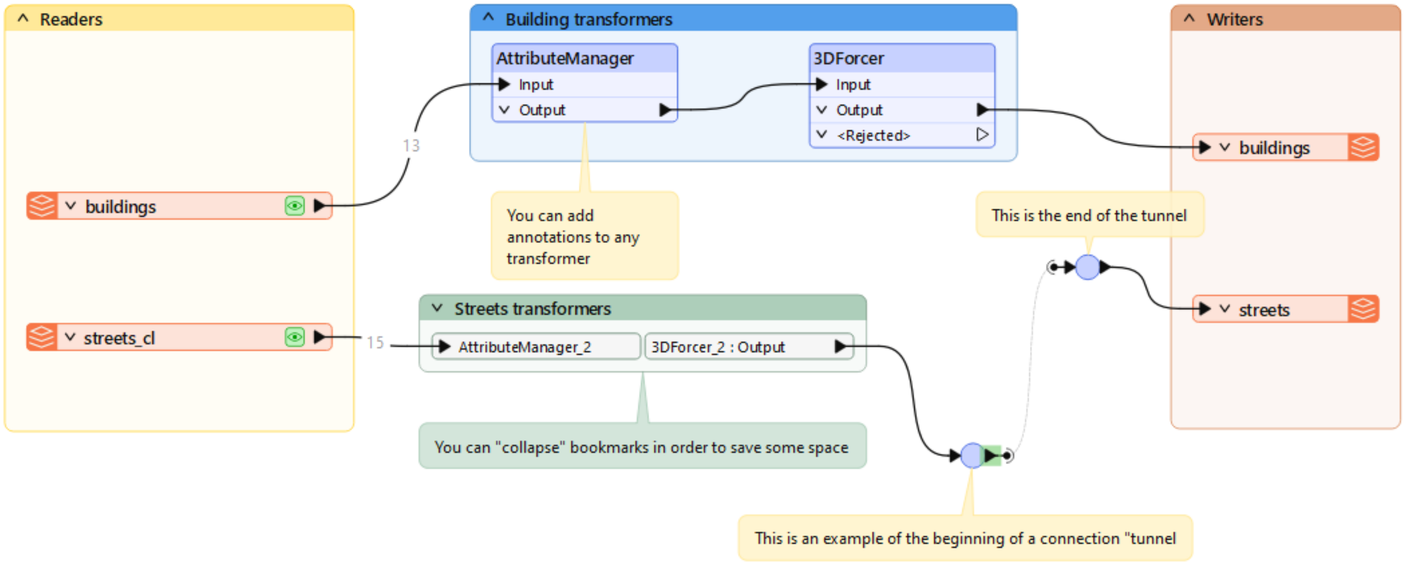
Common transformers:

- AttributeCreator, AttributeRemover, AttributeKeeper, AttributeRenamer
- AttributeManager
- Sorter
- AttributeFilter
- Tester
- FeatureMerger

- GeometryValidator, GeometryRemover
- GeometryCoercer
- GeometryFilter
- 2DForcer, 3DForcer, Extruder
- FaceReplacer, Orientor
- GeometryPropertySetter, GeometryPropertyRemover

Other goodies

- You can group transformers using **Bookmarks**
- You can attach **annotations** to each transformer, and to bookmarks, too
- You can change the **shape of the connection lines**
- You can create **"tunnels"** in the connections
- …and much more!



Dr. Giorgio Agugiaro

g.agugiaro@tudelft.nl

3D Geoinformation Group

TU Delft

The Netherlands

https://3d.bk.tudelft.nl/gagugiaro