URBAN VISUALISATION & MANAGEMENT GMBH

UVM SYSTEMS

CityGRID® Manager MANUAL

2024

Contents

Cover picture: Prague, Leonhard Niederwimmer, Pixabay

Note: This manual was produced by automatic translation. Strange expressions and sentence structure may therefore occur. In case of doubt, the German manual is to be used as a reference.

1 Basics

The Manager is a stand-alone module of the City**GRID®** Software Suite. With an existing database connection, all data in this database can be graphically visualised, selected and exported in all data formats offered by City**GRID®**.
The Manager is a stand-alone module of the City**GRID®** Software Suite. With an existing database connection, all data in this database can be graphically visualised, selected and exported in all data formats offered by City**GRID®**.

1.1. System requirements

To operate the des City**GRID®** Manager, a standard computer with 32 GB RAM and a 3D-capable graphics card is sufficient. To be able to display data in the manager, a City**GRID®** database must be available, therefore also an MSSQL or Oracle installation.

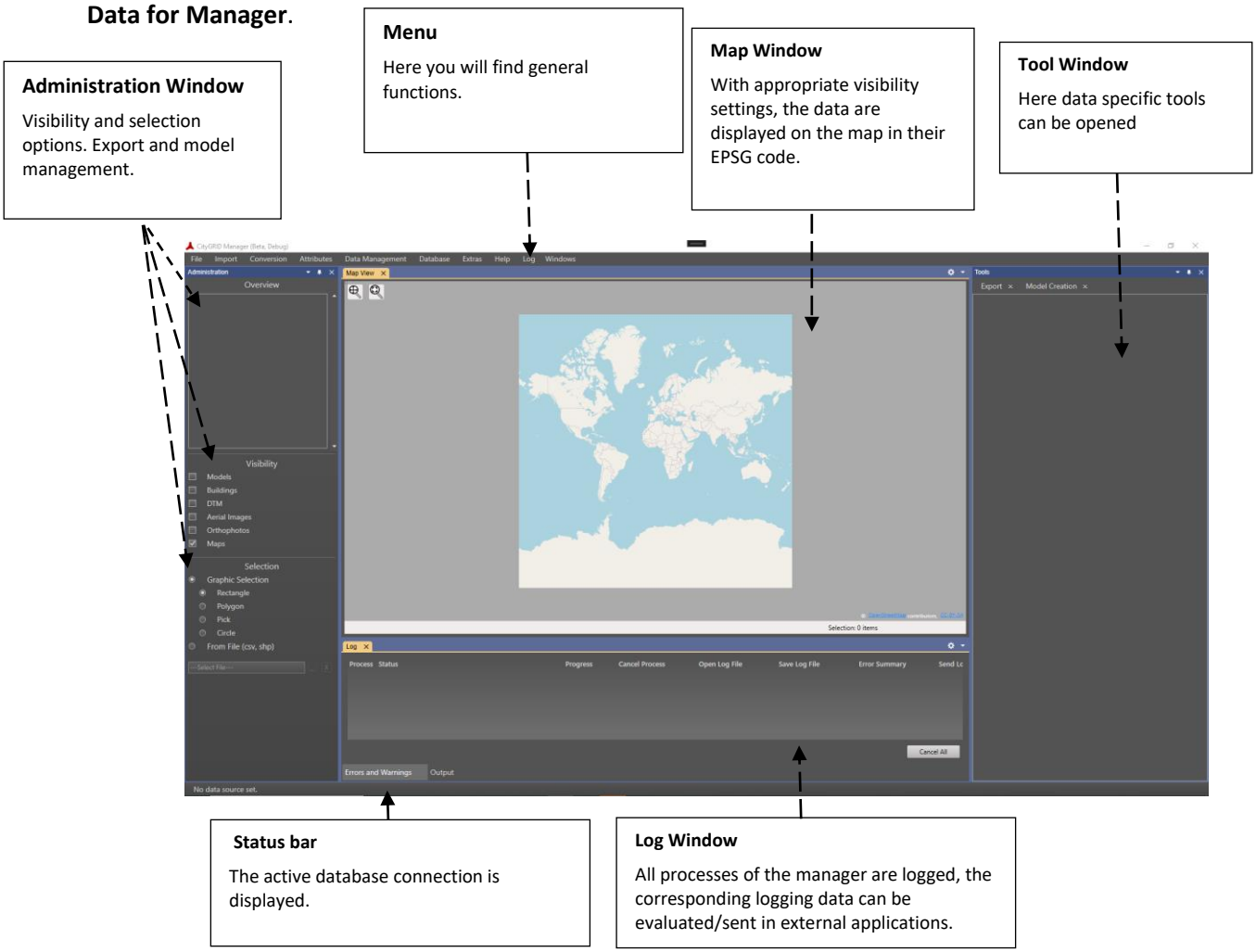
1.2. *Data prerequisites*

To be able to display data, a database connection must be specified under **File > Set Datasource**. If data is held in City**GRID®** before the manager is used for the first time, the data must first be prepared for the manager under **Extras > Prepare Data for Manager**.

*Note: Manager requires a known CRS. Since this has not been mandatory in in City**GRID®** so far, it is possible to enter an EPSG code during data preparation.*

2. The User Interface

After starting the manager, an initially empty user interface appears. To be able to see the full scope of the manager, a database connection must be created to then be able to display all data of this database. If the data do not yet have an EPSG code, they must first be prepared via **Extras > Prepare Data for Manager**.



The main window contains the central control mechanisms for the following tasks:

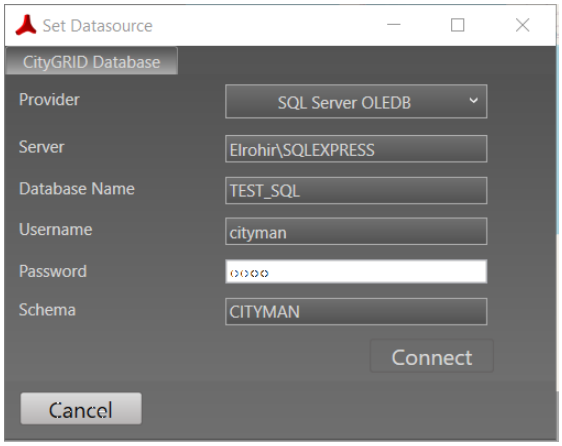
- Setting the selected data/models
- Visibility control of the data
- Selection methods (graphically or by file)
- Export and model creation of selected data

2.1. Menu

2.1.1. File

2.1.1.1. Set Data source.

Opens the dialogue for setting a database connection (City**GRID®**).



2.1.1.2. Stored Database Connections

Shows a list of the database connection already used and enables a quick connection.

2.1.1.3. Exit

Exits Manager

2.1.2. Extras

2.1.2.1. Prepare Data for Manager

Opens the dialogue for data preparation with an integrated database. The name of the model, the EPSG code already stored, and the status of the model are displayed. Data are only **`ready`** if an EPSG code is available. If it is not already available in the data, it must be entered manually by clicking in the text field. After entering an EPSG code, the data can be previewed using the **`Preview`** button in the map window to verify that it is the correct EPSG code. If the preview is correct, the data can be calculated with the new EPSG code using the **`Process`** button.

2.1.3. Help

2.1.3.1. Manual

Opens the City**GRID®** Manager Manual.

2.1.3.2. About

Shows the product and licence details of City**GRID®** Manager.

2.1.4. Log

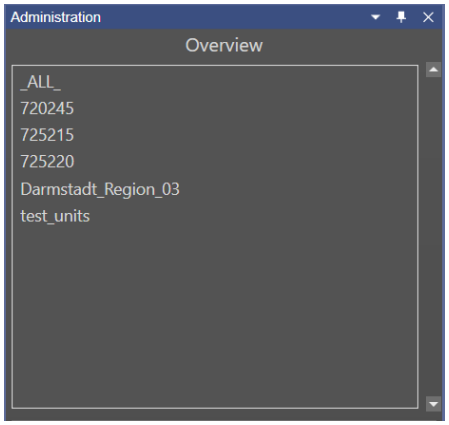
2.1.4.1. Open Log Directory

Opens the Log Directory.

2.1.4.2. Comprehensive Logging

In case of malfunctions in Manager, this option can be activated to write extended information into the log file to facilitate troubleshooting.

2.2. The Administration Window



In the administration window, all models of the active database connection are displayed in the ***Overview***. You can select one, several (by holding down the *Ctrl* key) or all models (_ALL_). RMC opens the context menu and allows the user to open tools to interact with the data. (i.e. Exporting selected Models, Creating new Models)

2.2.1. Visibility

Depending on the existing data in the models, these can be visualised in the map window. The checkboxes in the ***Visibility*** section control the visibility of the data classes. If a data class is activated, representative bounding boxes of the geometries are loaded, except for the OpenStreetMap map. What is not visible cannot be selected. If a data class with already selected items is switched to invisible, however, the selection remains until the selection is reset with *ESC*.

2.2.2. Selection

Visible data classes can be selected graphically across one or more models.

2.2.2.1. Graphic Selection

- **Rectangle**

The ***first mouse click*** in the map defines the starting point, ***the second*** the direction and the extension. To the right (blue), a rectangle is drawn that selects data that lie completely within it. To the left (green) a rectangle is drawn that selects data touched by it. ***Right click*** cancels. *Ctrl* selects and *Alt* deselects.

- **Polygon**

  By ***at least three mouse clicks***, a polygon is drawn in the map window, enabling a more precise selection. ***The second mouse click*** defines the direction, as with the rectangle. A ***right click*** cancels the polygon, a ***double click*** closes the polygon.

- **Pick**

  A ***single mouse click*** selects data touched by the mouse click. If several data (classes) are on top of each other - for example aerial photos over buildings over terrains, the "top" data set is always selected first. With ***another click*** the next lower data set can be selected, with a ***right click*** the other direction is selected.

- **Circle**

  Works analogously to rectangle selection, ***the first mouse click*** specifies the centre of a circle, ***the second*** the direction and the diameter. Depending on the direction, a circle is drawn that either selects everything it touches or only data that lies completely within it.

2.2.2.2. From File (csv, shp)

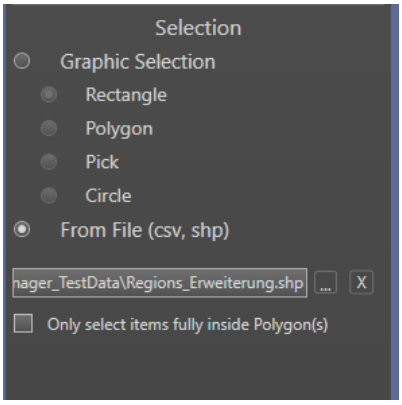
- **csv**:

  Allows the import of ***a list*** with UnitIDs via which a selection is made. The csv file must not have a header.

- **shp**:

  Allows the import of a shapefile with ***one or more polygons*** that selects data that lies within. When a shapefile is selected, the option appears to select either everything within the polygon(s) or, everything that is touched (default). The `X` **`Clear`** button deletes the loaded file from the file browser and resets the selection.

  When a file is loaded, further features can be added or removed from the selection when switching to the graphical selection by holding down the *Ctrl* or *Alt* key.



2.2.2.3. Model Creation

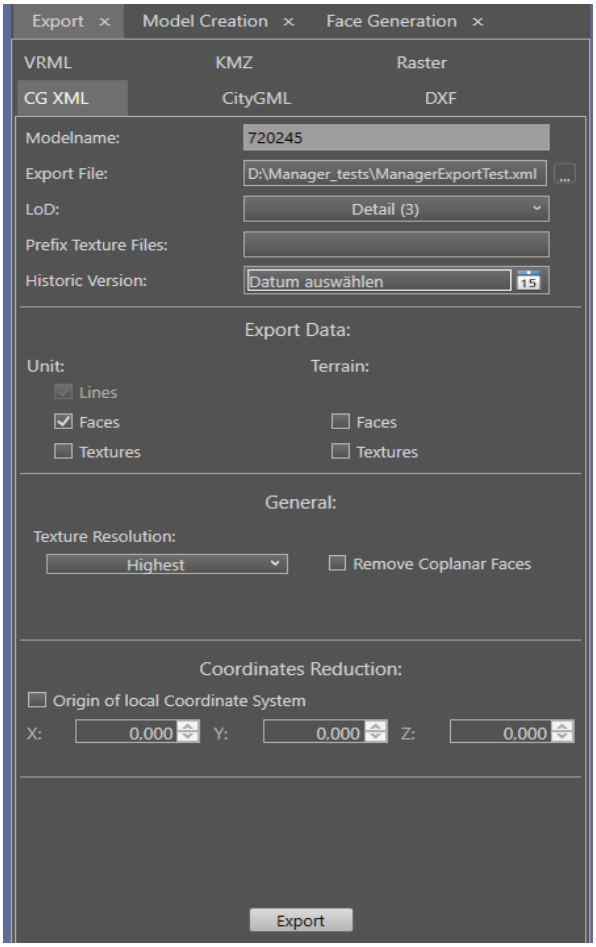
The **Model Creation** option becomes active after a selection has been made via one of the selection methods mentioned above. Clicking the button opens a dialogue that gives an overview of the selected database content.

A new model name for the selected items must be entered in the text field *Model Name*. The **Save** button saves the new model to the database and writes it as a new entry in the model list. More about City**GRID®** models in the Basics manual p. 11

2.2.3. Export

The Manager Export offers all export options already known from the Administrator: *VRML, DXF, XML, CityGML and KMZ*. Entire models, parts of a model and several models can be exported.

Note: The raster export is not available via a selection. At least 1 model must be selected for this.



To export in the desired format, switch to the corresponding tab in the export dialogue.

The export form opens in which the export parameters can be specified. By default, the export always accesses the last stable version of a unit, if no historical version is requested for a specific date.

The VRML export creates a WRL file of the selected model in the VRML 97 specification. Any texture images are stored in the folder "images" at the same path as the WRL file.

The DXF export generates either an ASCII or binary DXF format. During the export, all lines and surfaces of the units are separated according to their layers. Textures are not considered.

The City**GRID®** XML Export generates an XML file in the proprietary City**GRID®** XML format.

The CityGML export generates an OGC-compliant GML file that can contain all lines, faces, and textures of the model, as well as the hierarchical building structure and the possibly existing terrain model as LoD 2 TIN.

Using the KMZ export, City**GRID®** creates a KML file with an associated "link" directory in which each unit is saved in the form of a KMZ (zipped Collada file with texture directory).

Note: If several models are written via the KMZ export, they share the "link" folder.

2.2.3.1. Shared Export parameters

- *Model name*: (*CityGRID XML*)

  Depending on whether a selection, a complete model or several models are to be exported, a model name must be specified. If one or more complete models are to be exported, the model's name is taken from the original model(s)).

- *Export file: (XML, CityGML, dxf, vrml, raster)*

  The directory in which the export file will be saved.

- *Export Folder* (CityGRID XML, CityGML)

  If several models are to be exported, an export directory must be specified in which the models are to be saved.

Note: If the data is textured, it is helpful to create subfolders with the checkbox Create Subfolders.

- *LoD:*

  The generalisation level of the data to be exported can be defined here. DETAIL (LoD 3) corresponds to the highest possible level, BLOCK (LoD 1) to the lowest.

- *Historic Version:*

  Controls whether the export should be applied to historical versions. If this option is selected, data is exported from the database at the specified time, regardless of whether the version at the specified export date is the current one or not.

- *Export File:*

  Enter the path to the export file. The file name is always derived from the archived model. Optionally, the path can be selected with a file browser by means of buttons.

- *Prefix Texture files*:

  Places the specified string in front of the texture names of the exported texture images. If no value is given, the images receive the image name assigned by the database.

Note: The use of the prefix is necessary when using several databases, if exports from these databases are merged in another database (e.g. CityDB).

- *Create Subfolders:*

  Via the checkbox, each model can be stored in its own subdirectory of the export path. The directory name is derived from the model's name.

Note: The division into subdirectories should be done when exporting textured models if the texture images are to remain individually addressable for each model.

- *Coordinates reduction*:

  Origin of the local coordinate system: Defines the origin of the coordinate system to be used for the export. When the checkbox is unchecked, the coordinate system of the input data is used, otherwise it is replaced by a local coordinate system. Manual City**GRID®** Administrator - Page 21

Note: By using this parameter, VRML files can be transformed into a local coordinate system during export, so that commercially available viewers can display the data without any problems. VRMLs with national coordinates, on the other hand, can often only be displayed to a limited extent because the coordinate ranges are generally too large. When using an alternative coordinate origin, it is recommended to include the displacement values as part of the file name.

Export Data:

Depending on the export format, different geometries are available:

❖ Unit Data:

- Lines (CityGRID xml and DXF)
- Faces (CityGRID xml, CityGML, DXF, VRML, KMZ)
- Textures (CityGRID xml, CityGML, VRML, KMZ)

❖ Terrain:

- Faces (CityGRID xml, CityGML, DXF, VRML)
- Textures Faces (CityGRID xml, CityGML, VRML)

General:

- *Texture Resolution*: (CityGRID xml, CityGML, VRML, KMZ)

  Specify the quality of the exported texture images. The numerical value indicates the pixel size in mm. *Highest* is specified as the default value.

- *Unit*→ Layer: (DXF)

  Activate the checkbox when exporting to DXF if each unit is to be placed on a separate layer with the UnitID as the layer name.

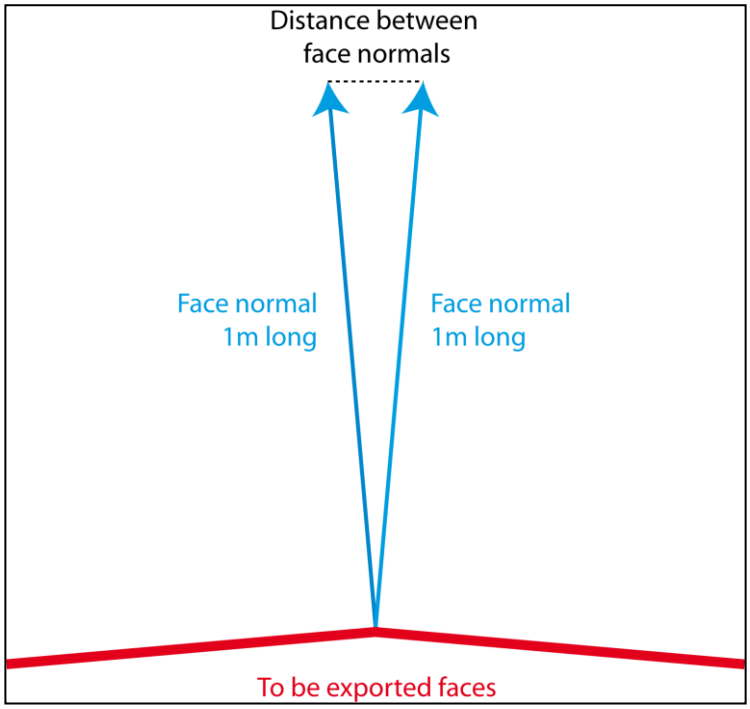
- *Remove Coplanar Faces:* (CityGRID xml, CityGML, DXF, VRML, KMZ)

  Eliminates all congruent façade surfaces within a unit. This results in buildings with a coherent interior space without partition walls, regardless of how many complexes make up the unit in City**GRID®**.

- *Join Polygons*: (CityGML, DXF, VRML, KMZ)

  Controls up to which cant polygons are considered planar and exported as one surface. The tolerance specifies the distance between adjacent surface normals with length one metre (unit vectors) in metres. For small angle values this corresponds to the angle between the surface normals in radians.

Example: A value of 0.01 corresponds to a deviation of 1cm of the one-metre-long surface normal, or an angle of 0.01 π. Converted into degrees, this results in a value of 1.8°.



2.2.3.2. Advanced export settings for CityGML

As an exchange format for city model data, CityGML offers a range of parameters compared to VRML and DXF to export building models according to individual specifications. Parameters such as CityGML version, attribute integration and semantic transformations of the buildings to be exported can be defined in a separate settings window.

In the export window for CityGML the button Advanced... is available to open the window for the export parameters.

With every City**GRID®** installation a standard set of parameters is delivered, named Default. Changes to the default export parameters are only applied if a new parameter set is saved when exiting the window. Changes to the default parameter set are not possible. In addition to the default parameter set, however, any number of data sets created by the user are possible.

*Note: All parameter sets are stored in the citygmlexport.ini file in the City**GRID®** protocol directory. In the event of support cases, please send this file to UVM Systems.*

- *Active Parameter set:*

  Calls up the saved export settings. After the initial installation, only the default parameter set is available.

- *CityGML Version*:

  Defines in which version of the CityGML standard the export should be written.

Export Data

Controls which geometry objects are written during export.

- *Thematic Surfaces.*:

  Generates thematic surfaces from the City**GRID®** elements for each building. (Roof to RoofSurface, façade to WallSurface, etc.).

- *Solid*:

  Combines all City**GRID**® elements into one surface model per building.

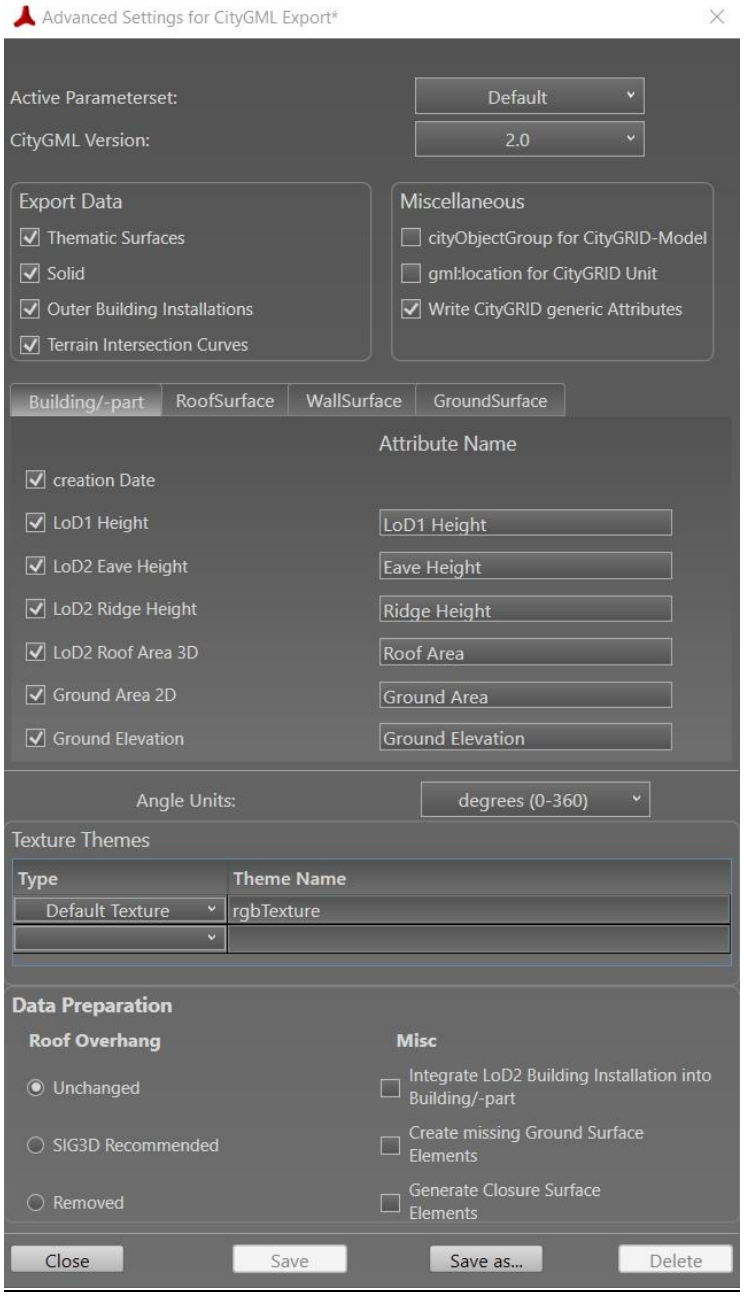
- *Outer Building Installations*:

  Includes all City**GRID**® Detail element complexes in the export.

- *Terrain Intersection Curves*:

  Writes the terrain intersection lines of the buildings into the export.

Note: To see building data in the CityGML export, either Thematic Surfaces or Solid must be checked.



Miscellaneous

summarises parameters that cannot be explicitly assigned to any group.

- *cityObjectGroup for:*

  Generates an independent cityObjectGroup in which the model's name of the City**GRID**® model is entered.

- *gml:location for Units:*

  Writes a representative insertion point of the unit into the CityGML file.

Write CityGRIDs generic. Attribute:

Exports all existing attributes from the City**GRID**® dataset.

*Note: Attributes for model, units, object, element complex and element can be managed with City**GRID**®).*

Automatically generated attributes

offers the possibility to derive certain attributes from the geometry data. The attributes are added to the specified complexes. All calculated values, except for *base height (NN*), are given in relative building height. The attribute name can be defined by the user. Attributes that do not make sense for certain complexes are greyed out by default.

- *Creation Date*:

  Writes the current date into the CityGML file when exporting.

Note: Creation date therefore refers to the creation date of the GML file and not the generation of the model itself.

- *LoD1 Height*:

  Calculates the mean of the highest and lowest point of the eave's polygon.

- *LoD2 Eave Height*:

  Gives the lowest point of the eave's polygon.

Note: If the parameter "Roof overhangs according to German SIG3D ("AG Qualität") recommendation" is set, the lowest point of the façade upper edge polygon is used here.

- *LoD2 Ridge Height*:

  Delivers the highest building point as attribute value.

- *LoD2 Roof Area 3D*:

  Outputs the sum of the individual roof areas in square metres for each building or building part. For the thematic surfaces, the surface area always refers to the respective polygon.

*Note: The calculation uses every LoD2 surface. Thus, a face under roof details is also included if the face generation in City**GRID**® has not been adjusted accordingly (extrusion type "Extrude to parent roof with hole", or via the penetration resolution, see City**GRID**® Modeler manual).*

- *Ground Area 2D (Ground Surfaces*:

  Indicates the enclosed area of the building in square metres. If floor polygons are present, their area is calculated. Otherwise, the upper façade polygon serves as a reference. Any existing inner courtyards are considered in the calculation. If there is also no upper edge of the façade, all areas with the surface normal pointing upwards are used.

- *Ground Elevation (NN):*

  Indicates the lowest point of the building at sea level (based on the coordinate system used).

- *Steepness (Roof Surfaces)*:

  Indicates the slope in degrees for each polygon.

- *Exposition (Roof Surfaces Wall Surfaces)*:

  Indicates the rotation to the north direction of surfaces. Positive values rotate from north via east to south, negative via west. Horizontal surfaces have no orientation set.

- *Angle Units*:

  Determines whether slope and orientation are written in old degrees (0-360) or new degrees (0-400).

Tip: If generic attributes also have angle values, it is advisable to choose the same units for the automatically generated attributes.

Textur Themes

allows you to define which topics are written into the CityGML file.

- *Type*:

  Offers the choice between the standard colour, the standard texture, and any themes. Standard colours are the set City**GRID®** face colours, standard texture all textures applied so far and theme name is a texture theme specified by the user (e.g. solar potential).

*Note: Offers the choice between the standard colour, the standard texture, and any themes. Standard colours are the set City**GRID®** surface colours, standard texture all textures applied so far and theme name is a texture theme specified by the user (e.g. solar potential)..*

*Note: Texture themes are not supported in City**GRID®** at present.*

- *Theme name*:

  Carries the name of the texture theme that was specified during automatic texturing.

Data preparation

offers the possibility to influence the geometry or semantics of the building models before they go through the CityGML export.

*Note: Each data pre-processing option modifies the building geometry during the export process, which on the one hand leads to the fact that the result can (strongly) deviate from the City**GRID®** dataset and on the other hand increases the susceptibility to geometry errors, which can occur due to the fully automatic processing. The options should therefore only be activated if necessary.*

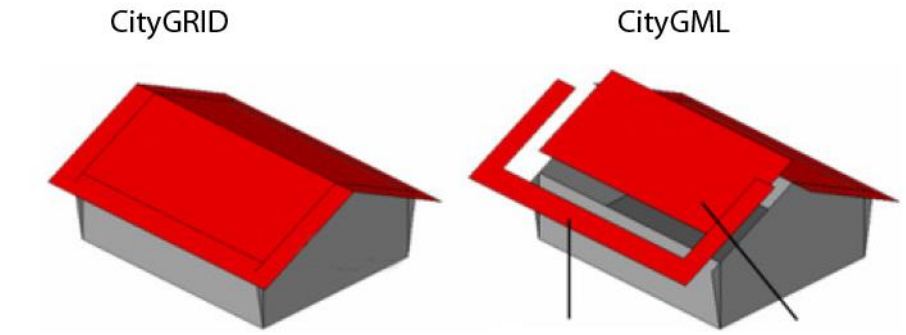
- *Roof Overhang*
  - *Unchanged:*

    Leaves the roof overhangs analogous to the situation in City**GRID®**. The roof surfaces remain geometrically identical to the original dataset, (possibly combined into coplanar polygons if the corresponding general export option has been set, but the roof overhang element of the City**GRID®** dataset remains unconsidered, as there is no equivalent for it in the CityGML data model).

  - *SIG 3D Recommended:*

    Modifies existing roof overhangs in such a way that they become independent building parts. The existing roof overhang element in City**GRID®** is dissolved and the surfaces facing the ground are removed. The roof of the building begins at the

intersection between the façade and roof surfaces. The roof overhang joins directly to the roof element formed in this way but forms an independent building structure.



Roof with roof overhang *Roof overhang* *Roof Surface*

- *remove*

  modifies existing roof surfaces in such a way that only those parts of the roof are retained that can be located within the façade polygons. Roof surfaces and façade surfaces merge flush with each other.

- *Integrate LoD2 BuildingInstallations in Building/BuildingPart*:

  Analyses all detail element complexes for the set LoD. For LoD2, the CityGML export converts the City**GRID®** roof detail into CityGML Building Parts, exposes the roof areas under the complex and changes the building semantics accordingly. In CityGML the roof details then appear as components of the main roof.

- *Create missing GroundSurface-Elements*:

  City**GRID**® complexes generally do not have ground surfaces when the facades are extruded to the terrain model. However, if ground surfaces are desired during the CityGML export, they can be created during the export using this parameter.

- *Generate ClosureSurface-Elemente*:

  For coplanar surfaces CityGML offers the possibility to introduce so-called closure surfaces. The coplanar surfaces are detached from the thematic surfaces and stored as independent objects. If required, they can be displayed or deliberately omitted. Closure surfaces can be used to identify interior surfaces and create buildings with the greatest possible interior space.

If a parameter of the window is changed, the change must be saved in an existing parameter set (**`Save`** button) or a new data set must be created (**`Save as`**... button). The parameter set that was last used is always used during export.

2.2.3.3. Advanced export settings for KMZ

In the export window for KML, the Advanced... button is available to open the window for the export parameters. Here you can define the method of height reference and the underlying coordinate system of the data set.

Altitude mode

regulates how the exported units are to be placed on the Google Earth terrain.

Clamp to Ground:

Places the units over a central insertion point on the Google Earth terrain. This ensures that the buildings do not hover above the terrain. The Z-coordinate of the insertion point stored in the kml is ignored.

Absolute Z-Coordinates:

Uses the existing height values of the unit to convert them into the coordinate system of Google Earth. In this way, the buildings retain their actual 3D position, and thus their height accuracy. If the generalised terrain model of Google Earth deviates too much from the City**GRID®** model used, there is a risk that units will float or sink into the terrain.

Height Reference

controls whether the geoid or the reference ellipsoid of WGS84 is to be used as the height reference.

Geoid Heights (Default):

Recommended setting. The WGS 84 geoid is used as the height reference.

Ellipsoidal Heights (WGS84):

In rare applications, the reference ellipsoid of WGS 84 is used instead of the geoid for the transformation of the height values.

2.2.3.4. Export *Raster*

The Export Raster function is used to create greyscale terrain and surface raster from the vector City**GRID®** data.

Select the desired export settings.

LoD:

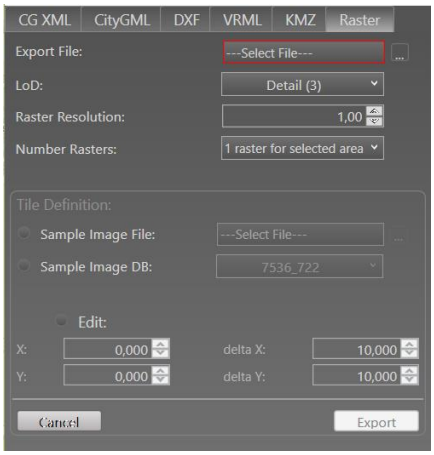
The generalisation level of the building data to be exported can be defined here. DETAIL (LoD 3) corresponds to the highest possible level, BLOCK (LoD 1) to the lowest.

Export File:

Specifies the path to the export file. The name of the export file corresponds to the model or terrain name by default. The formats jpg, png, tif and asc are available.

Raster-Resolution:

Determines the cell size of the formed raster image.



Number Rasters:

Determines whether all selected (terrain) models are to be combined into a surface grid or divided according to a tiling. If *tiled raster* is selected, the *Tile definition* section is activated.

Sample-Image File:

Offers the possibility to define a division grid via an image file with World file. The button ... opens a file browser in which a corresponding file can be selected.

Sample-Image DB:

Lists all orthoimages in the database. A division grid is formed via the stored orientation information.

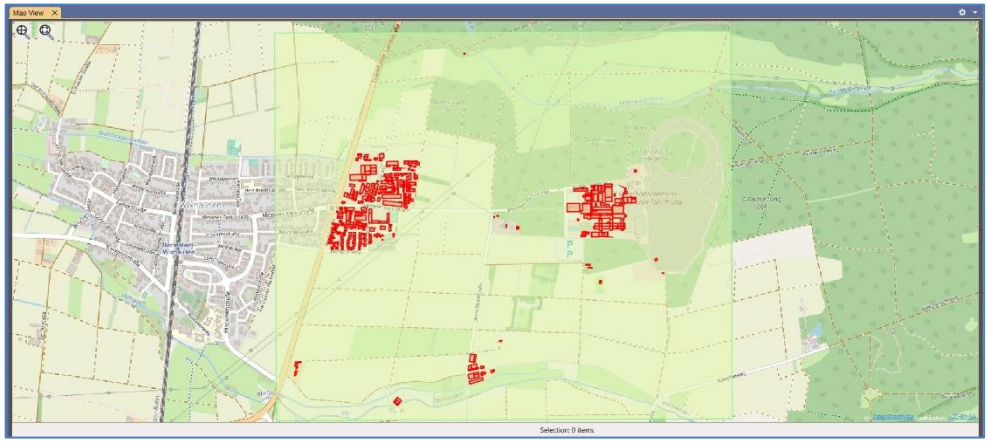
Edit:

Displays the division information, showing a grid interpolation point and the mesh size. If necessary, these parameters can be changed by the user.

Note: The tiles formed carry the name of the (terrain) model followed by consecutive numbers. The first number stands for the row number, the second for the column number.

Clicking on Export starts the export process. The button only becomes active when all necessary parameters for the export have been specified in the input mask. A description of the process control window can be found in the section.

2.3. The Map Window



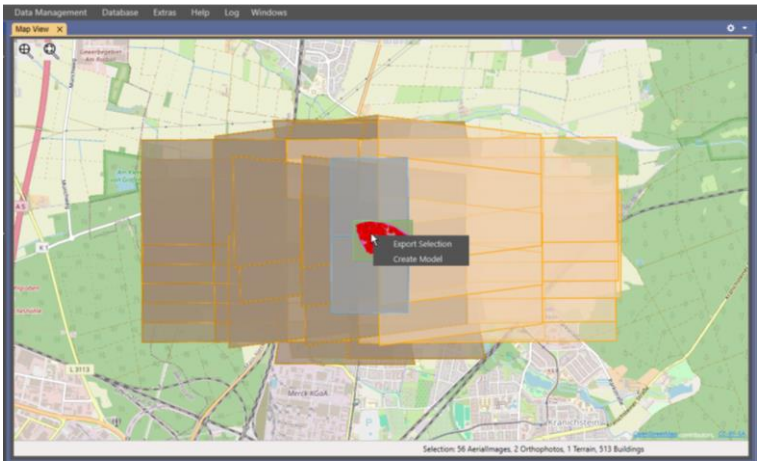
In the map window the content of the database is visualised, depending on the selected model and the visibility modes (see chapter 2.2.1.) Prerequisite for the display of the data is the specification of an EPSG code under *Extras > Prepare Data for Manager*.

The **`Zoom to Selection button`** zooms to the extent of an active selection.

The **`Zoom to Extents button`** zooms to the extent of the active model(s).

In the map window, a graphical selection can be made over the visible database content (Chapter 2.2.2.1), the selection status at the bottom of the map window shows the number of selected items. Selection: 1 Terrain, 37 Buildings

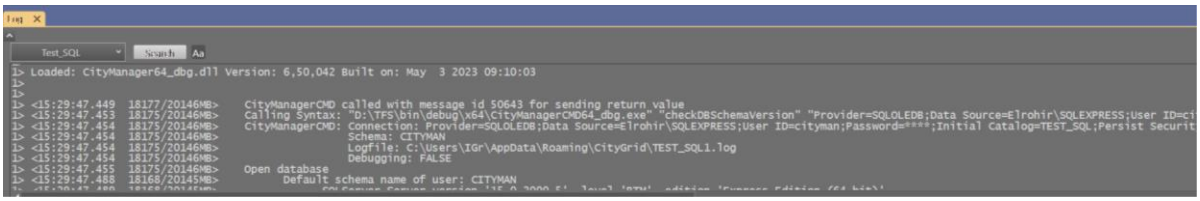
After a selection, the context menu can be opened with an RMK in the map window and the available tools for the selection can be opened (Export, Create Model).



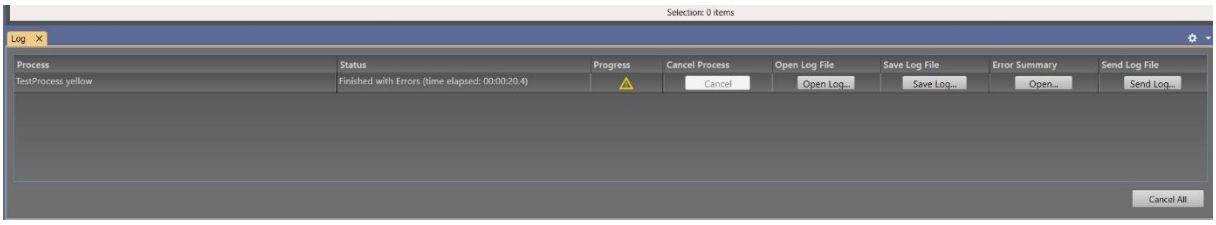
2.4. The Log Window

The log window consists of two tabs.

The **output** tab displays the current log file in the manager. The expander arrow in the upper left corner opens a search field that can be used to search for strings in the current log file. The **`Aa`** button can be used to force a case-sensitive notation.



The Error and Warnings tab displays a summary of the process history and offers options for handling the log file.



In the column ***Process*** the name of the current process is displayed.

The ***Status*** column shows whether the process is running (*Running*), has been completed successfully (*Finished with Success*), has been completed with warnings (*Finished with Errors*), has been completed with errors (*Finished with Severe Error*), and the duration of the respective process.

The ***Progress*** column contains a progress bar when the process is running and - depending on the type of completion - an icon for success, warnings, and errors.

The **`Cancel`** button in the Cancel Process column can be used to cancel an individual process.

The **`Open Log...`** button in the ***Open Log File*** column opens the manager's log file and jumps to the location of the affected process.

With **`Save Log...`** a new storage location for the log file can be specified.

The **`Open...`** button in the ***Error Summary*** column opens an overview of the error descriptions if the process was completed with warnings.

The **`Send Log`** button in the ***Send Log File*** column zips all necessary log files and attaches them to an email in the installed default mail client at support@uvmsystems.com.

The **`Cancel All`** button cancels all running processes.

3 Contact

UVM Systems GmbH

www.citygrid.at
www.uvmsystems.com

UVM Systems GmbH
www.citygrid.at
www.uvmsystems.com