



UVM
SYSTEMS

CityGRID[®] Reader/Writer for FME

CityGRID[®] Reader/Writer documentation

contents

Installation	3
Structure overview	4
Parameters of the reader	5
Read CityGRID® Models	5
LoD (only DB)	5
Historic Versions at Date (dd.mm.yyyy, only DB)	5
Building Part IDs	5
Read CityGRID® Images	5
Read Image Raster Content	6
Texture Resolution (in mm, only DB)	6
Parameters of the writer	7
Generate Faces	7
Check in new versions of Units:	7
Create a new version for an unchanged image:	7
Create a new version for an unchanged terrain:	7
Attributes of the feature types	8
General:	8
semantics	9
Unit	9
Element:	9
Geometric feature types (edgelines) (Linien)	10
Surface	10
CityGRID Models and their referenced datasets	11
Model	11
Image	12
Terrain	12
Tips	13
Task:	13

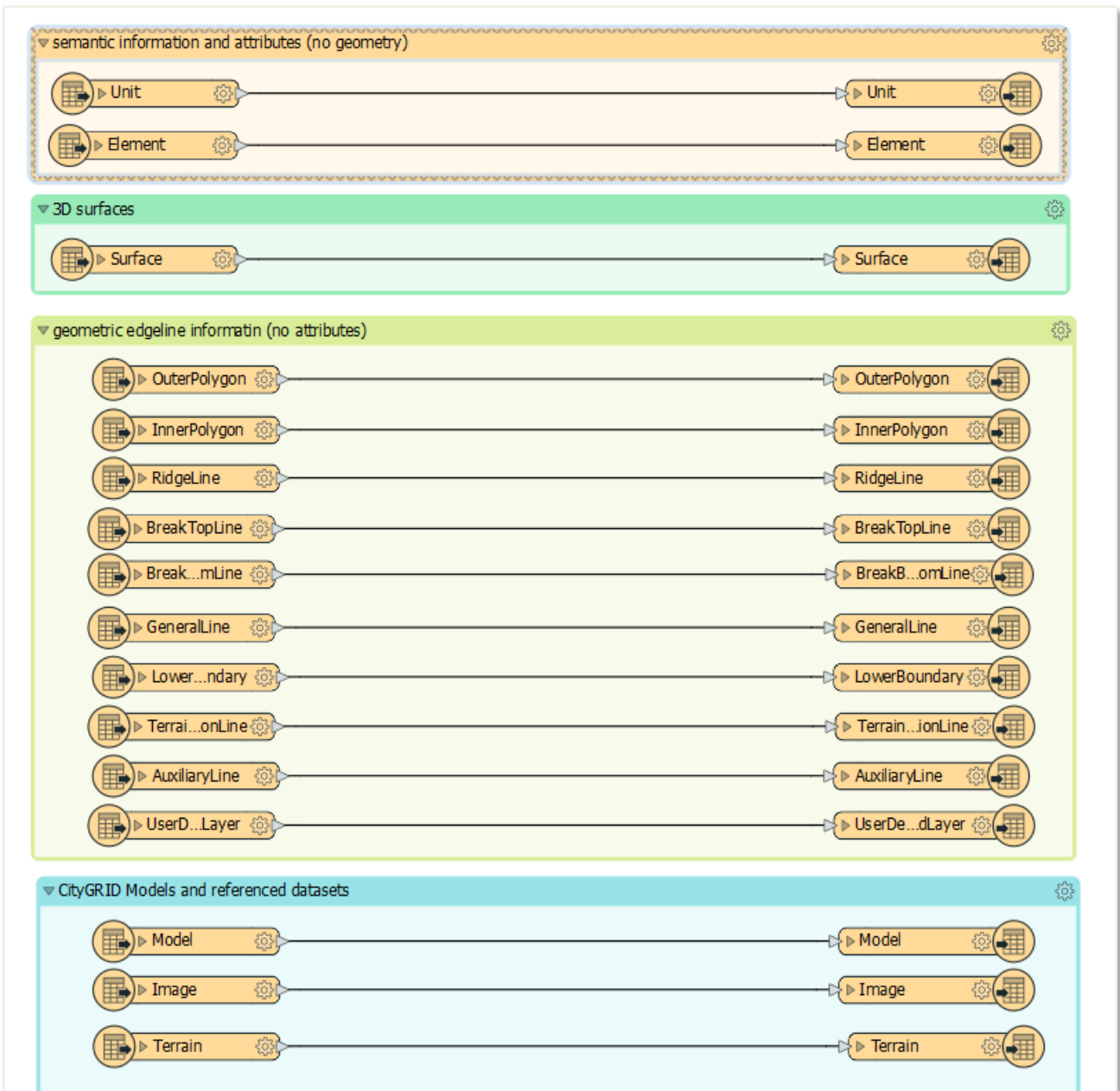
Installation

The CityGRID® Reader/Writer part of the CityGRID® Setups. If an FME installation already exists on the computer before CityGRID® is installed, the files of the reader/writer are written to the corresponding FME directories, when CityGRID® setup is run. If FME was installed afterwards, it is necessary to run CityGRID® setup again.

Please note that CityGRID® is only available as a 64-bit version since version 2019, build 13.50.007, which means that FME must also be available as a 64-bit version. Please also note, that the FME versions supported by your CityGRID® license, which can be found in the release notes of your version, for example: https://downloads.uvmsystems.com/Citygrid/Release_notes_en_1400011.pdf

Structure overview

The structure is based on that of a relational database schema, the data packets are available in granular form, with as far as possible stores information redundancy free.



Parameters of the reader

CityGRID City Model Format Parameters

Read CityGRID Models

Model Names (separate names by \';\');:

LoD (only DB): 3-Detail

Historic Versions at Date (dd.mm.yyyy, only DB):

Building Part IDs: globally-unique

Read CityGRID Images

Image Names (separate names by \';' or use \'_ALL_\');:

Read Image Raster Content:

Read CityGRID Terrains

Terrain Names (separate names by \';' or use \'_ALL_\');:

Texture Resolution (in mm, only DB): No

Presets

OK Cancel

Read CityGRID® Models

Yes/No. By entering one or more model names when reading a database connection, you can specify which models should be read. When reading a CityGRID® XML, only the checkbox must be set, since only one model can exist within an XML file.

LoD (only DB)

In a CityGRID® database, the lower LoD levels are automatically derived from a LoD3 object. This parameter can be used to specify which LoD state of the building is to be loaded. (LoD3 default)

Historic Versions at Date (dd.mm.yyyy, only DB)

Historical versions of buildings can be kept in a CityGRID® database. With this optional parameter, a historical version can be loaded from the database.

Building Part IDs

This parameter can be used to determine whether the Building Part ID is globally unique (within the entire database / XML file) or locally unique (within a unit). Default: globally unique. For the Building Part ID see below.

Read CityGRID® Images

Yes/No; By specifying one or more Image Names it can be determined which or how many aerial images should be read.

Read Image Raster Content

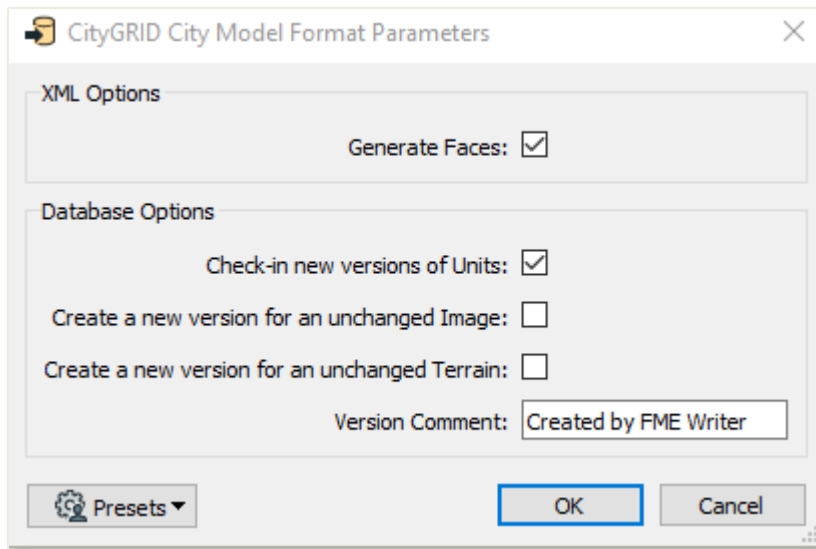
By activating this checkbox, not only the meta information and the reference of the image is loaded, but also the raster file itself. This checkbox should only be set if the raster information is relevant for the workspace or the intended data transformation, due to the usually very large images and the associated loss of performance.

Texture Resolution (in mm, only DB)

This value determines whether and if so, in what resolution the building textures are loaded from the database. The textures from an XML file are automatically loaded in the highest resolution, unless "No" is set. Default: No

- No
- Lowest
- Highest
- 1
- 2
- 5
- 10
- 20
- 50
- 100

Parameters of the writer



Generate Faces

Yes/No. Defines whether the units are triangulated or not. Affects the performance of the workspace, depending on the number of units to be triangulated, the process can take some time.

Check in new versions of Units:

Yes/No. Defines whether all units should be checked in as new versions (DB)

Create a new version for an unchanged image:

Yes/No. Defines whether an unchanged image should be saved as a new version (DB) (default no).

Create a new version for an unchanged terrain:

Yes/No. Defines whether an unchanged terrain should be saved as a new version (DB) (default no).

Attributes of the feature types

General:

The most important attribute of the CityGRID® feature types is the Building Part ID *citygrid_BuildingXx_id*. Each feature type of the hierarchy levels unit, object, element complex and element have an attribute *citygrid_BuildingXx_id*. This indicates the hierarchical superordinate feature to which they belong

Each **Edgeline** feature has a *citygrid_BuildingElement_id*, which states which element the Edgeline is assigned to (roof, facade floor), which is why the Edgeline feature types are generally held in the reader as a collective feature type. An edgeline of the "outer polygon" type is assigned to an element and thus a function based on its join attributes (*citygrid_BuildingElement_id*). The assigned element ultimately determines whether it is an Upper Facade Polygon or an Outer Boundary: Eave.

Each **element** has a *citygrid_BuildingElementComplex_id* and a *citygrid_BuildingObject_id* that indicate the relationship between the element, element complex, and object.

These IDs are inherited from "top to bottom", which means that the eaves line of the roof element, of a detail element complex of a main roof, of one of two objects of a unit would have following IDs (from top to bottom):

<i>citygrid_BuildingUnit_id</i>	for the Unit
<i>citygrid_BuildingObject_id</i>	for the Object within the Unit
<i>citygrid_BuildingElementComplex_id</i>	for the Element Complex within the Object.
<i>citygrid_BuildingParentElement_id</i>	for the Parent roof Element (because it's a detail Element Complex)
<i>citygrid_BuildingElement_id</i>	for the Element (roof/façade/floor)

This ID can be unique either globally (in the entire file / database) or locally (within the unit), but in general it is used as globally unique, which means that there is no need to check for uniqueness.

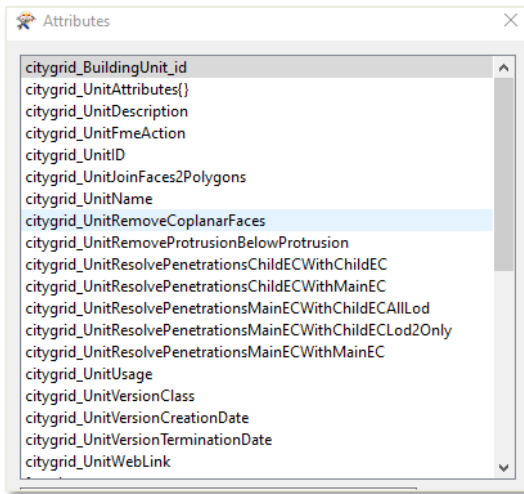
It can be used to:

- Change hierarchies (e.g. a unit becomes the object of another unit by changing its *citygrid_BuildingUnit_id*),
- Move features within a workspace,
- change the dependencies. Thus, said detail element complex can be assigned to another parent element by controlling this ID, which among other things can affect the triangulation behaviour, for example.
- As a Join attribute (the *citygrid_BuildingElement_id*), to inherit element attributes to the geometry (using the FeatureMerger Transformer)

The Building Part ID is not visible in the XML because it is created by the reader, recognized by the writer and used to carry out the hierarchical assignment. After this assignment, the ID is resolved and NOT written out in the file.

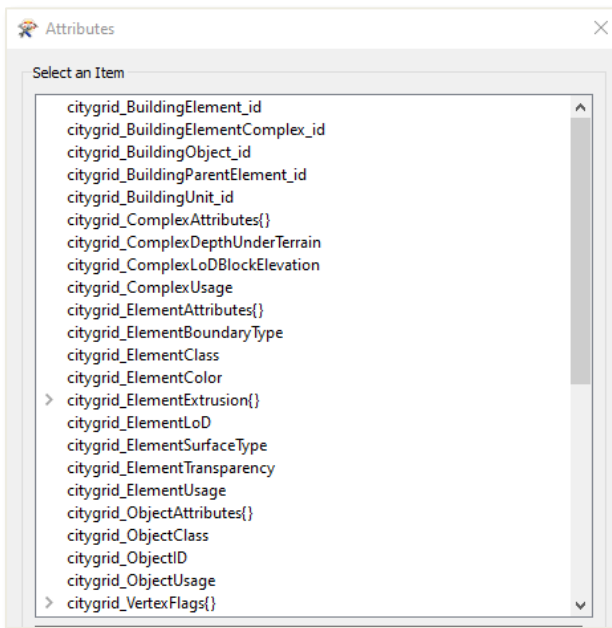
semantics

Unit



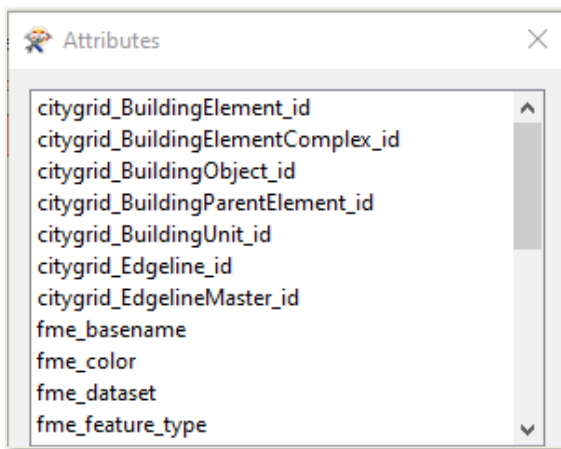
The attributes of the unit in FME include a building part ID, unit properties known from CityGRID® applications (such as the unique Unit_ID, usage, description, etc.), a list of generic unit attributes (citygrid_UnitAttributes {}), also attributes for controlling the penetration resolution and the behaviour of the face generation within the unit.

Element:



The element, on the other hand, is the main carrier of semantic information within CityGRID®. Here the attributes of the object, element complex and element are defined, with the Building Part IDs of all hierarchy levels (description see above) as well as the behaviour of the area formation (such as extrusion specifically its type, direction and length, transparency, etc.) are controlled.

Geometric feature types (edge lines)

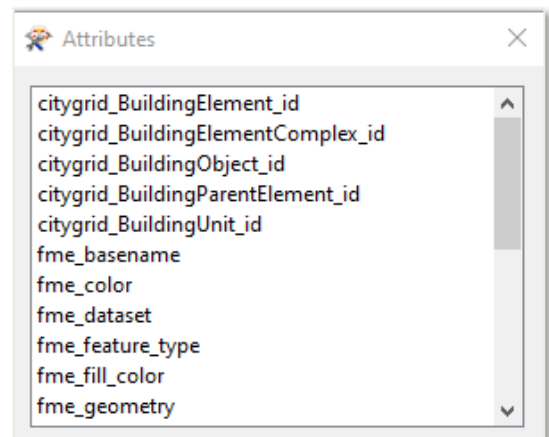


The geometry-bearing feature types of the CityGRID® Reader have no semantic information with the exception of the citygrid_BuildingXx_ids (Building Part IDs), which assign the line to an element, element complex, object and a unit. The Edgeline_id and the EdgelineMaster_id is used to define interdependencies between lines (e.g. eaves facade, upper and lower fault lines, etc.)

Surface

The 3D surfaces are derivatives of the CityGRID® lines and therefore have the same information as the line feature types, with the exception of the citygrid_Edgeline_id, as well as the citygrid_EdgelineMaster_id.

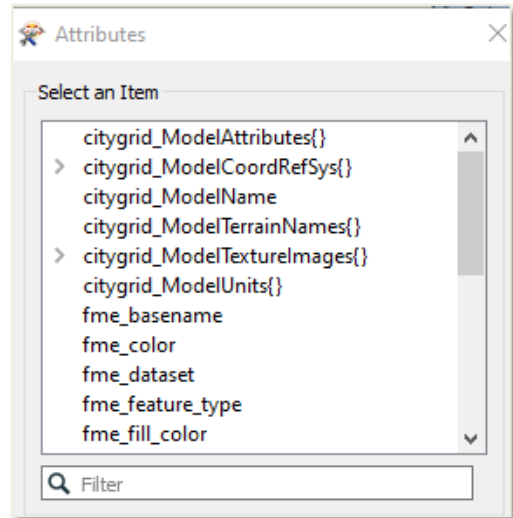
The other IDs (Building Part IDs) indicate the hierarchical position of the 3D areas within the CityGRID® structure an. (see description on page 8)



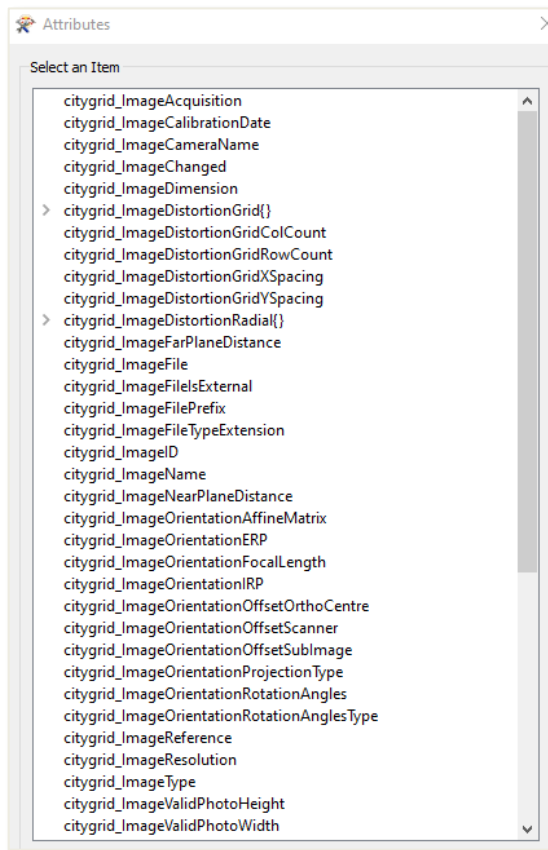
CityGRID Models and their referenced datasets

Model

In contrast to the feature types mentioned so far, the model does not have a `citygrid_BuildingXx_id`, as it is known that the model is not part of a defined hierarchy within CityGRID®. Units can be part of different models, removed from models or added without any changes within the unit. For this reason, the model attributes are limited to general properties that all units within a model must have in common, such as generic model attributes, the coordinate system, DTM(s) referenced in the model, texture images referenced in the model, and a list of all units within this model.

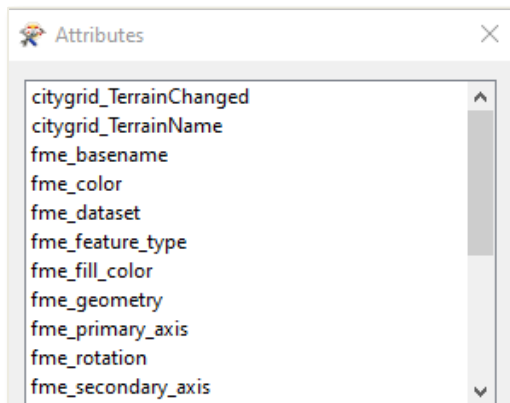


Image



Depending on the role of the image in the database and on the type of orientation parameters of the image, the Image Feature Type has attributes that can contain information on camera calibration, internal and external orientation of the image on the one hand, as well as resolution, meta information about the file type of the image on the other hand.

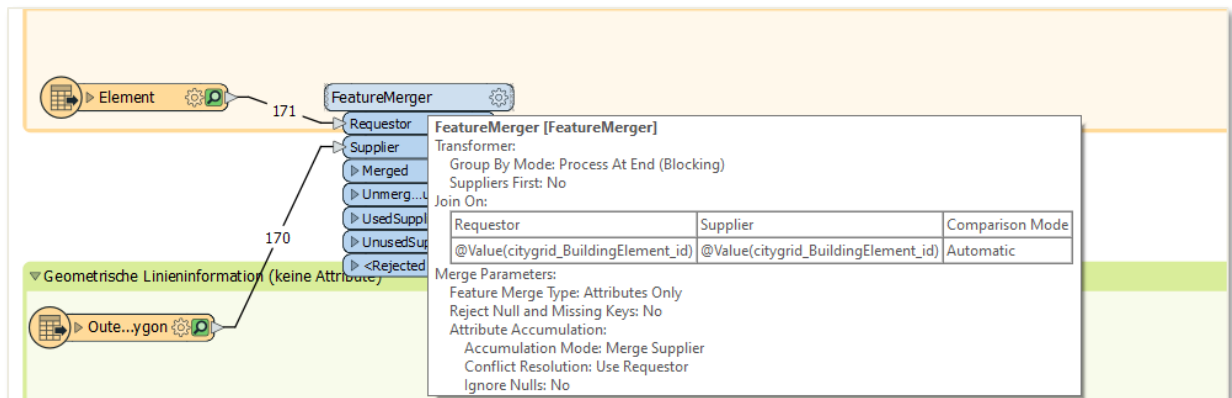
Terrain



The terrain has only two attributes that can be used for referencing within a model and the associated geometric manipulations of the units based on the DTM.

Tips

For almost all data transformation processes with the CityGRID® reader/writer, the geometry must be recognizable on the basis of the attributes, for example it must be possible to filter main roof trusses, detail element complexes, etc. Since this is not possible via the geometry alone, it is recommended as a first step to always inherit the attributes of the "element" feature type with a FeatureMerger Transformer to the desired geometry feature type.



Task:

To test your knowledge of the CityGRID® Reader/Writer you can create a simple exercise workspace with a CityGRID® demo data set. Please note that you can only use the Reader / Writer if you have a licensed CityGRID® version.

The requirement:

Using the correct Building Part ID, find the outer boundary line of the roof elements of the detail element complexes and write out the result as a .shp file.

Note: There are 66 detail element complexes in the demo data set.