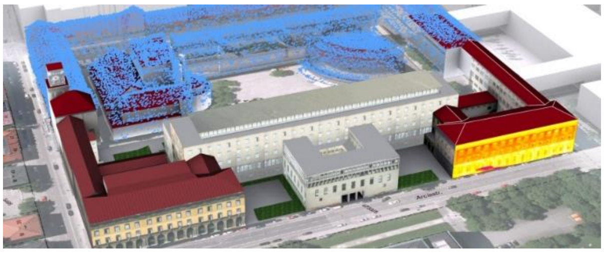
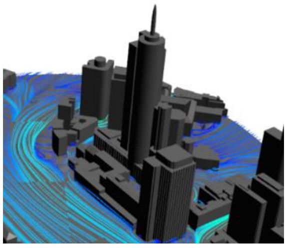
3D City Database for CityGML

A Hands-on Tutorial for Beginners

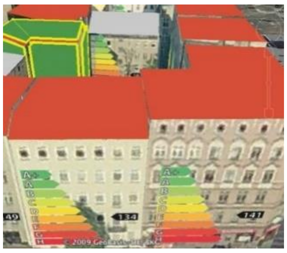
Version 3.3.0

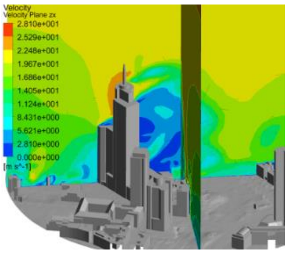
2016





3D City Database





Chair of Geoinformatics, Technical University of Munich

Prof. Dr. Thomas H. Kolbe
Arcisstr. 21
80333 Munich
Tel.: +49.89.289.22578
Fax: +49.89.289.22878
thomas.kolbe(at)tum.de
https://www.gis.bgu.tum.de/

License

According to CC BY-NC-SA 3.0 permission is granted to share this document, i.e. copy and redistribute the material in any medium or format, and to adapt it, i.e. remix, transform, and build upon the material under the following conditions:

1. **Attribution** — You must **give appropriate credit**, provide a link to the license, and **indicate if changes were made**. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
2. **NonCommercial** — You may **not use** the material **for commercial purposes**.
3. **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the **same license** as the original.
4. **No additional restrictions** — You may not apply legal terms or **technological measures** that legally restrict others from doing anything the license permits.

A clarification concerning point 2: non-commercial usage means that this hands-on tutorial is not allowed to be used in a commercial training course on the 3DCityDB. It is, however, allowed to use this hands-on tutorial for learning about 3DCityDB within commercial companies or projects.

Authors of this tutorial

Name	**Institution**	**Email**
Zhihang Yao	Chair of Geoinformatics, Technical University of Munich	zhihang.yao@tum.de
Son Nguyen	Chair of Geoinformatics, Technical University of Munich	son.nguyen@tum.de
Maximilian Sindram	Chair of Geoinformatics, Technical University of Munich	maximilian.sindram@tum.de
Kanishk Chaturvedi	Chair of Geoinformatics, Technical University of Munich	kanishk.chaturvedi@tum.de
Thomas H. Kolbe	Chair of Geoinformatics, Technical University of Munich	thomas.kolbe@tum.de

Active participants in development of 3DCityDB

Name	**Institution**	**Email**
Thomas H. Kolbe Zhihang Yao	Chair of Geoinformatics, Technical University of Munich	thomas.kolbe@tum.de zhihang.yao@tum.de
Claus Nagel Richard Redweik	virtualcitySYSTEMS GmbH, Berlin	cnagel@virtualcitysystems.de rredweik@virtualcitysystems.de
Philipp Willkomm György Hudra Arda Müftüoglu	M.O.S.S. Computer Grafik Systeme GmbH, Taufkirchen, Germany	pwillkomm@moss.de ghudra@moss.de amueftueoglu@moss.de
Felix Kunde	-	felix-kunde@gmx.de

The images on the cover page were provided by:

- Chair of Photogrammetry and Remote Sensing & Chair of Cartography, Technische Universität München

- Geobasisdaten: © Stadtvermessung Frankfurt am Main
- IDAC Ltd, UK.
- virtualcitySYSTEMS GmbH, Berlin, Germany

- Chair of Geoinformatics, Technische Universität München. Image created based on master thesis work of Matthias Körner, jointly supervised with HTW Dresden
- 3D City Model of Berlin © Berlin partner GmbH

- M.O.S.S. Computer Grafik Systeme GmbH, Taufkirchen, Germany

Table of Contents

Disclaimer

The *3D City Database version 3.3.0* developed in collaboration of the *Chair of Geoinformatics, Technische Universität München (TUMGI), virtualcitySYSTEMS GmbH*, and *M.O.S.S. Computer Grafik System GmbH* is free software and licensed under the Apache License, Version 2.0. See the file LICENSE file shipped together with the software for more details. You may obtain a copy of the license at http://www.apache.org/licenses/LICENSE-2.0.

Please note that releases of the software before version 3.3.0 continue to be licensed under GNU LGPL 3.0. To request a previous release of the 3D City Database under Apache License 2.0 create a GitHub issue at https://github.com/3dcitydb.

The official 3DCityDB project homepage is http://www.3dcitydb.org

THE SOFTWARE AND THIS TUTORIAL ARE PROVIDED BY *TUMGI* "AS IS" AND "WITH ALL FAULTS." *TUMGI* MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE QUALITY, SAFETY OR SUITABILITY OF THE SOFTWARE AND TUTORIAL, EITHER EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

TUMGI MAKES NO REPRESENTATIONS OR WARRANTIES AS TO THE TRUTH, ACCURACY OR COMPLETENESS OF ANY STATEMENTS, INFORMATION OR MATERIALS CONCERNING THE SOFTWARE THAT IS CONTAINED ON AND WITHIN ANY OF THE WEBSITES OWNED AND OPERATED BY *TUMGI*.

IN NO EVENT WILL *TUMGI* BE LIABLE FOR ANY INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES HOWEVER THEY MAY ARISE AND EVEN IF *TUMGI* HAVE BEEN PREVIOUSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Requirements

The 3D City Database (3DCityDB) is an open source software package allowing for efficient storage, management, and visualization of 3D city models in CityGML format. Setting up a 3DCityDB environment requires a running installation of several software tools which are available **completely free of costs**. This section provides an overview of the relevant software components and the minimum system requirements to complete a proper installation of 3D City Database. Additional software programs which are specifically required for this tutorial are also listed.

1. Java

Prior to the setup of 3D City Database, the Java **8 Runtime Environment (or higher)** must be installed on your system. The latest Java versions can be downloaded from:

http://java.com/en/download/manual.jsp

2. PostgreSQL + PostGIS

Setting up a 3DCityDB instance requires a running installation of an Oracle or PostgreSQL database server. In this tutorial we will use PostgreSQL which is a free and Open Source database server providing all functionalities and performance the 3DCityDB requires. Please note that only **PostgresSQL 9.1 or higher** with **PostGIS extension 2.0 or higher** are supported. The database could be either managed via command line interface **psql** that is shipped with the original installation packages of **pgAdmin III** or the graphical user interface that supports all required functionalities. The installation package of the actual versions of the PostgreSQL/PostGIS database can be downloaded via the following web link:

http://www.postgresql.org/download/

The installation packages of the current version of pgAdmin III can be downloaded via the following web link:

http://www.pgadmin.org/download/

3. 3DCityDB Scripts

The 3DCityDB software package comes with a set of SQL scripts to create the relational schema on the spatial database system (PostgreSQL/PostGIS) and with a group of PL/pgSQL scripts to manage the 3D city model stored in the database. The 3DCityDB scripts can be downloaded via the following link:

http://www.3dcitydb.org/3dcitydb/downloads/ (Homepage and official release channel)

https://github.com/3dcitydb/3dcitydb/releases (Development platform)

If you’ve already installed the Importer/Exporter tool (see the next paragraph), the 3DCityDB scripts can be found in the application folder where the Importer/Exporter tool has been installed, and no separate download of the 3DCityDB scripts is required.

4. Importer-Exporter-Tool

The 3DCityDB software package provides a Java-based Importer/Exporter tool for high performance importing and exporting of CityGML datasets and exporting of KML/COLLADA/glTF datasets. For installation of the Importer/Exporter tool, a universal installer is available to guide you through the steps of the setup process.

A full installation of the Importer/Exporter tool including extensions (Plugins), documentation, and example CityGML files requires approx. 149 MB of hard disk space. The latest version of the installer package can be obtained from:

http://www.3dcitydb.org/3dcitydb/downloads/

5. Spreadsheet-Generator-Plugin

Functionalities in the Importer/Exporter tool can be extended by the installation of plugins that can add specific abilities for interacting with the 3D City Database or external data. The Spreadsheet Generator is a plugin that allows for exports of 3D City Database contents in tabular form (CSV and Microsoft Excel). This Plugin is an additional component which can be optionally installed together with the 3DCityDB Importer/Exporter tool. During the installation of the Importer/Exporter tool, the installation wizard will ask you if you want to install the Spreadsheet Generator Plugin. Its source code can be downloaded via the GitHub link below:

https://github.com/3dcitydb/plugin-spreadsheet-generator

6. 3DCityDB-Web-Map

The 3DCityDB software package provides a specific WebGL-based 3D web client which has been developed using the Open Source Cesium Web Globe for interactive visualization and exploration of the 3D city models within a web browser without the needs of additional Browser Plugins. The 3D web client is a static web client purely written in HTML and JavaScript and can be very easily deployed by uploading the web client files to a web server. The web client files are shipped with the Importer-Exporter installer package and can also be separately downloaded via the GitHub link below:

https://github.com/3dcitydb/3dcitydb-web-map

If you don’t want to deploy the 3D web client on your own server, there is also an official web link (see the link below) that allows to run the 3D web client directly.

http://www.3dcitydb.org/3dcitydb-web-map/1.1/3dwebclient/index.html

7. Node.js (required for this tutorial)

The web server used in this tutorial is a lightweight JavaScript program that has been developed by Cesium-group and can be very easily deployed on a computer. This JavaScript-based web server relies on the Node.js framework which is an open source JavaScript runtime environment that can interpret and run JavaScript programs without a web browser. The latest version of Node.js can be download via the web link below:

https://nodejs.org/en/

8. Google Earth Pro (required for this tutorial)

In this tutorial, the created KML/COLLADA data will be visually inspected by using the Google Earth Pro program which is nowadays **freely** available via the following link:

https://www.google.de/earth/download/gep/agree.html

Preparation of the course data

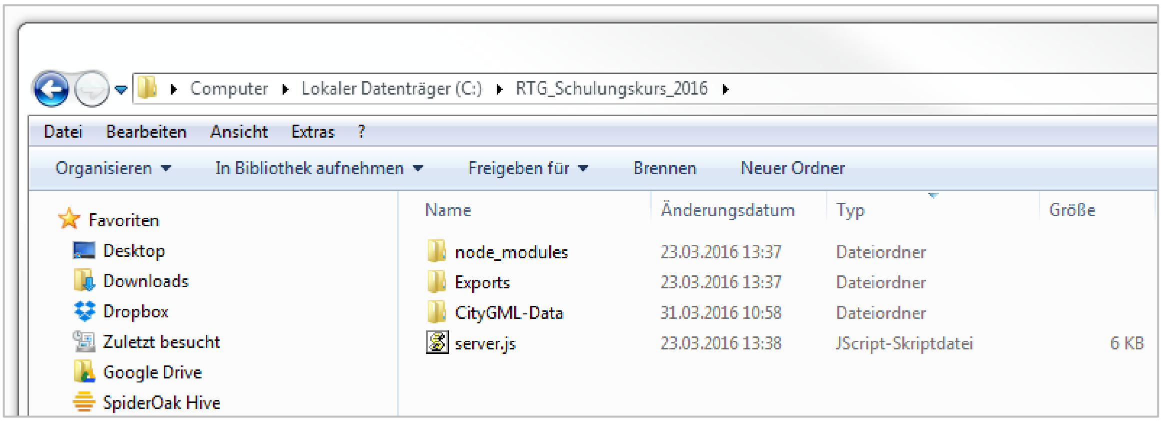
The data file used in this tutorial can be downloaded from the following link:

http://www.3dcitydb.net/3dcitydb/fileadmin/TUM_Workshop/RTG_Schulungskurs_2016.zip

□ Download this zip file and extract it into the following directory, which will be used as the working directory throughout this course.

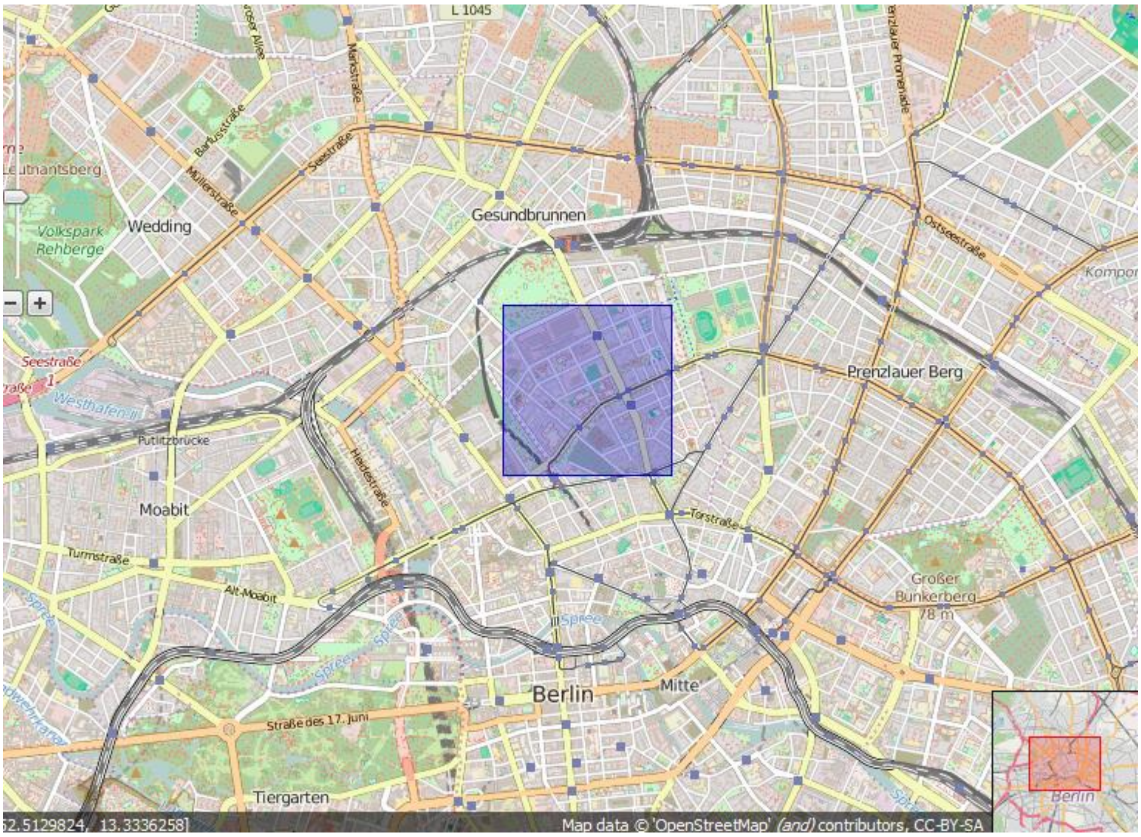
"C:\RTG_Schulungskurs_2016"

If you use the Windows operating system, please make sure that the system location and contents of the ***RTG_Schulungskurs_2016*** folder should look something like the following figure.



Name	Änderungsdatum	Typ	Größe
node_modules	23.03.2016 13:37	Dateiordner	
Exports	23.03.2016 13:37	Dateiordner	
CityGML-Data	31.03.2016 10:58	Dateiordner	
server.js	23.03.2016 13:38	JScript-Skriptdatei	6 KB

- The **node_modules** folder and the **server.js** file build a local web server that will be used to host the 3D web client and publish data in exercise 7.
- The **Exports** folder will be used to store your created data file throughout the course. There are three pre-created subfolders which will contain the output **CityGML**, **KML**, and **Spreadsheet** files respectively.
- In the **CityGML-Data** folder you can find the CityGML file that will be used throughout this course. This CityGML file contains **954** building objects from Berlin with textures and the corresponding city area can be seen in the following 2D map:


L 1045
Leutnantsberg
Volkspark Rehberge
Wedding
Barfusstraße
Müllerstraße
Seestraße
Gesundbrunnen
Ostseestraße
Prenzlauer Berg
Westhafen
Putlitzbrücke
Moabit
Turmstraße
Alt-Moabit
Heidestraße
Torstraße
Großer Bunkerberg 78 m
Spree
Berlin
Mitte
Straße des 17. Juni
Tiergarten
52.5129824, 13.3336258]
Map data © 'OpenStreetMap' (and) contributors, CC-BY-SA

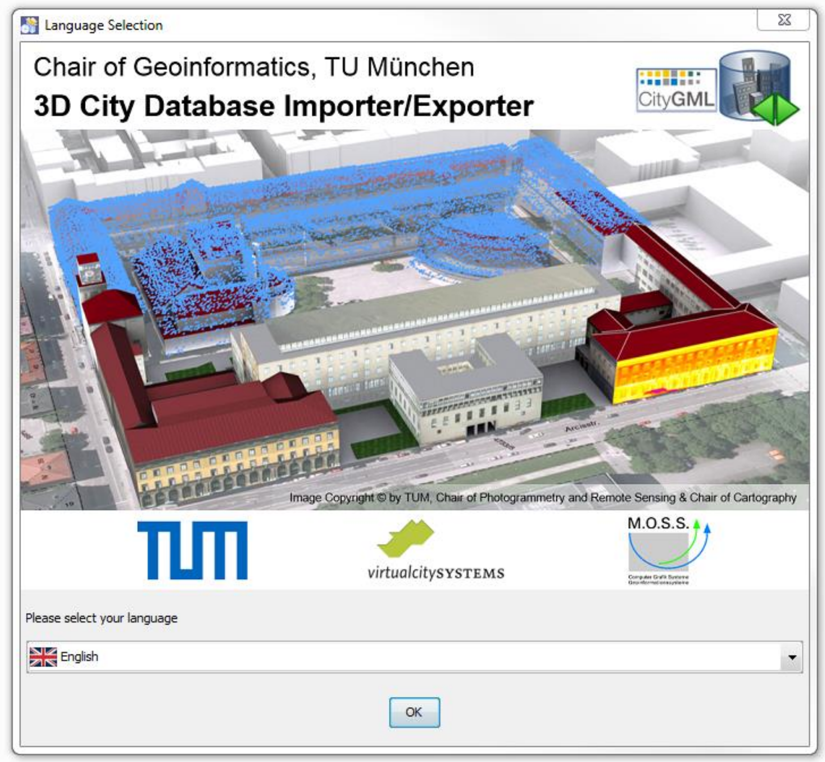
Exercise 1: Importer/Exporter Tool Installation

□ Download the latest version of the Importer/Exporter tool installer (**3DCityDB-Importer-Exporter-3.3-Setup.jar**) from:

https://github.com/3dcitydb/importer-exporter/releases

□ After the download, double click on the **3DCityDB-Importer-Exporter-3.3-Setup.jar** file to start the installation process

The first page of the installation wizard looks like the following figure:



Language Selection

Chair of Geoinformatics, TU München

3D City Database Importer/Exporter

Image Copyright © by TUM, Chair of Photogrammetry and Remote Sensing & Chair of Cartography

Please select your language

English

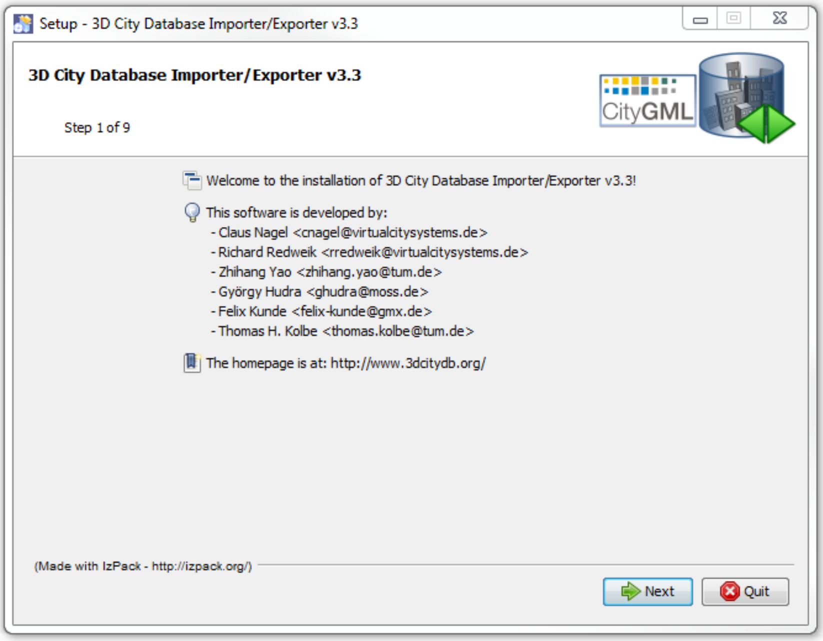
In this wizard page you are able to select the language you prefer. The selected language will be applied to label the graphic user interface of the Importer/Exporter tool. Currently, there are two languages available: **English** and **German**. In this tutorial we chose the language **English**:

□ Click the dropdown list ***please select your language*** and select the option ***English***

□ Click the ***OK*** button to continue the installation process

The next page of the wizard (see the following figure) shows a list of all people who are involved in the development of the 3DCityDB. You can also find the URL of the 3DCityDB homepage there.

□ Click on the ***Next*** button to continue the installation process



3D City Database Importer/Exporter v3.3

Step 1 of 9

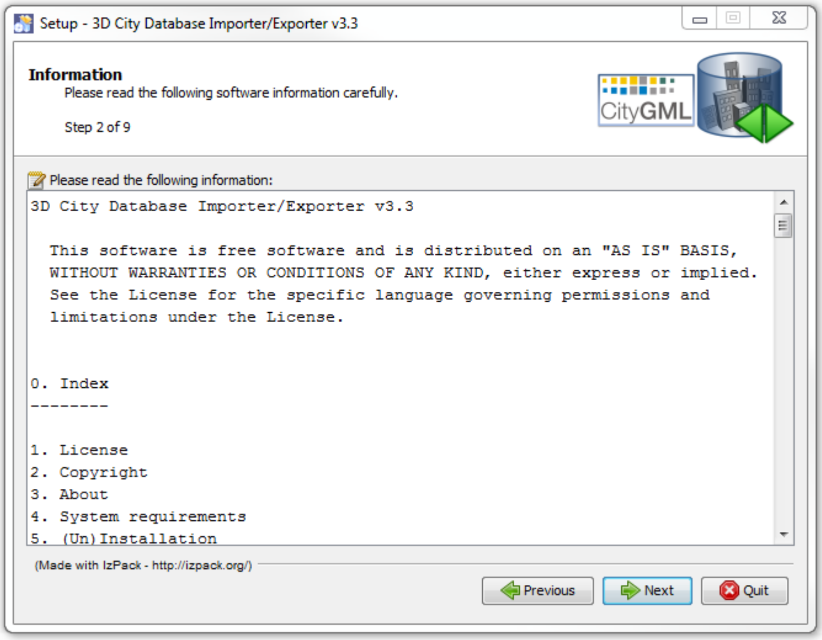
Welcome to the installation of 3D City Database Importer/Exporter v3.3!

This software is developed by:
- Claus Nagel <cnagel@virtualcitysystems.de>
- Richard Redweik <rredweik@virtualcitysystems.de>
- Zhihang Yao <zhihang.yao@tum.de>
- György Hudra <ghudra@moss.de>
- Felix Kunde <felix-kunde@gmx.de>
- Thomas H. Kolbe <thomas.kolbe@tum.de>

The homepage is at: http://www.3dcitydb.org/

(Made with IzPack - http://izpack.org/)

On the next page of the wizard (see the screenshot below) you can find the detailed information (e.g. Copyright, System requirements etc.) of the Importer/Exporter tool.



Information

Please read the following software information carefully.

Step 2 of 9

Please read the following information:

```
3D City Database Importer/Exporter v3.3

  This software is free software and is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.


0. Index
--------

1. License
2. Copyright
3. About
4. System requirements
5. (Un)Installation
```

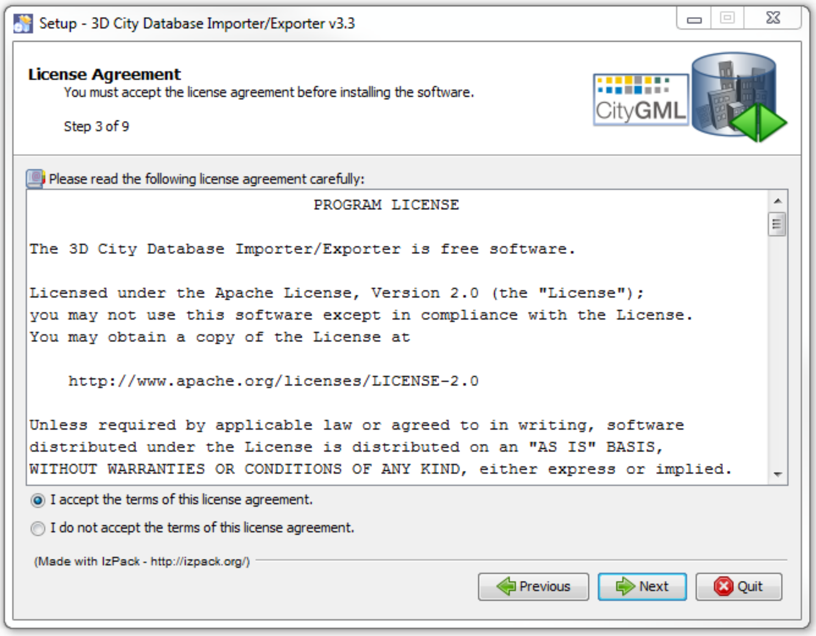
(Made with IzPack - http://izpack.org/)

□ Click on the ***Next*** button to continue the installation process

On the next page of the wizard (see the following screenshot) you should read the license agreement.

□ Check the radio button ***I accept the terms of the license agreement***

□ Click on the ***Next*** button to continue the installation process



License Agreement
You must accept the license agreement before installing the software.

Step 3 of 9

Please read the following license agreement carefully:

```
                              PROGRAM LICENSE

The 3D City Database Importer/Exporter is free software.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this software except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

● I accept the terms of this license agreement.

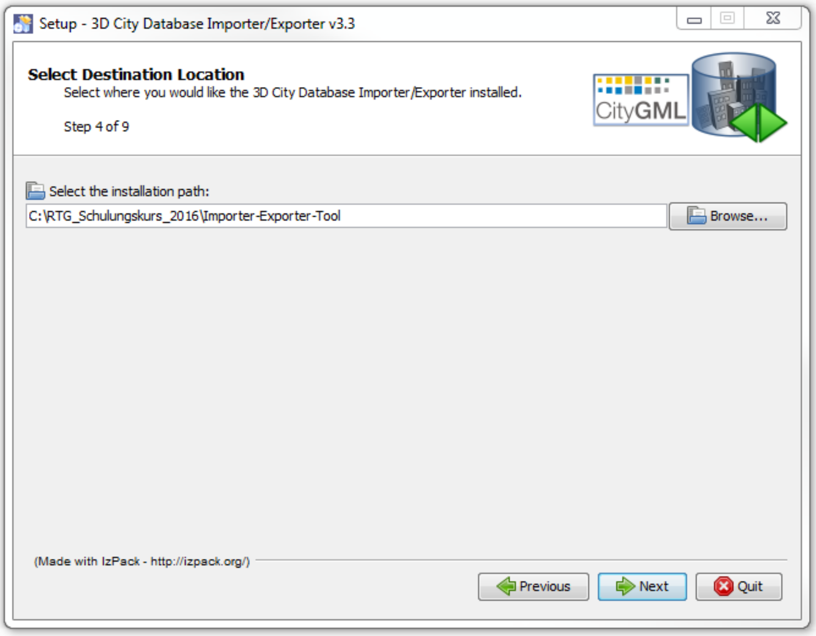
○ I do not accept the terms of this license agreement.

(Made with IzPack - http://izpack.org/)

On the next page of the wizard (see the screenshot below) you need to select a folder where you’ll install the Importer/Exporter tool.

□ Click ***Browse…*** button, The *Choose Folder* dialog window will be displayed. Create and select the following folder as installation folder

“***C:\RTG_Schulungskurs_2016\Importer-Exporter-Tool***“



Select Destination Location
Select where you would like the 3D City Database Importer/Exporter installed.

Step 4 of 9

Select the installation path:
C:\RTG_Schulungskurs_2016\Importer-Exporter-Tool

(Made with IzPack - http://izpack.org/)

□ Click the ***Next*** button on the wizard.

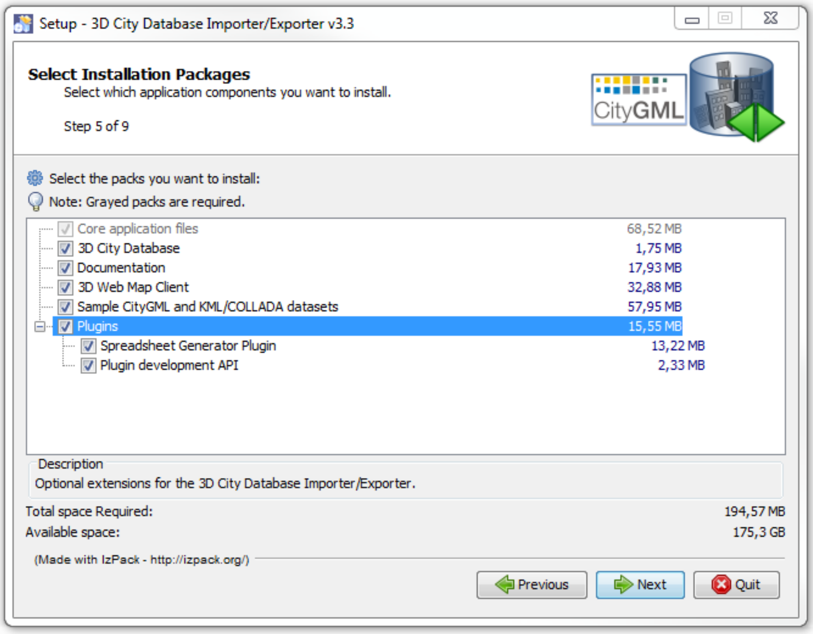
A dialog window may appear to inform you that a new folder will be created or the already existing folder will be overwritten. Click ***OK*** or ***YES*** button to confirm it.

□ Click ***OK*** or ***YES*** button to confirm it and continue the installation process.

On the next page of the installation wizard (see the screenshot below) you can choose the optional installation packages that can be installed in addition to the mandatory core application files of the Importer/Exporter tool.

□ Activate all checkboxes, we install **all** the available packages

□ Click on the ***Next*** button to continue the installation process



Setup - 3D City Database Importer/Exporter v3.3

Select Installation Packages
Select which application components you want to install.

Step 5 of 9

Select the packs you want to install:
Note: Grayed packs are required.

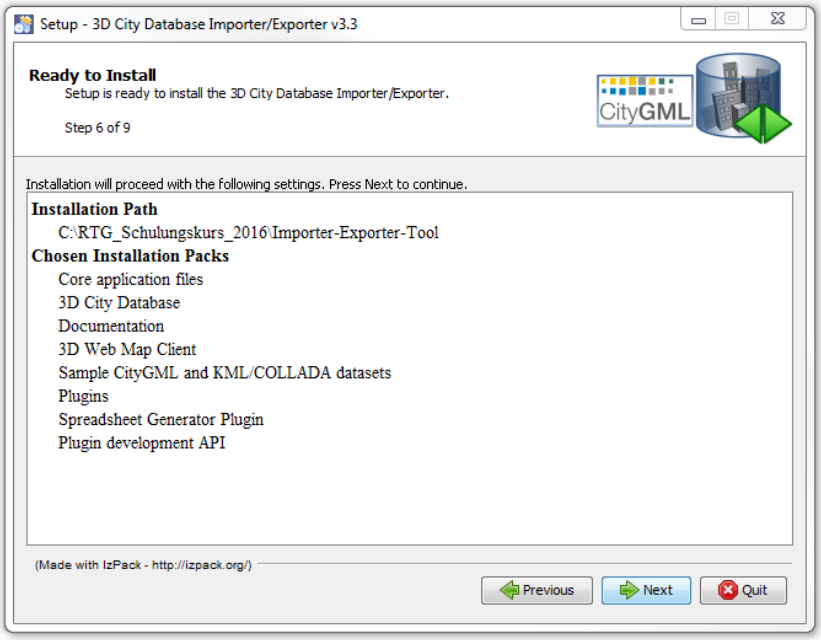
☑ Core application files	68,52 MB
☑ 3D City Database	1,75 MB
☑ Documentation	17,93 MB
☑ 3D Web Map Client	32,88 MB
☑ Sample CityGML and KML/COLLADA datasets	57,95 MB
☑ Plugins	15,55 MB
☑ Spreadsheet Generator Plugin	13,22 MB
☑ Plugin development API	2,33 MB

Description
Optional extensions for the 3D City Database Importer/Exporter.

Total space Required: 194,57 MB
Available space: 175,3 GB

On the next page of the wizard, the installation path and the chosen installation packages are listed. You can review them again.

□ Click on the ***Next*** button to continue the installation process



Setup - 3D City Database Importer/Exporter v3.3

Ready to Install
Setup is ready to install the 3D City Database Importer/Exporter.

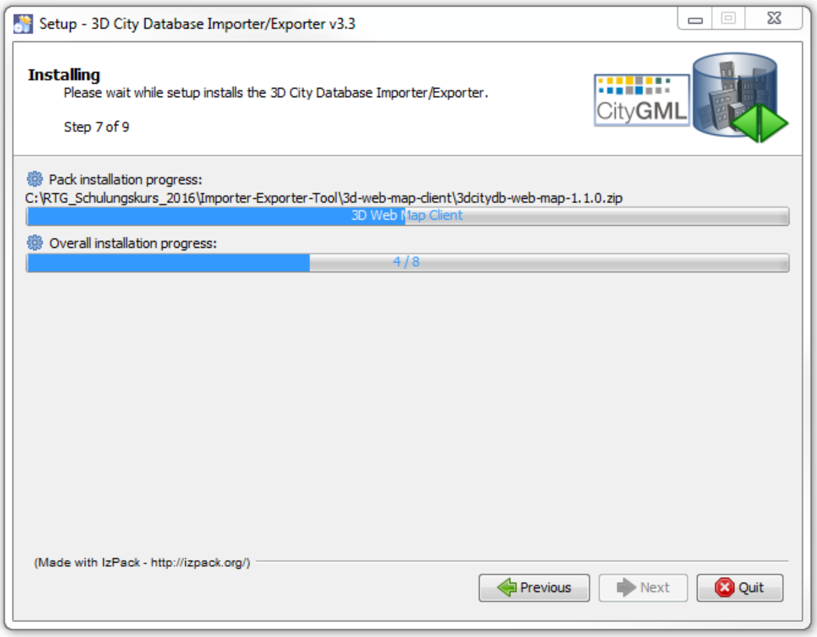
Step 6 of 9

Installation will proceed with the following settings. Press Next to continue.

Installation Path
C:\RTG_Schulungskurs_2016\Importer-Exporter-Tool
Chosen Installation Packs
Core application files
3D City Database
Documentation
3D Web Map Client
Sample CityGML and KML/COLLADA datasets
Plugins
Spreadsheet Generator Plugin
Plugin development API

Now, the selected installation packages are going to be installed on your machine. The installation process takes just around half a minute. The progress bars on the wizard (see the screenshot below) show how the installation is proceeding.

□ Once the installation is finished, click on the ***Next*** button to continue the installation process



Setup - 3D City Database Importer/Exporter v3.3

Installing

Please wait while setup installs the 3D City Database Importer/Exporter.

Step 7 of 9

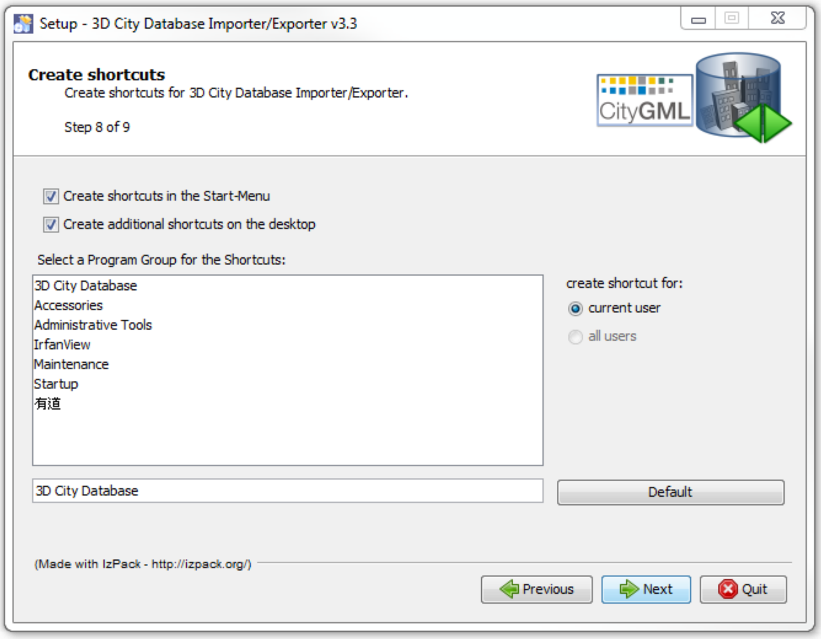
Pack installation progress:
C:\RTG_Schulungskurs_2016\Importer-Exporter-Tool\3d-web-map-client\3dcitydb-web-map-1.1.0.zip
3D Web Map Client

Overall installation progress:
4 / 8

(Made with IzPack - http://izpack.org/)

On the next page of the wizard (see the screenshot below) you can create shortcuts in both Start-Menu and on the desktop for quick launching of the Importer/Exporter tool. You can also specify whether the created shortcut will be available just for the current system user or for all system users.

□ Activate both checkboxes ***Create shortcuts in the Start-Menu*** and ***Create additional shortcuts on the desktop***



Setup - 3D City Database Importer/Exporter v3.3

Create shortcuts

Create shortcuts for 3D City Database Importer/Exporter.

Step 8 of 9

☑ Create shortcuts in the Start-Menu

☑ Create additional shortcuts on the desktop

Select a Program Group for the Shortcuts:

3D City Database
Accessories
Administrative Tools
IrfanView
Maintenance
Startup
有道

3D City Database

create shortcut for:
● current user
○ all users

(Made with IzPack - http://izpack.org/)

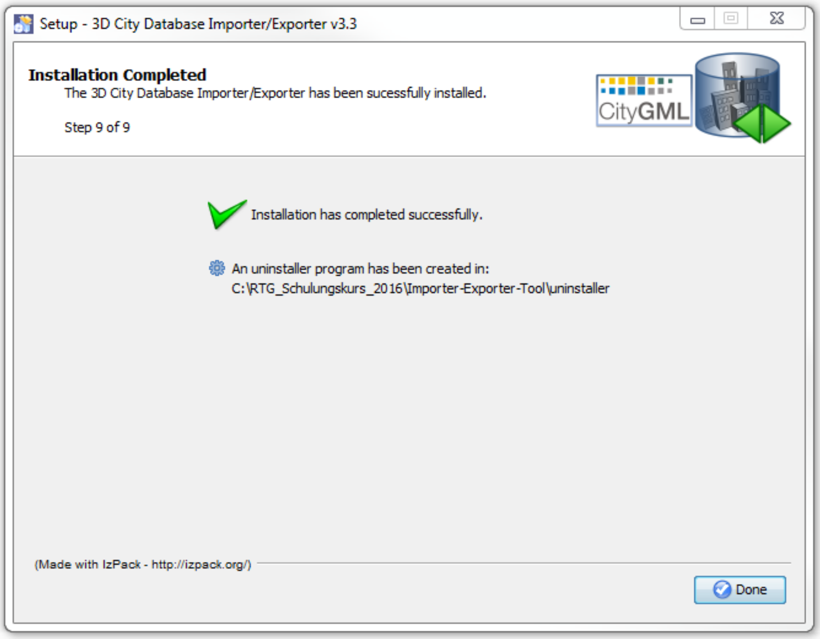
□ Activate the radio button ***current user***

□ Fill the text field with the text “***3D City Database***”

□ Click on the ***Next*** button to continue the installation process

The last page of the wizard (see the screenshot below) will appear to show that the Importer/Exporter tool has been successfully installed.

□ Click on the ***Done*** button to close the wizard and complete the installation process



Installation Completed
The 3D City Database Importer/Exporter has been sucessfully installed.

Step 9 of 9

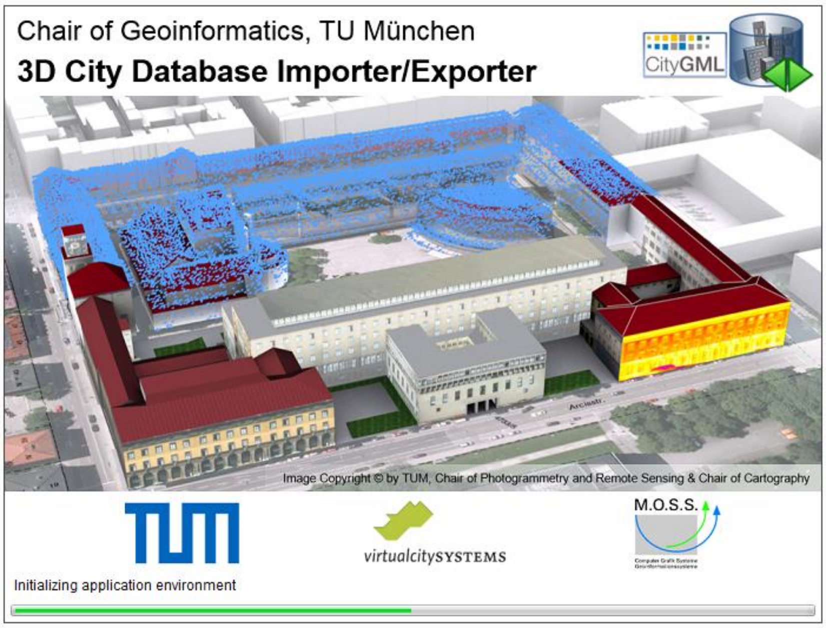
Installation has completed successfully.

An uninstaller program has been created in:
C:\RTG_Schulungskurs_2016\Importer-Exporter-Tool\uninstaller

(Made with IzPack - http://izpack.org/)

□ Double-click the shortcut icon on the desktop to run the Importer/Exporter tool

A splash window will be displayed (see the screenshot below) while the Importer/Exporter is starting up.

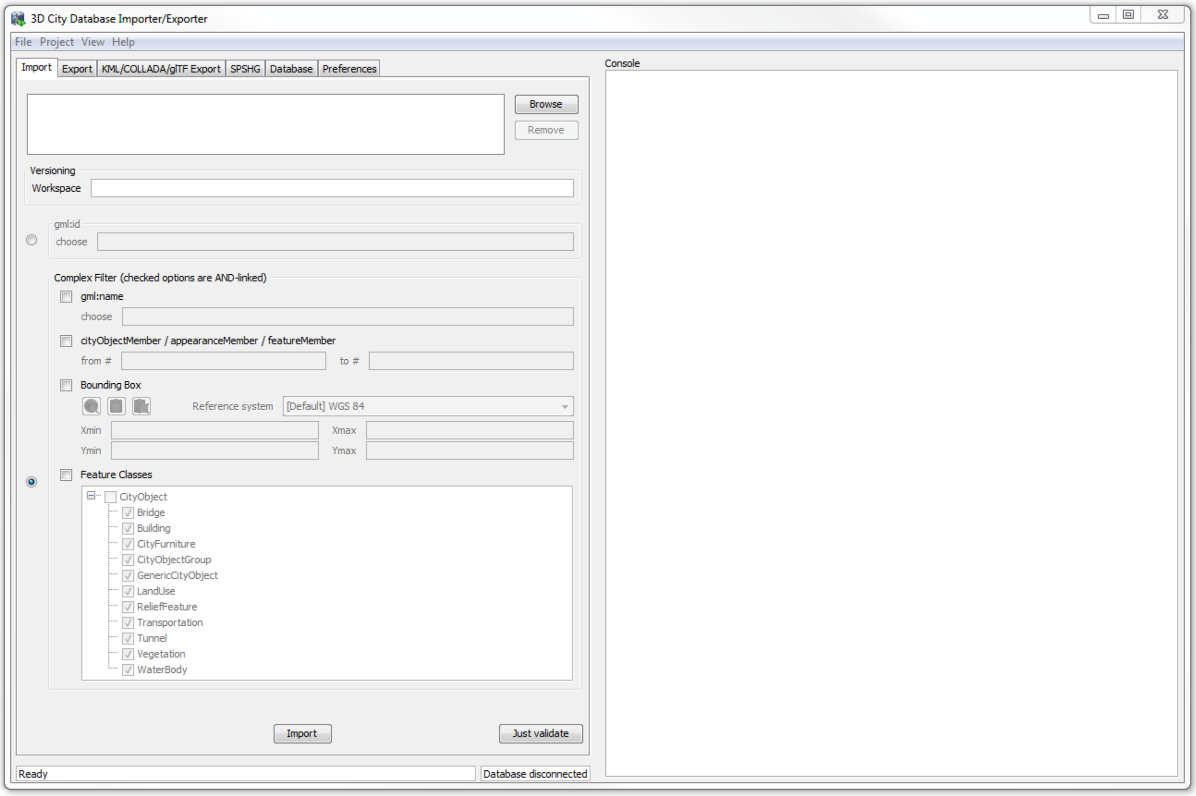


Chair of Geoinformatics, TU München
3D City Database Importer/Exporter

Image Copyright © by TUM, Chair of Photogrammetry and Remote Sensing & Chair of Cartography

Initializing application environment

After a few seconds the user interface or the Importer/Exporter tool will appear (see the screenshot below).



○ gml:id

● Complex Filter (checked options are AND-linked)

□ gml:name

□ cityObjectMember / appearanceMember / featureMember

□ Bounding Box

Reference system: [Default] WGS 84

□ Feature Classes

- □ CityObject
  - ☑ Bridge
  - ☑ Building
  - ☑ CityFurniture
  - ☑ CityObjectGroup
  - ☑ GenericCityObject
  - ☑ LandUse
  - ☑ ReliefFeature
  - ☑ Transportation
  - ☑ Tunnel
  - ☑ Vegetation
  - ☑ WaterBody

Ready

Database disconnected

Before we can continue using the Importer/Exporter tool we now need to install and setup the database management system and a 3DCityDB database account.

Exercise 2: 3D City Database Setup

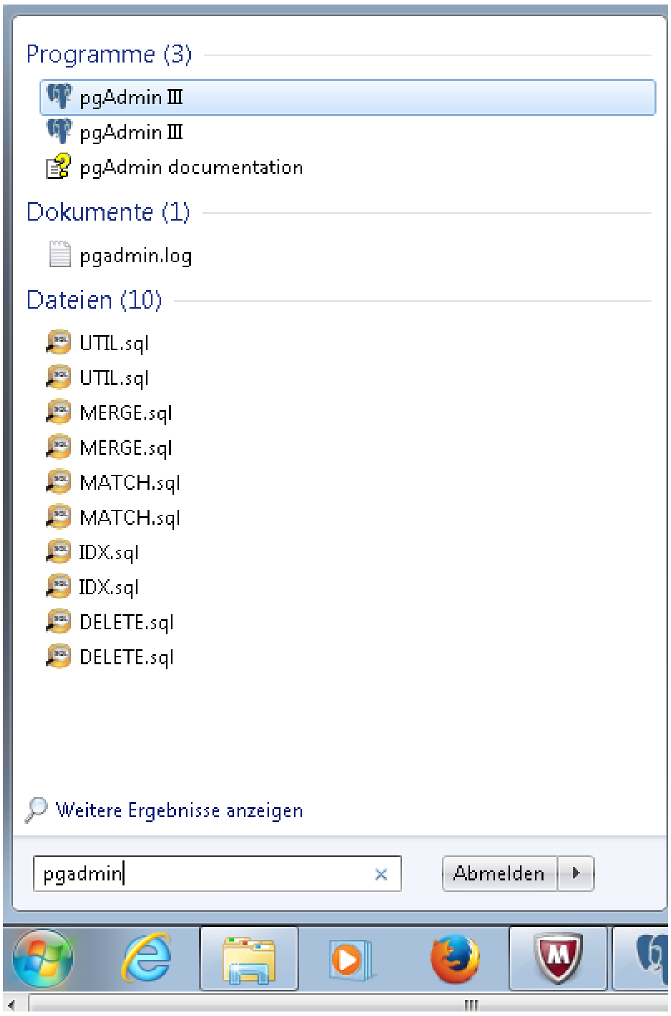
In this section we will explain how you can set up an empty database for CityGML models.

Please make sure that you have installed the PostgreSQL data base management system (DMBS) with its extension PostGIS (cf. Requirements) and pgAdmin III on your local computer.

Let’s run the pgAdmin program

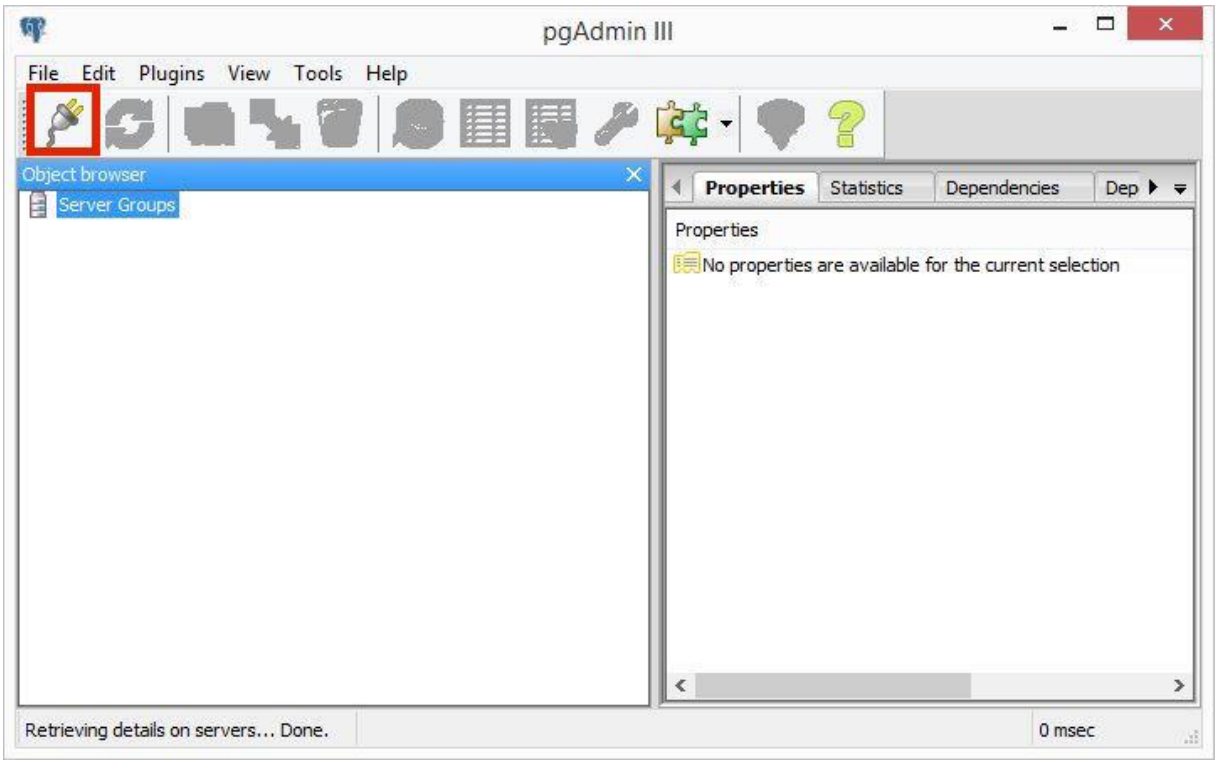
□ Click on the windows ***Start Menu*** button on the desktop

□ Search the program “**pgAdmin**” (see the screenshot below)



□ Click and run the pgAdmin program

The following figure shows the graphical user interface of pgAdmin III.


pgAdmin III
File Edit Plugins View Tools Help
Object browser
Server Groups
Properties Statistics Dependencies Dep
Properties

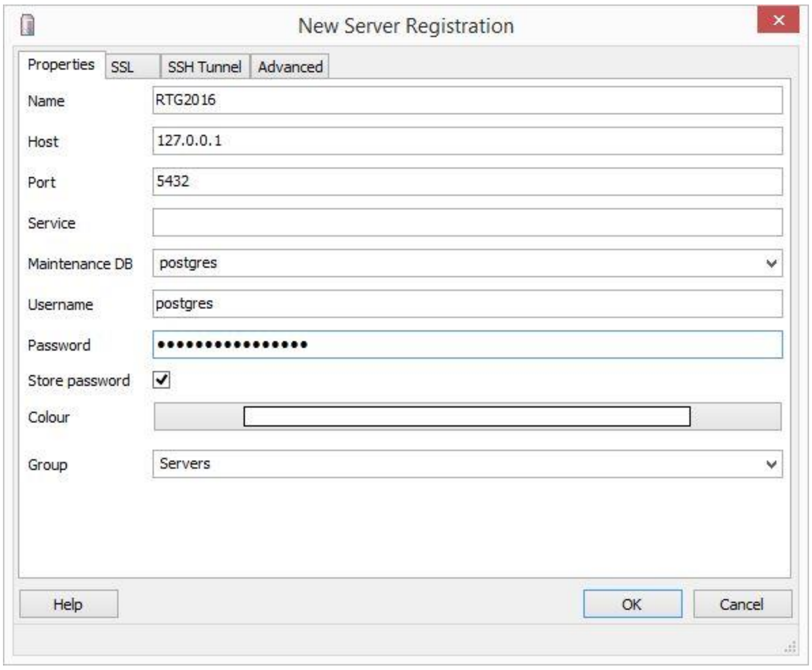
No properties are available for the current selection

Retrieving details on servers... Done.

0 msec

□ Add a new connection to a database server. Click on the red marked power plug sign (see the screenshot above).

A pop-up window appears for prompting the connection parameters to the database.



Name	RTG2016
Host	127.0.0.1
Port	5432
Service	
Maintenance DB	postgres
Username	postgres
Password	••••••••••••••••
Store password	☑
Colour	
Group	Servers

□ Type in a recognisable name (for example “**RTG2016**”) for your database connection.

□ Type the IP address “**127.0.0.1**” into the ***Host*** input field

> **Note**: This is the address of the Server hosting the DBMS. If the host system (database) and the guest system (pgAdmin III) are running on the same PC use the IP address **127.0.0.1** for local host connections.

□ Type “**5432**” into the ***Port*** input field. This value is the standard port number for PostreSQL databases.

□ Type “**postgres**” into the ***Username*** input field

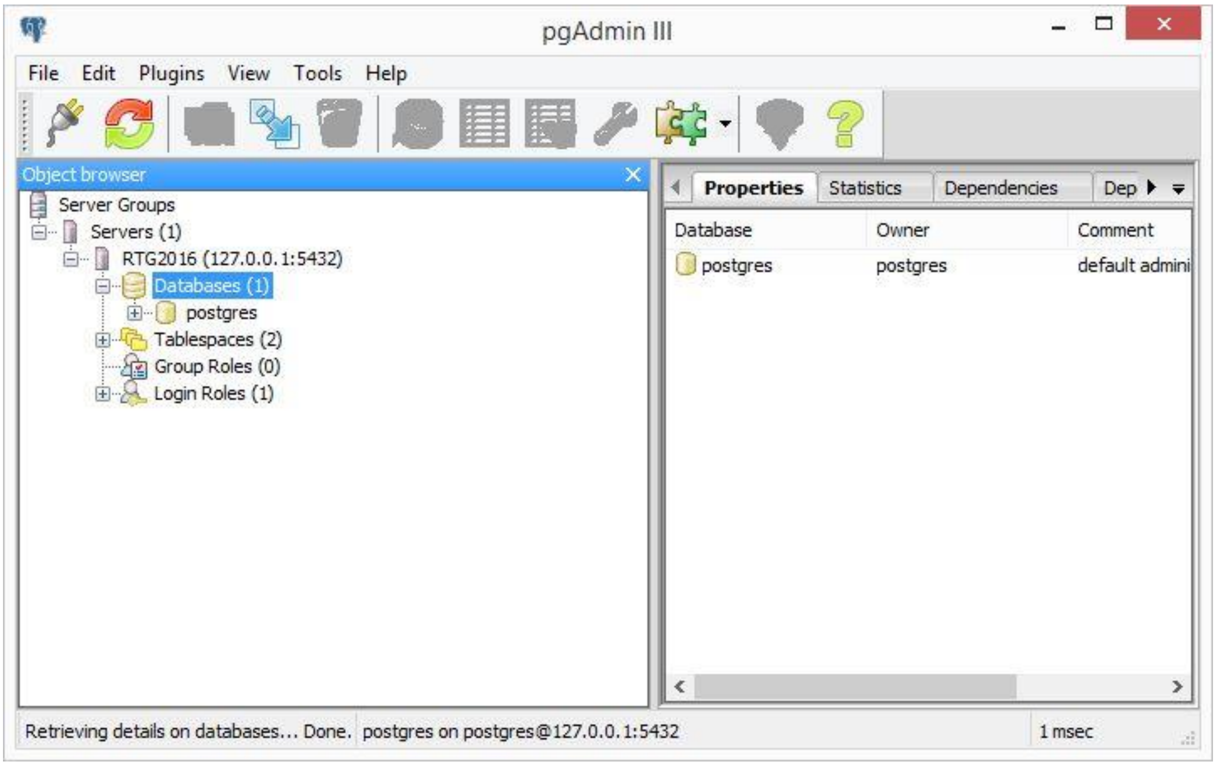
□ Type the password “**RTG@Schulung2016**” into the ***Password*** input field

Use default values for all other settings and parameters. The completed settings look like the screenshot above.

□ Click on the ***OK*** button

> **Note**: A pop up window may appear to inform you that the password will be saved, just click the ***OK*** button on the pop-up window to ignore it.

Now you see the new connection to the database. You can browse through the tree structure of the pre-defined database to get familiar with the of PostgreSQL structures.

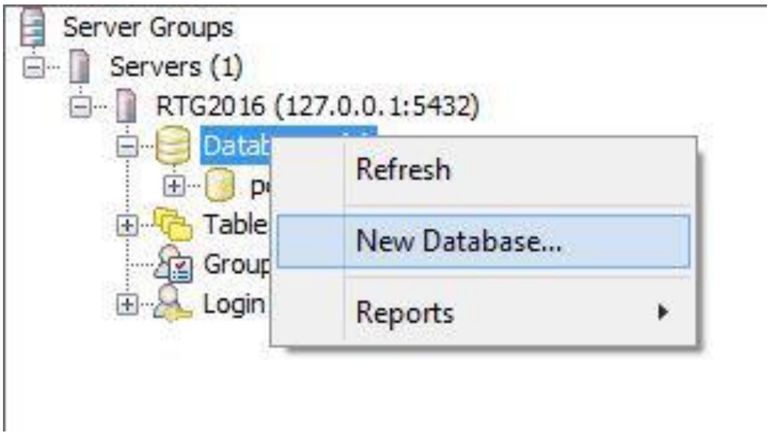


Server Groups
Servers (1)
RTG2016 (127.0.0.1:5432)
Databases (1)
postgres
Tablespaces (2)
Group Roles (0)
Login Roles (1)

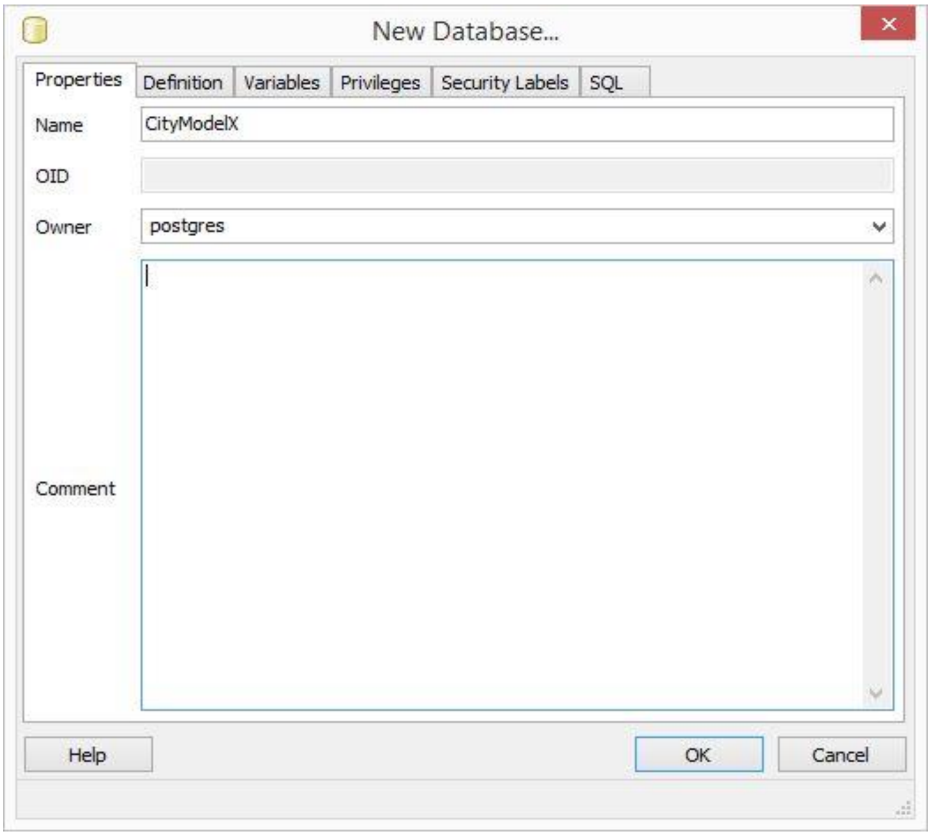
Database	Owner	Comment
postgres	postgres	default admini

Retrieving details on databases... Done. postgres on postgres@127.0.0.1:5432 1 msec

□ Right-click on the tree node ***Databases*** and choose the item ***New Database…*** to create a new database (see the screenshot below)



In the following pop-up window, you can define the name and the owner of the new database. For advanced users it is possible to define more optional settings.



Name: CityModelX

OID:

Owner: postgres

Comment:

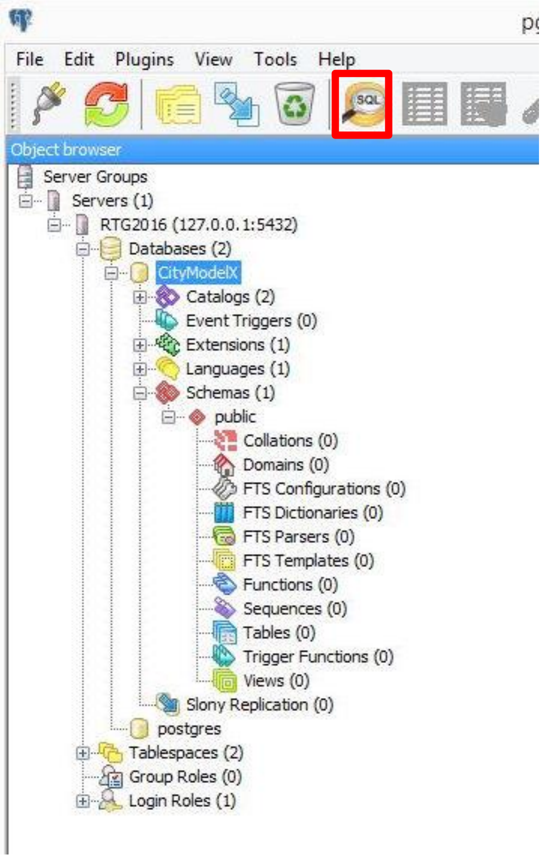
□ Choose a name (for example “**CityModelX**”) for your Database

□ Click the dropdown list ***Owner***, select ***postgres*** as owner of the database

The completed settings look like the screenshot above.

□ Click ***OK*** button to continue

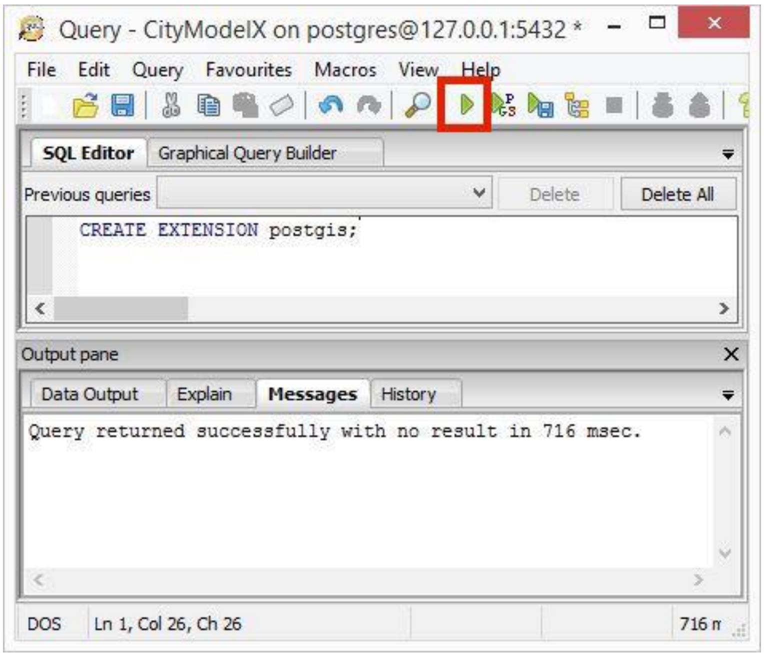
You can now access your created database in the main view of pgAdmin III.



In order to enable all spatial functions and data types you have to create the ***PostGIS*** Extension with the new database.

□ Click on the ***SQL-Button*** (marked in the screenshot above) in the menu bar of pgAdmin III.

In the following pop-up window, you are able to prompt any SQL-statement in the SQL Editor tab. The results of your SQL query are shown in the output pane.

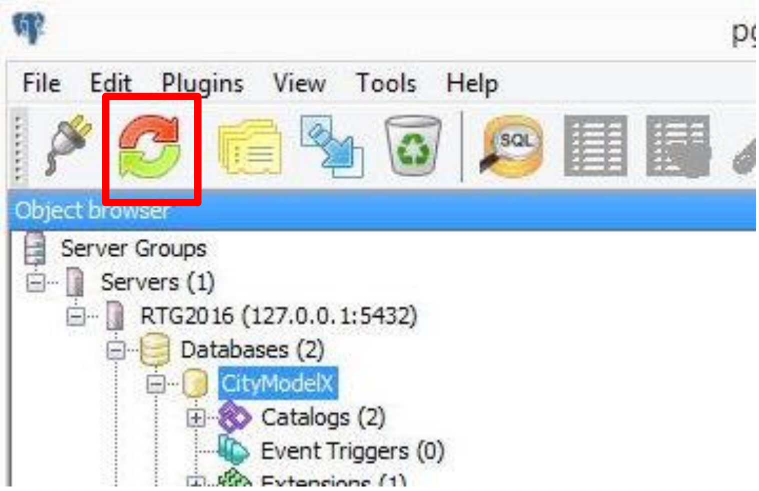


```
CREATE EXTENSION postgis;
```

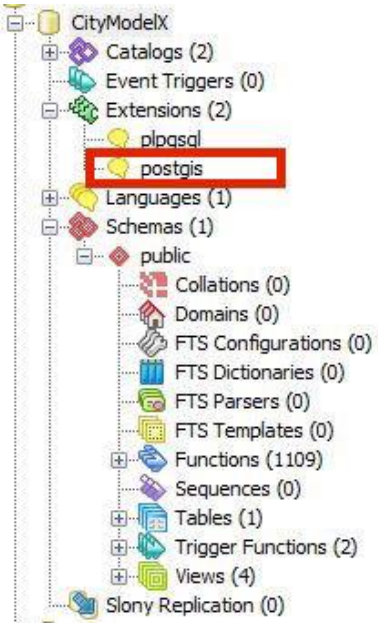
```
Query returned successfully with no result in 716 msec.
```

□ Type in the SQL-Statement “***CREATE EXTENSION postgis;***” and click on the run button (see the screenshot above)

□ Click the ***Refresh*** button on the pgAdmin window (see the screenshot below)



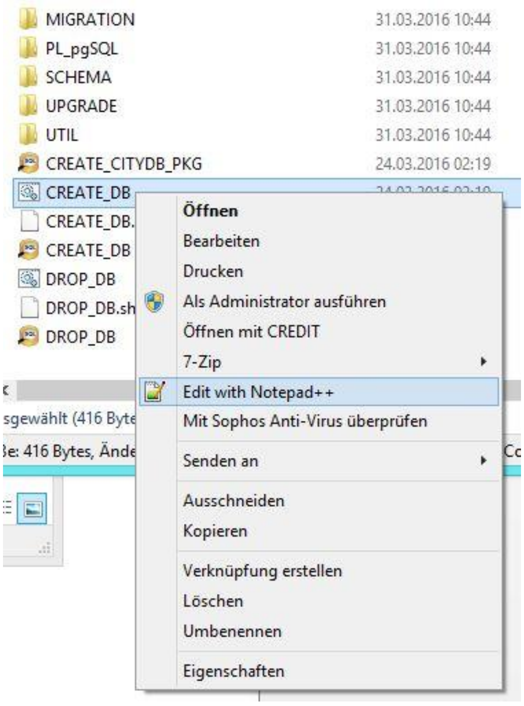
The new PostGIS extension will appear in the tree structure in the main view of pgAdmin III.



A new database is created with all required extensions to store and manage spatial features. In the next steps we are applying the CityGML data schema to the database instance. The 3DCityDB software package comes with a set of SQL scripts to create the relational schema on the spatial database system (PostgreSQL/PostGIS) and with a group of PL/pgSQL scripts to manage the 3D city model stored in the database.

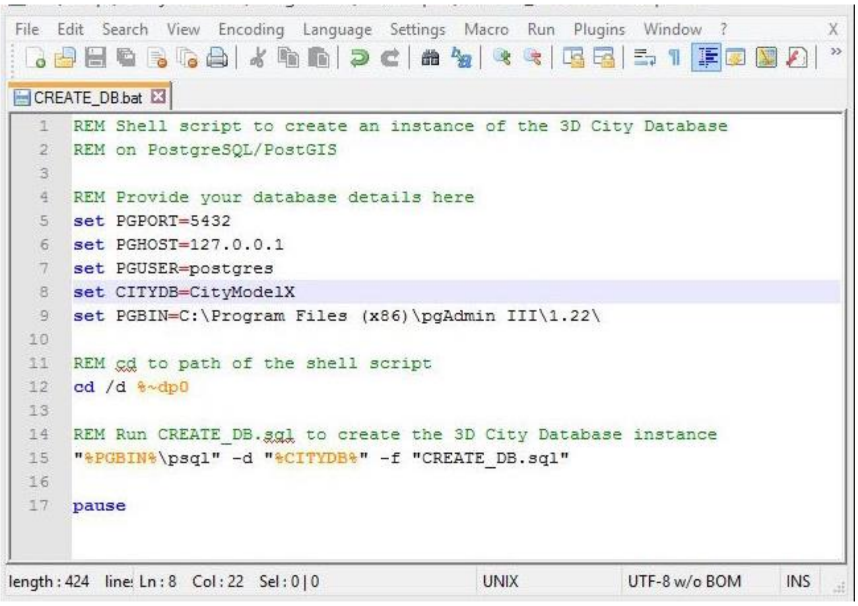
□ Go to the following folder with Windows Explorer and open the file **CREATE_DB.bat** file with a text editor of your choice (for example Nodepad++, see the screenshot below).

“**C:\RTG_Schulungskurs_2016\Importer-Exporter-Tool\3dcitydb\postgresql**”



□ Change the parameters for the database connection according to your database (see the following screenshot)

> **Note:** The value of the variable “***PGBIN***” (see line 9 in the screenshot) must be adapted to the system location of the psql executable file which is shipped with pgAdmin III by default.


File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
CREATE_DB.bat
length : 424 lines Ln : 8 Col : 22 Sel : 0 | 0 UNIX UTF-8 w/o BOM INS

```
 1  REM Shell script to create an instance of the 3D City Database
 2  REM on PostgreSQL/PostGIS
 3
 4  REM Provide your database details here
 5  set PGPORT=5432
 6  set PGHOST=127.0.0.1
 7  set PGUSER=postgres
 8  set CITYDB=CityModelX
 9  set PGBIN=C:\Program Files (x86)\pgAdmin III\1.22\
10
11  REM cd to path of the shell script
12  cd /d %~dp0
13
14  REM Run CREATE_DB.sql to create the 3D City Database instance
15  "%PGBIN%\psql" -d "%CITYDB%" -f "CREATE_DB.sql"
16
17  pause
```

□ Save and close the CREATE_DB.bat file

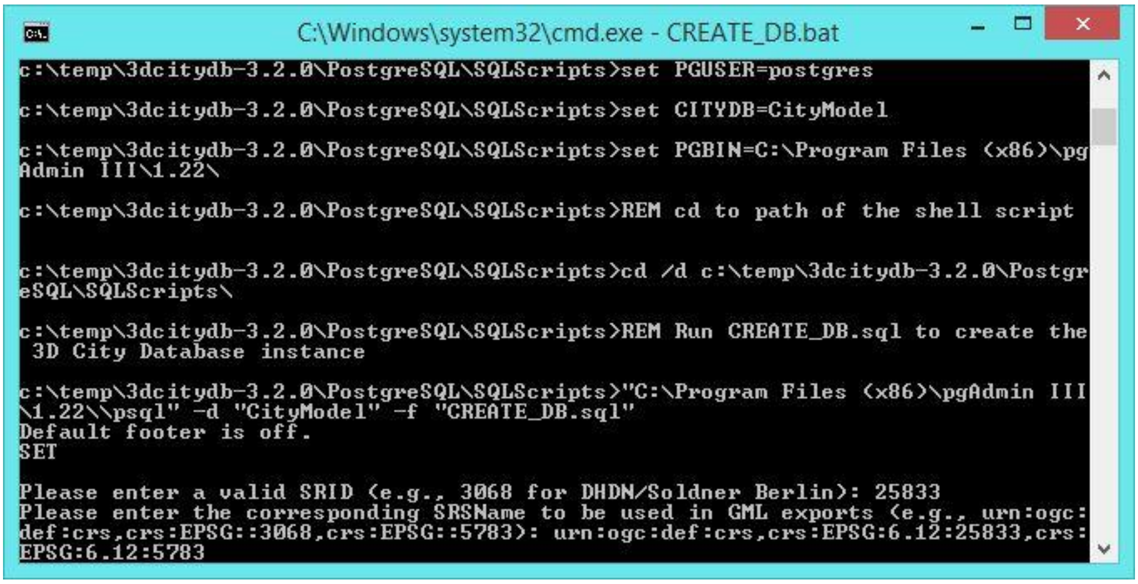
□ Start the script with double-click on it

You are asked to enter a valid *Spatial Reference Identifier* (SRID).

□ Enter “**25833**” into the command line and press **ENTER** key

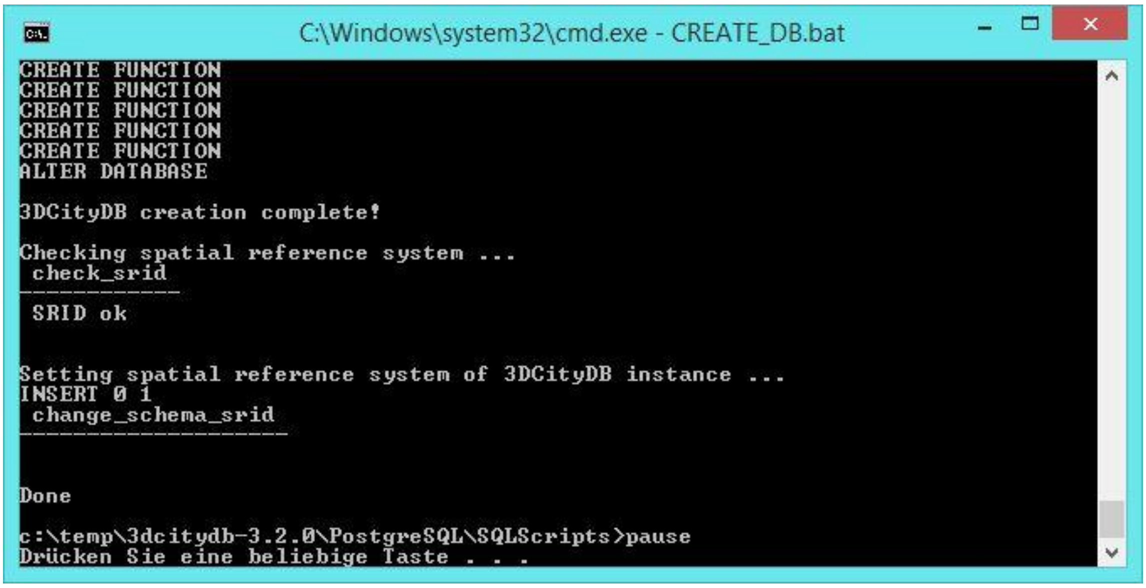
You are asked to enter a *SRSName* which is an OGC GML conformant *CRS (coordinate reference system) definition identifier*.

□ Enter "**urn:ogc:def:crs,crs:EPSG:6.12:25833,crs:EPSG:6.12:5783**" into the command line and press **ENTER** key (see the screenshot below)



```
c:\temp\3dcitydb-3.2.0\PostgreSQL\SQLScripts>set PGUSER=postgres

c:\temp\3dcitydb-3.2.0\PostgreSQL\SQLScripts>set CITYDB=CityModel

c:\temp\3dcitydb-3.2.0\PostgreSQL\SQLScripts>set PGBIN=C:\Program Files (x86)\pg
Admin III\1.22\

c:\temp\3dcitydb-3.2.0\PostgreSQL\SQLScripts>REM cd to path of the shell script


c:\temp\3dcitydb-3.2.0\PostgreSQL\SQLScripts>cd /d c:\temp\3dcitydb-3.2.0\Postgr
eSQL\SQLScripts\

c:\temp\3dcitydb-3.2.0\PostgreSQL\SQLScripts>REM Run CREATE_DB.sql to create the
 3D City Database instance

c:\temp\3dcitydb-3.2.0\PostgreSQL\SQLScripts>"C:\Program Files (x86)\pgAdmin III
\1.22\\psql" -d "CityModel" -f "CREATE_DB.sql"
Default footer is off.
SET

Please enter a valid SRID (e.g., 3068 for DHDN/Soldner Berlin): 25833
Please enter the corresponding SRSName to be used in GML exports (e.g., urn:ogc:
def:crs,crs:EPSG::3068,crs:EPSG::5783): urn:ogc:def:crs,crs:EPSG:6.12:25833,crs:
EPSG:6.12:5783
```

The 3D city database schema will be created and the installation process takes around one minute. After the installation is done, you may see the command window looks in like the screenshot below.



```
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
ALTER DATABASE

3DCityDB creation complete!

Checking spatial reference system ...
 check_srid
------------
 SRID ok


Setting spatial reference system of 3DCityDB instance ...
INSERT 0 1
 change_schema_srid
--------------------


Done

c:\temp\3dcitydb-3.2.0\PostgreSQL\SQLScripts>pause
Drücken Sie eine beliebige Taste . . .
```

□ Exit the script by pressing any key to finish the installation process

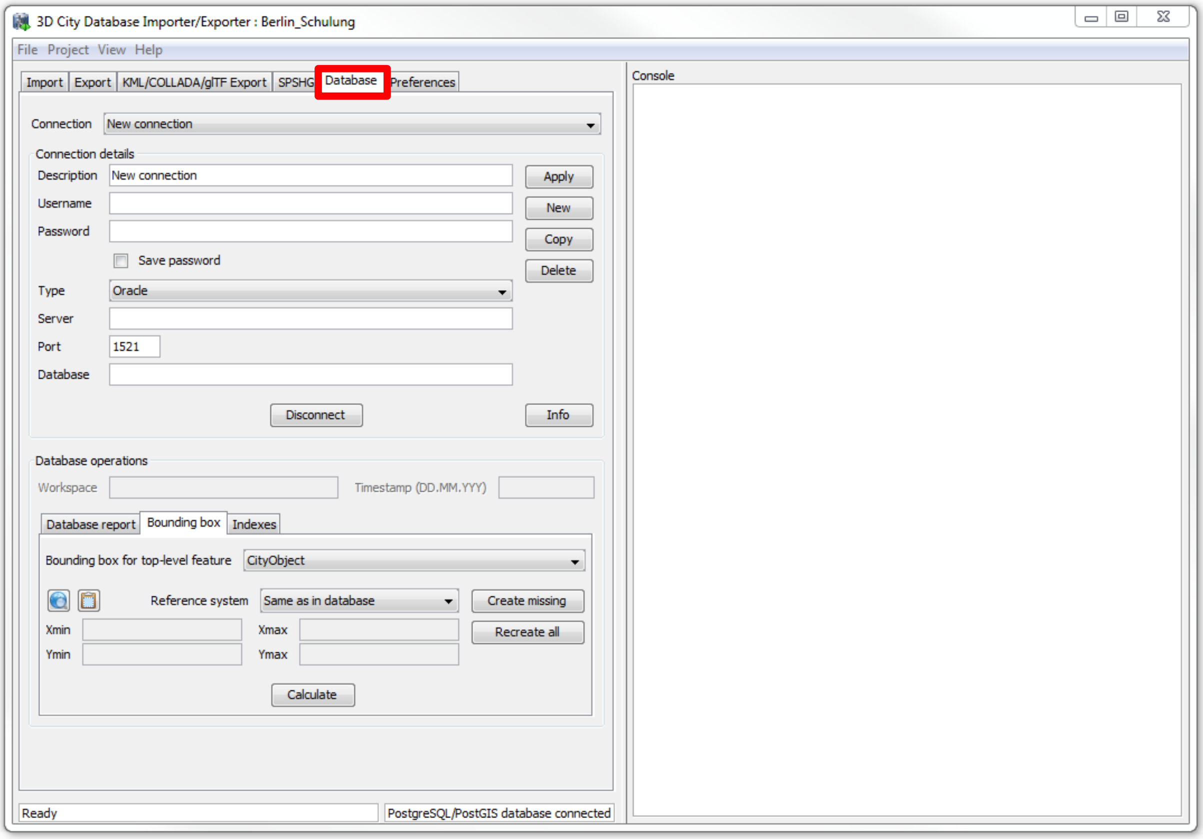
By now, you have successfully set up a 3D city database instance for storing the CityGML data.

Exercise 3: CityGML Import

In this exercise we will learn how to use the Importer/Exporter tool to import a CityGML file into a 3DCityDB instance.

□ Run the Importer/Exporter tool by a double click on the shortcut icon on the desktop

□ After starting up the Importer/Exporter tool, click on the ***Database*** tab to open the Database connection window (see the figure below)



3D City Database Importer/Exporter : Berlin_Schulung

Connection: New connection

Connection details

Description: New connection

Username:

Password:

□ Save password

Type: Oracle

Server:

Port: 1521

Database:

Database operations

Workspace:

Timestamp (DD.MM.YYY):

Bounding box for top-level feature: CityObject

Reference system: Same as in database

Xmin: Xmax:

Ymin: Ymax:

Ready

PostgreSQL/PostGIS database connected

Now, we need to establish the connection to the 3DCityDB instance by giving the required parameters for the database access. Please perform the following operations to complete the database connection:

□ Enter a short description (for example "***Berlin_Schulung***") in the ***Description*** text field

□ Enter your database ***username*** and ***password*** (refer to exercise 2) into the corresponding text fields

□ Mark the checkbox ***Save password***

> **Note:** The password will be stored in a configuration file. When you restart the Importer/Exporter tool the stored password will be reloaded and you don't need to type the password again. Please note, that the password will be stored as plain text in the configuration file. In the case of high security needs, it is therefore recommended to uncheck this option.

□ Click the dropdown list ***Type*** and select ***PostreSQL/PostGIS*** to specify the database type

□ Enter the database server name or IP address ("***127.0.0.1***") into the ***Server*** text field

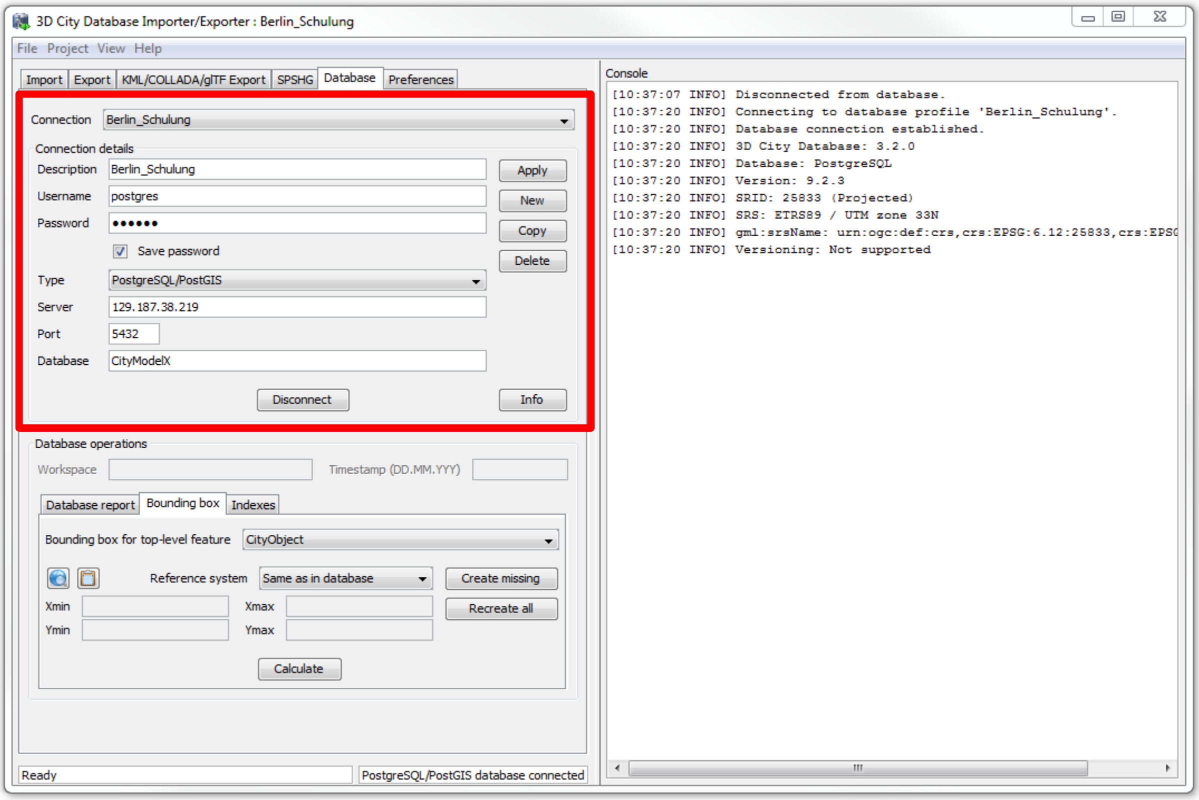
□ Enter the port number ("***5432***") of the database into the ***Port*** text field

□ Enter the database name ("**CityModelX**") into the **Database** text filed

□ Click the **Connect** button to establish the database connection

If the required parameters have been correctly entered, the information of the connected 3DCityDB instance will be printed in the console window on the right side of the Importer/Exporter user interface that looks like the figure below.

> **Note:** In case the connection could not be established, the corresponding error messages will be printed to the console window to help in identifying the cause of the connection problem.



3D City Database Importer/Exporter : Berlin_Schulung

Connection: Berlin_Schulung

Connection details

Description: Berlin_Schulung

Username: postgres

Password: ••••••

☑ Save password

Type: PostgreSQL/PostGIS

Server: 129.187.38.219

Port: 5432

Database: CityModelX

Bounding box for top-level feature: CityObject

Reference system: Same as in database

Ready | PostgreSQL/PostGIS database connected

Console

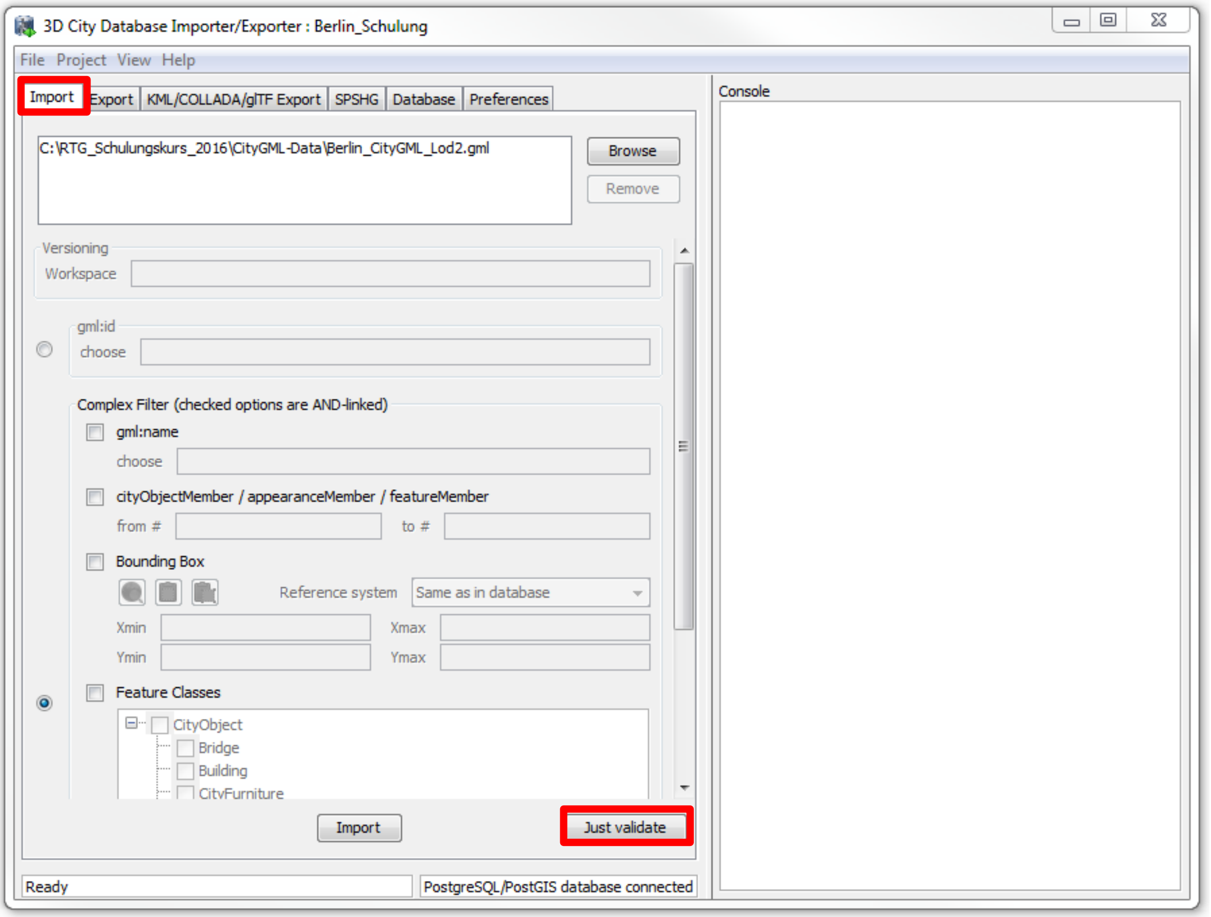
```
[10:37:07 INFO] Disconnected from database.
[10:37:20 INFO] Connecting to database profile 'Berlin_Schulung'.
[10:37:20 INFO] Database connection established.
[10:37:20 INFO] 3D City Database: 3.2.0
[10:37:20 INFO] Database: PostgreSQL
[10:37:20 INFO] Version: 9.2.3
[10:37:20 INFO] SRID: 25833 (Projected)
[10:37:20 INFO] SRS: ETRS89 / UTM zone 33N
[10:37:20 INFO] gml:srsName: urn:ogc:def:crs,crs:EPSG:6.12:25833,crs:EPS
[10:37:20 INFO] Versioning: Not supported
```

Before importing, the CityGML files should be **validated** against the CityGML XML schema. Please follow the steps below to complete the validation process.

□ Click on the ***Import*** tab to activate the CityGML-Importer window (see the following screenshot)

□ Click the ***Browse*** button to open a file selection dialog and select the CityGML file to be imported. The CityGML file used in this tutorial is located in the path below:

"***C:\RTG_Schulungskurs_2016\CityGML-Data\ Berlin_CityGML_Lod2.gml***"



3D City Database Importer/Exporter : Berlin_Schulung

C:\RTG_Schulungskurs_2016\CityGML-Data\Berlin_CityGML_Lod2.gml

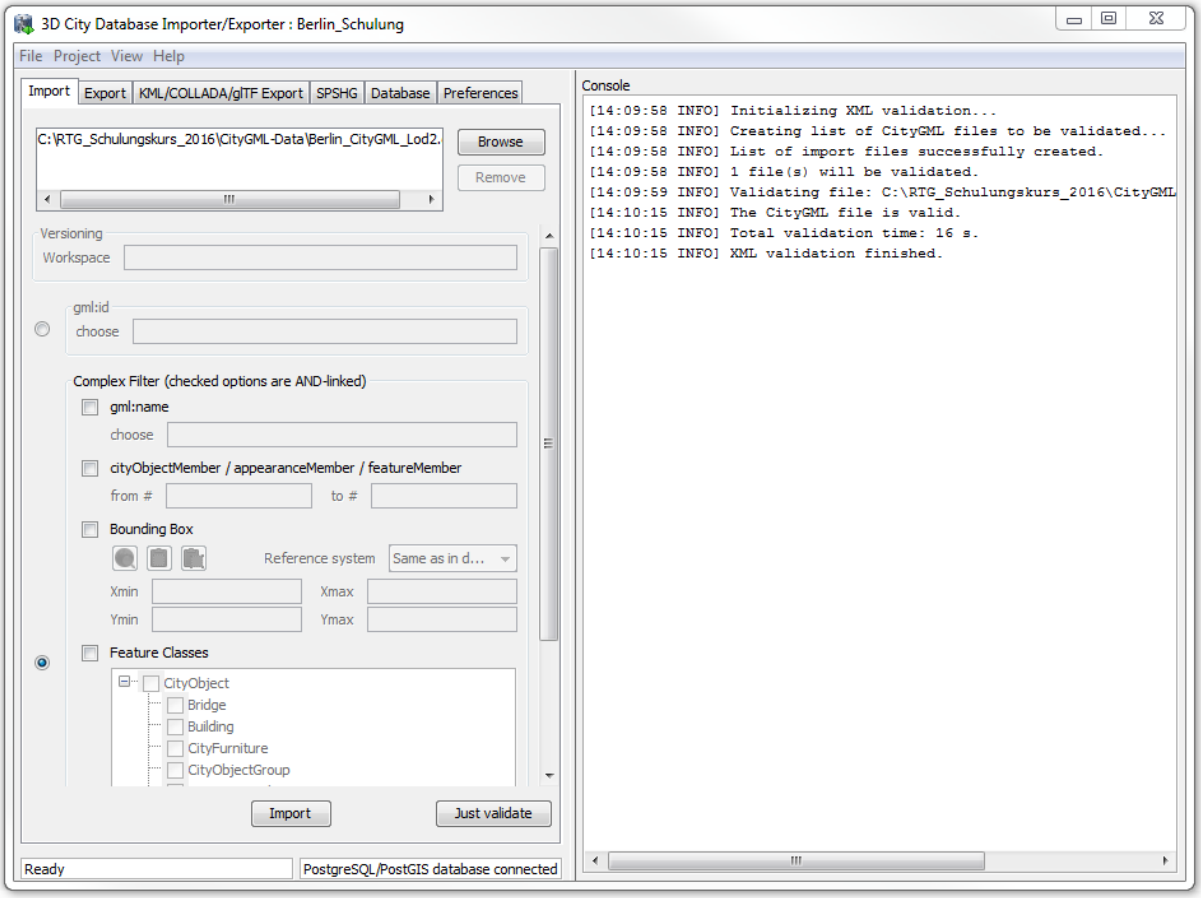
Reference system: Same as in database

Ready

PostgreSQL/PostGIS database connected

□ Click on the ***Just validate*** button to run the validation process.

After finishing the validation process the validation results will be printed to the console window (see the following figure).



3D City Database Importer/Exporter : Berlin_Schulung

C:\RTG_Schulungskurs_2016\CityGML-Data\Berlin_CityGML_Lod2.

Reference system: Same as in d...

Ready

PostgreSQL/PostGIS database connected

```
[14:09:58 INFO] Initializing XML validation...
[14:09:58 INFO] Creating list of CityGML files to be validated...
[14:09:58 INFO] List of import files successfully created.
[14:09:58 INFO] 1 file(s) will be validated.
[14:09:59 INFO] Validating file: C:\RTG_Schulungskurs_2016\CityGML
[14:10:15 INFO] The CityGML file is valid.
[14:10:15 INFO] Total validation time: 16 s.
[14:10:15 INFO] XML validation finished.
```

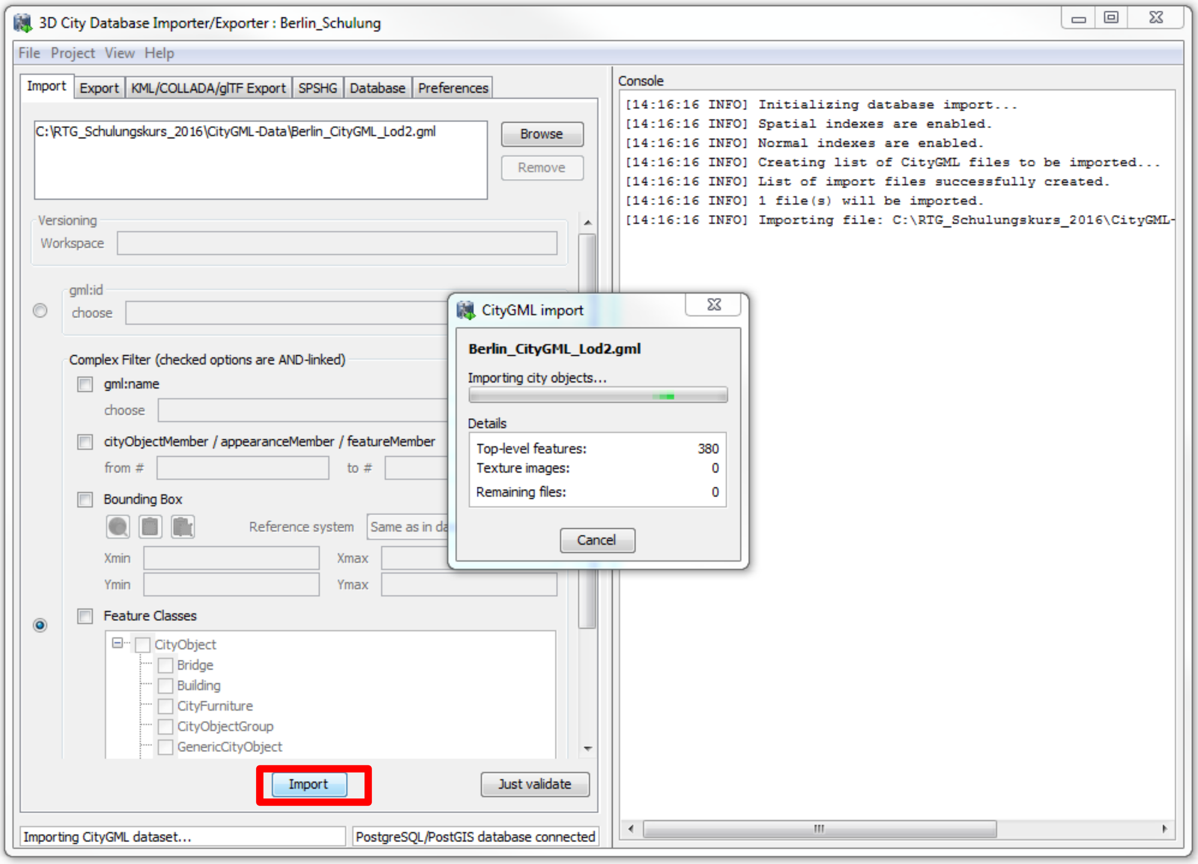
Note: It is strongly recommended that only CityGML files which have successfully passed the validation are imported into the database. Otherwise, errors in the data may lead to unexpected behavior or abnormal termination.

Having successfully completed the validation process you can start the import process.

Note: The CityGML-Importer allows for setting thematic and spatial filter criteria to constrain the set of CityGML top-level features (such as Building, Bridge, CityFurniture, Waterboly etc.). For example, you can choose the gml:id filter to only import the CityGML features which have to be matched the entered gmlids. More complex filters such as boundingbox-filter, FeatureClass-filter can also be applied if the corresponding checkboxes are activated (For more details please refer to the 3DCityDB-Documentation). If no checkbox is enabled, no filter criteria are applied and thus all CityGML features contained in the input CityGML files will be imported

- Click the ***Import*** button to start the import process

A status window is shows the import progress (see the following figure). The import process can be aborted at any time by pressing the ***Cancel*** button with the status window.



C:\RTG_Schulungskurs_2016\CityGML-Data\Berlin_CityGML_Lod2.gml

```
[14:16:16 INFO] Initializing database import...
[14:16:16 INFO] Spatial indexes are enabled.
[14:16:16 INFO] Normal indexes are enabled.
[14:16:16 INFO] Creating list of CityGML files to be imported...
[14:16:16 INFO] List of import files successfully created.
[14:16:16 INFO] 1 file(s) will be imported.
[14:16:16 INFO] Importing file: C:\RTG_Schulungskurs_2016\CityGML-
```

Berlin_CityGML_Lod2.gml

Importing city objects...

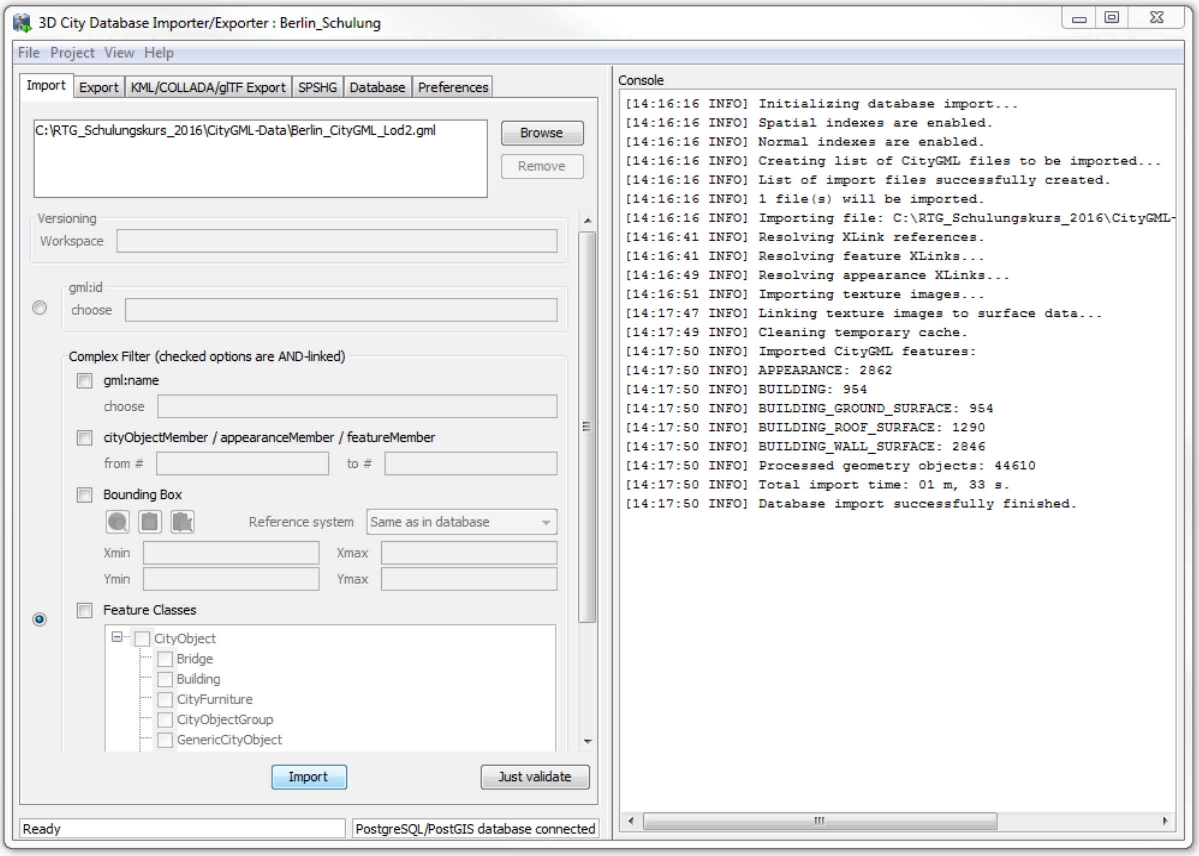
Details

Top-level features:	380
Texture images:	0
Remaining files:	0

Importing CityGML dataset...

PostgreSQL/PostGIS database connected

After completing the import, a summary list enumerating the types and number of the imported CityGML top-level features will be printed in the console window (see the figure below).



C:\RTG_Schulungskurs_2016\CityGML-Data\Berlin_CityGML_Lod2.gml

Ready | PostgreSQL/PostGIS database connected

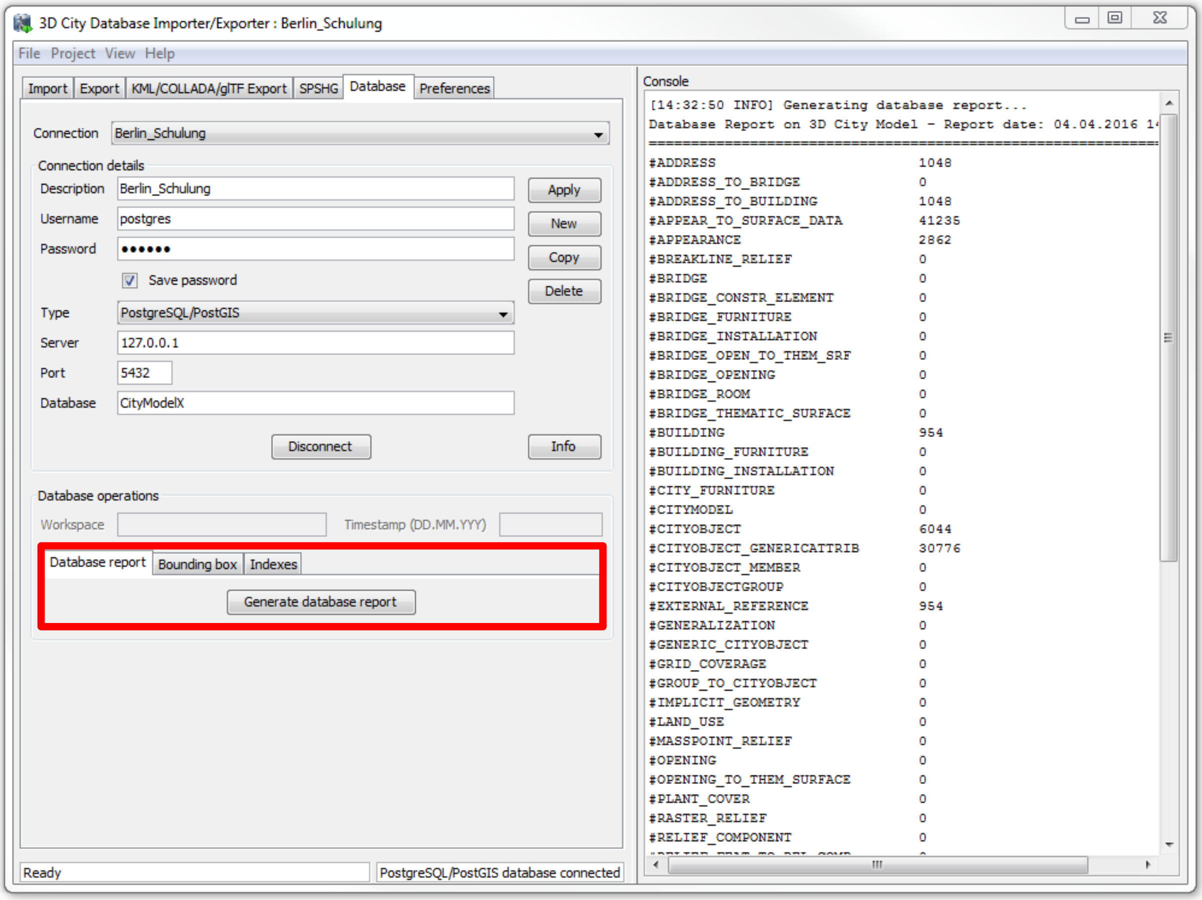
```
[14:16:16 INFO] Initializing database import...
[14:16:16 INFO] Spatial indexes are enabled.
[14:16:16 INFO] Normal indexes are enabled.
[14:16:16 INFO] Creating list of CityGML files to be imported...
[14:16:16 INFO] List of import files successfully created.
[14:16:16 INFO] 1 file(s) will be imported.
[14:16:16 INFO] Importing file: C:\RTG_Schulungskurs_2016\CityGML-
[14:16:41 INFO] Resolving XLink references.
[14:16:41 INFO] Resolving feature XLinks...
[14:16:49 INFO] Resolving appearance XLinks...
[14:16:51 INFO] Importing texture images...
[14:17:47 INFO] Linking texture images to surface data...
[14:17:49 INFO] Cleaning temporary cache.
[14:17:50 INFO] Imported CityGML features:
[14:17:50 INFO] APPEARANCE: 2862
[14:17:50 INFO] BUILDING: 954
[14:17:50 INFO] BUILDING_GROUND_SURFACE: 954
[14:17:50 INFO] BUILDING_ROOF_SURFACE: 1290
[14:17:50 INFO] BUILDING_WALL_SURFACE: 2846
[14:17:50 INFO] Processed geometry objects: 44610
[14:17:50 INFO] Total import time: 01 m, 33 s.
[14:17:50 INFO] Database import successfully finished.
```

By now, the CityGML file should have been successfully imported into the database. You can use the Importer/Exporter tool to verify the imported CityGML contents in the database.

□ Click on the ***Database*** tab to go back to the database connection window

□ Activate the ***Database report*** tab, and click the ***Generate database report*** button.

A list of all tables of the 3D City Database together with their total number of rows will be printed to the console window (see the figure below).



Connection: Berlin_Schulung

Description: Berlin_Schulung

Username: postgres

Password: ••••••

☑ Save password

Type: PostgreSQL/PostGIS

Server: 127.0.0.1

Port: 5432

Database: CityModelX

Workspace:

Timestamp (DD.MM.YYY):

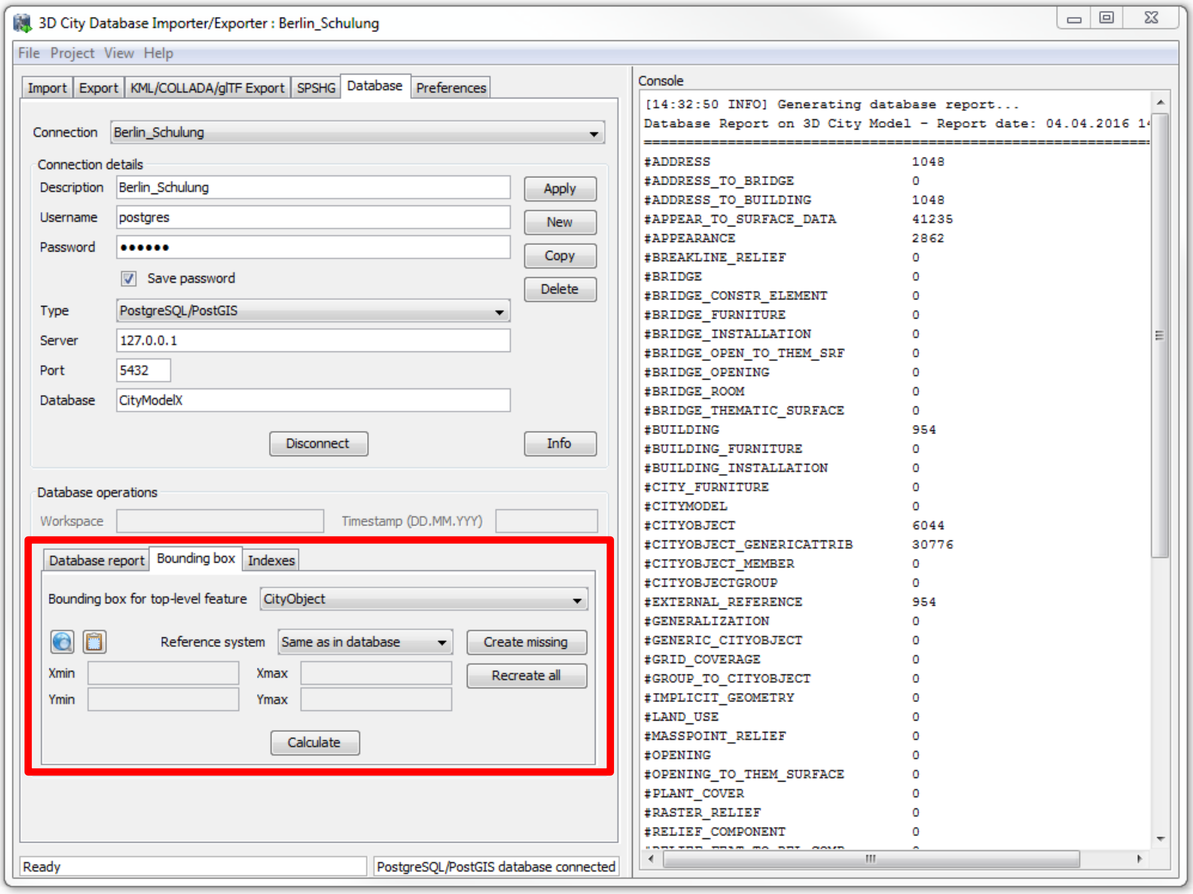
Ready

PostgreSQL/PostGIS database connected

```
[14:32:50 INFO] Generating database report...
Database Report on 3D City Model - Report date: 04.04.2016 1
==============================================================
#ADDRESS                        1048
#ADDRESS_TO_BRIDGE              0
#ADDRESS_TO_BUILDING            1048
#APPEAR_TO_SURFACE_DATA         41235
#APPEARANCE                     2862
#BREAKLINE_RELIEF               0
#BRIDGE                         0
#BRIDGE_CONSTR_ELEMENT          0
#BRIDGE_FURNITURE               0
#BRIDGE_INSTALLATION            0
#BRIDGE_OPEN_TO_THEM_SRF        0
#BRIDGE_OPENING                 0
#BRIDGE_ROOM                    0
#BRIDGE_THEMATIC_SURFACE        0
#BUILDING                       954
#BUILDING_FURNITURE             0
#BUILDING_INSTALLATION          0
#CITY_FURNITURE                 0
#CITYMODEL                      0
#CITYOBJECT                     6044
#CITYOBJECT_GENERICATTRIB       30776
#CITYOBJECT_MEMBER              0
#CITYOBJECTGROUP                0
#EXTERNAL_REFERENCE             954
#GENERALIZATION                 0
#GENERIC_CITYOBJECT             0
#GRID_COVERAGE                  0
#GROUP_TO_CITYOBJECT            0
#IMPLICIT_GEOMETRY              0
#LAND_USE                       0
#MASSPOINT_RELIEF               0
#OPENING                        0
#OPENING_TO_THEM_SURFACE        0
#PLANT_COVER                    0
#RASTER_RELIEF                  0
#RELIEF_COMPONENT               0
```

The spatial extent of the imported CityGML objects can also be easily determined by using the Importer/Exporter Tool:

□ Activate the ***Bounding box*** tab, and click the ***Calculate*** button (see the following screenshot).



Connection: Berlin_Schulung

Connection details

Description: Berlin_Schulung

Username: postgres

Password: ••••••

☑ Save password

Type: PostgreSQL/PostGIS

Server: 127.0.0.1

Port: 5432

Database: CityModelX

Database operations

Workspace:

Timestamp (DD.MM.YYY):

Bounding box for top-level feature: CityObject

Reference system: Same as in database

Xmin:

Xmax:

Ymin:

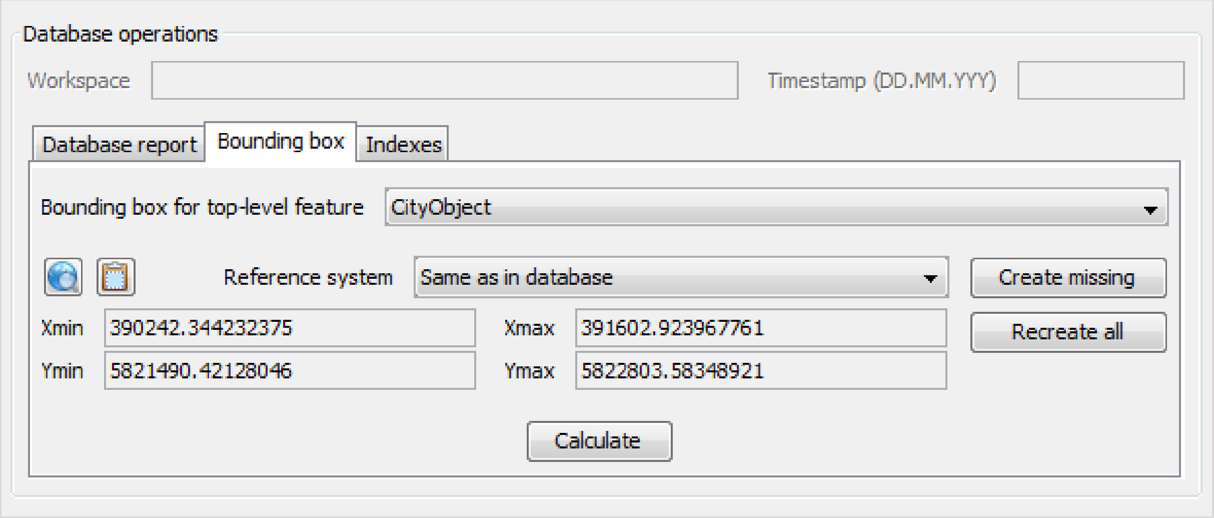
Ymax:

Ready

PostgreSQL/PostGIS database connected

```
[14:32:50 INFO] Generating database report...
Database Report on 3D City Model - Report date: 04.04.2016 1
================================================================
#ADDRESS                        1048
#ADDRESS_TO_BRIDGE              0
#ADDRESS_TO_BUILDING            1048
#APPEAR_TO_SURFACE_DATA         41235
#APPEARANCE                     2862
#BREAKLINE_RELIEF               0
#BRIDGE                         0
#BRIDGE_CONSTR_ELEMENT          0
#BRIDGE_FURNITURE               0
#BRIDGE_INSTALLATION            0
#BRIDGE_OPEN_TO_THEM_SRF        0
#BRIDGE_OPENING                 0
#BRIDGE_ROOM                    0
#BRIDGE_THEMATIC_SURFACE        0
#BUILDING                       954
#BUILDING_FURNITURE             0
#BUILDING_INSTALLATION          0
#CITY_FURNITURE                 0
#CITYMODEL                      0
#CITYOBJECT                     6044
#CITYOBJECT_GENERICATTRIB       30776
#CITYOBJECT_MEMBER              0
#CITYOBJECTGROUP                0
#EXTERNAL_REFERENCE             954
#GENERALIZATION                 0
#GENERIC_CITYOBJECT             0
#GRID_COVERAGE                  0
#GROUP_TO_CITYOBJECT            0
#IMPLICIT_GEOMETRY              0
#LAND_USE                       0
#MASSPOINT_RELIEF               0
#OPENING                        0
#OPENING_TO_THEM_SURFACE        0
#PLANT_COVER                    0
#RASTER_RELIEF                  0
#RELIEF_COMPONENT               0
```

The coordinate values of the lower left (Xmin, Ymin) and upper right (Xmax, Ymax) corner of the calculated bounding box are printed in the coordinate value fields of the dialog (see the screenshot below).



Database operations

Workspace:

Timestamp (DD.MM.YYY):

Bounding box for top-level feature: CityObject

Reference system: Same as in database

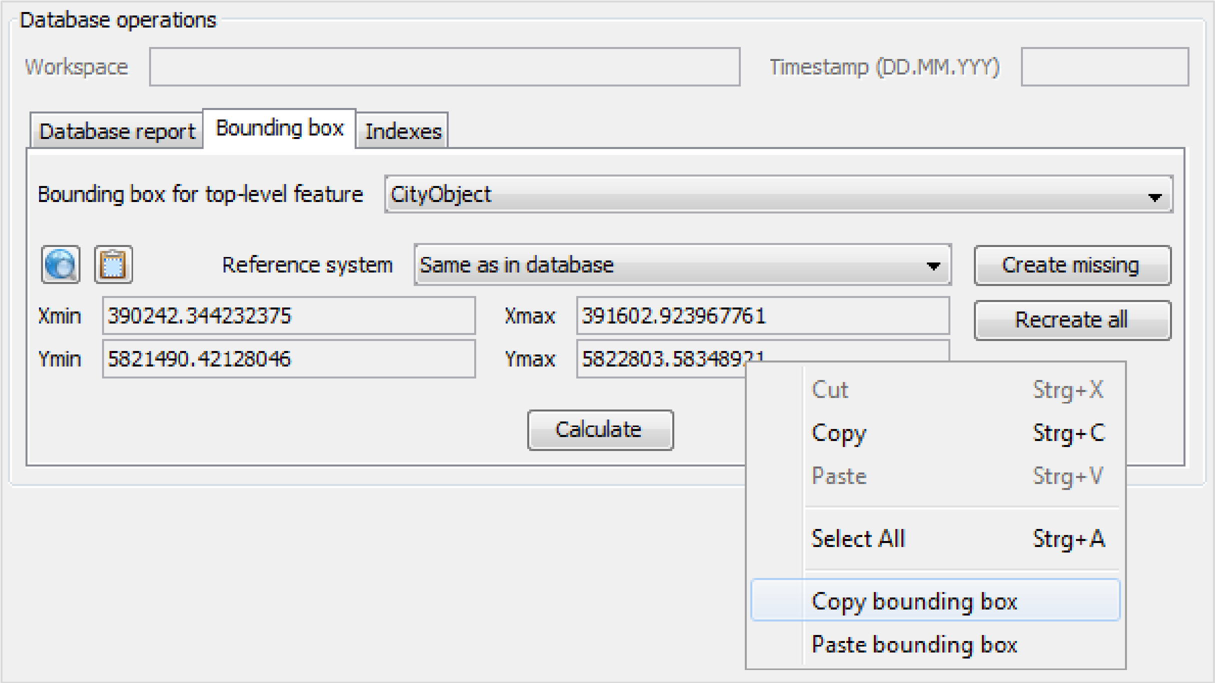
Xmin: 390242.344232375

Xmax: 391602.923967761

Ymin: 5821490.42128046

Ymax: 5822803.58348921

You can copy the coordinate values to the clipboard by clicking the 📋 button, or by right-clicking on a coordinate value field and choosing the ***Copy bounding box*** option from the context menu (see the following figure)**.** This function will be used in the following Exercises 5 and 6).



Bounding box for top-level feature: CityObject

Reference system: Same as in database

Xmin: 390242.344232375

Xmax: 391602.923967761

Ymin: 5821490.42128046

Ymax: 5822803.58348921

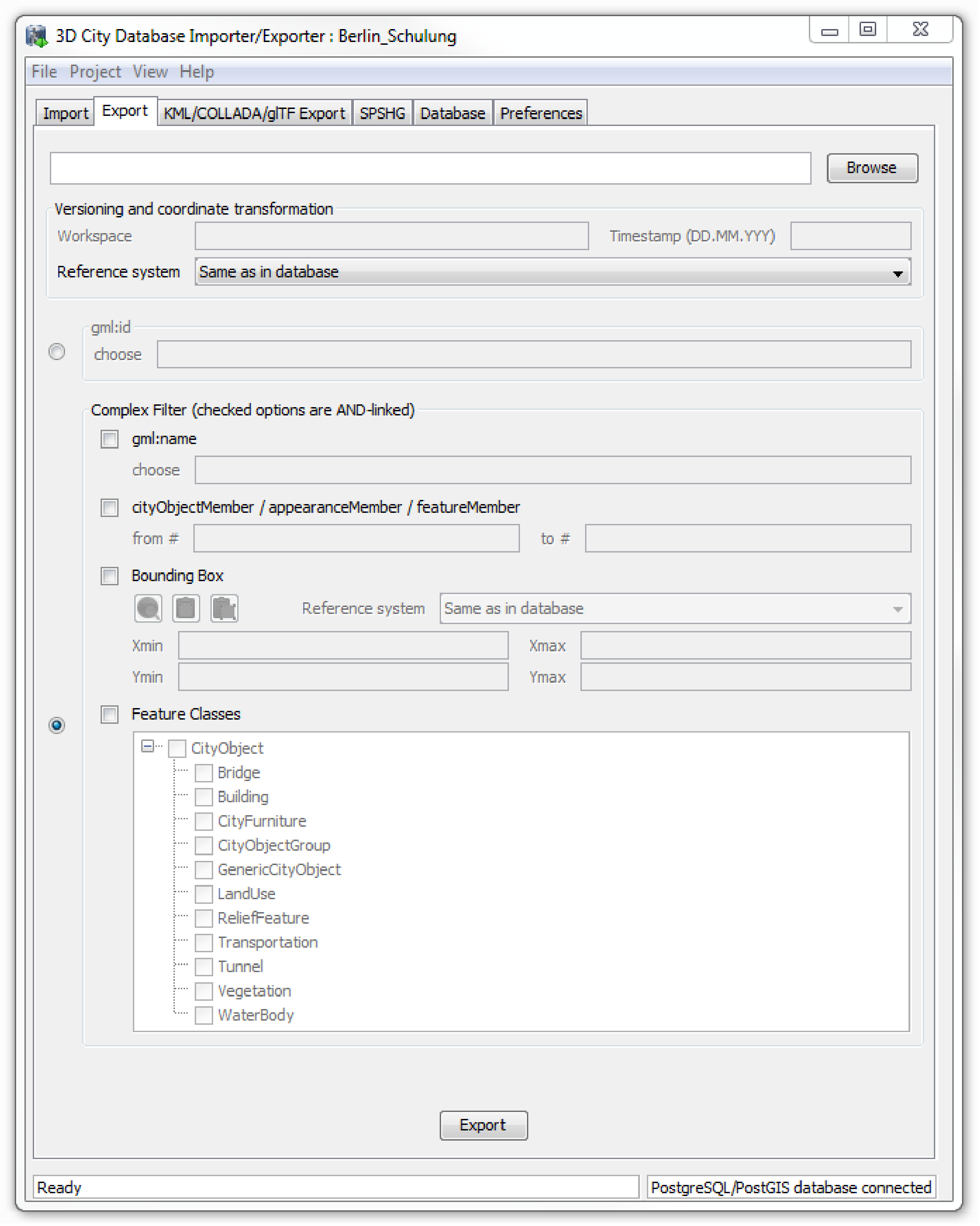
Exercise 4: CityGML Export

In this exercise you will learn how to export a CityGML file from the 3D city database by using the Importer/Exporter tool along with its filter abilities.

Exercise 4.1 – Export a single CityGML feature using its gmlid

Each CityGML top-level feature (e.g. Building) has been assigned a unique gmlid value which allows applications to consistently maintain CityGML models within GIS and spatial database management systems. Thus, it is possible to trace and export the desired CityGML top-level feature by using its gmlid.

□ Click on the ***Export*** tab to open the CityGML-Exporter window (see the screenshot below)



3D City Database Importer/Exporter : Berlin_Schulung

Reference system: Same as in database

Status: Ready | PostgreSQL/PostGIS database connected

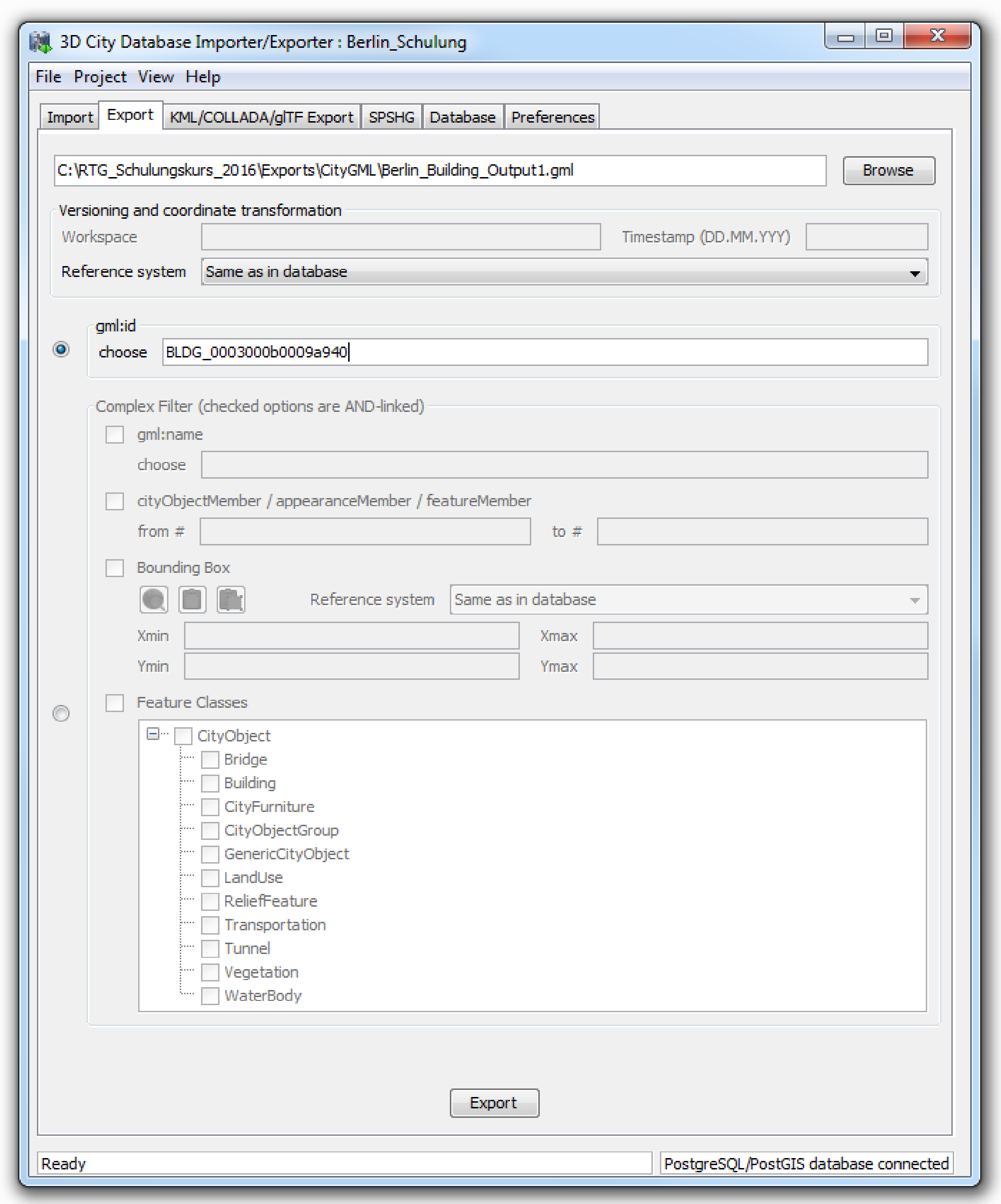
□ Click the ***Browse*** button to open a file selection dialog and specify the path and filename for the output CityGML file.

“***C:\RTG_Schulungskurs_2016\Exports\CityGML\Berlin_Building_Output1.gml***”

□ Activate the first radio button (marked in the following figure)

□ Enter the text “***BLDG_0003000b0009a940***” into the ***choose*** text field

> **Note**: You can select more than one CityGML feature by adding further gml:ids into the list but limited to **1000** gml:ids. The entered gml:ids muss be **comma-separated**.



3D City Database Importer/Exporter : Berlin_Schulung

C:\RTG_Schulungskurs_2016\Exports\CityGML\Berlin_Building_Output1.gml

Versioning and coordinate transformation
Workspace
Timestamp (DD.MM.YYY)
Reference system: Same as in database

● gml:id
choose: BLDG_0003000b0009a940

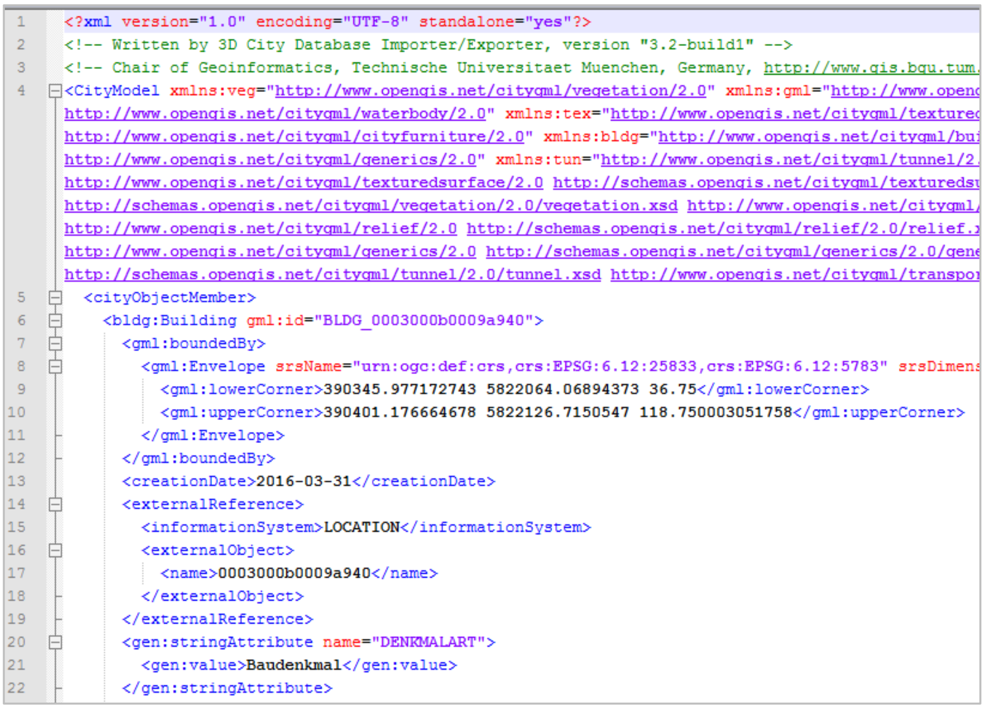
○ Complex Filter (checked options are AND-linked)
□ gml:name
choose
□ cityObjectMember / appearanceMember / featureMember
from # to #
□ Bounding Box
Reference system: Same as in database
Xmin Xmax
Ymin Ymax
□ Feature Classes
□ CityObject
□ Bridge
□ Building
□ CityFurniture
□ CityObjectGroup
□ GenericCityObject
□ LandUse
□ ReliefFeature
□ Transportation
□ Tunnel
□ Vegetation
□ WaterBody

Ready
PostgreSQL/PostGIS database connected

□ Click ***Export*** button to start the export process

After completing the export process, the CityGML top-level features matching the entered gml:id values are exported.

You may check the content the of exported CityGML file by opening it using a text editor e.g. Notepad++ (see the following figure).



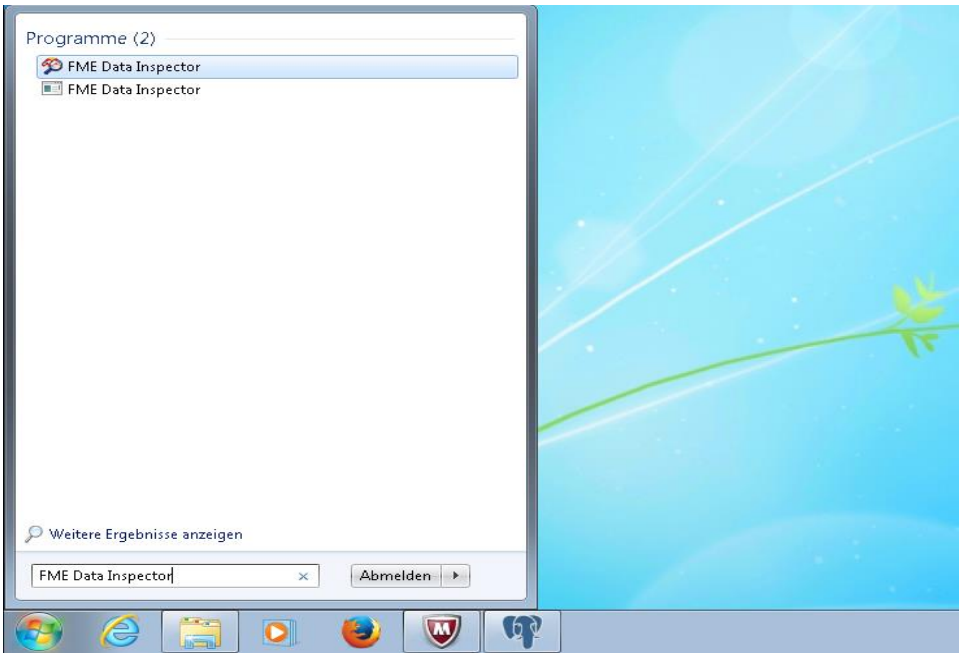
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Written by 3D City Database Importer/Exporter, version "3.2-build1" -->
<!-- Chair of Geoinformatics, Technische Universitaet Muenchen, Germany, http://www.gis.bgu.tum.
<CityModel xmlns:veg="http://www.opengis.net/citygml/vegetation/2.0" xmlns:gml="http://www.openg
http://www.opengis.net/citygml/waterbody/2.0" xmlns:tex="http://www.opengis.net/citygml/texturec
http://www.opengis.net/citygml/cityfurniture/2.0" xmlns:bldg="http://www.opengis.net/citygml/bui
http://www.opengis.net/citygml/generics/2.0" xmlns:tun="http://www.opengis.net/citygml/tunnel/2.
http://www.opengis.net/citygml/texturedsurface/2.0 http://schemas.opengis.net/citygml/texturedsu
http://schemas.opengis.net/citygml/vegetation/2.0/vegetation.xsd http://www.opengis.net/citygml/
http://www.opengis.net/citygml/relief/2.0 http://schemas.opengis.net/citygml/relief/2.0/relief.x
http://www.opengis.net/citygml/generics/2.0 http://schemas.opengis.net/citygml/generics/2.0/gene
http://schemas.opengis.net/citygml/tunnel/2.0/tunnel.xsd http://www.opengis.net/citygml/transpor
  <cityObjectMember>
    <bldg:Building gml:id="BLDG_0003000b0009a940">
      <gml:boundedBy>
        <gml:Envelope srsName="urn:ogc:def:crs,crs:EPSG:6.12:25833,crs:EPSG:6.12:5783" srsDimens
          <gml:lowerCorner>390345.977172743 5822064.06894373 36.75</gml:lowerCorner>
          <gml:upperCorner>390401.176664678 5822126.7150547 118.750003051758</gml:upperCorner>
        </gml:Envelope>
      </gml:boundedBy>
      <creationDate>2016-03-31</creationDate>
      <externalReference>
        <informationSystem>LOCATION</informationSystem>
        <externalObject>
          <name>0003000b0009a940</name>
        </externalObject>
      </externalReference>
      <gen:stringAttribute name="DENKMALART">
        <gen:value>Baudenkmal</gen:value>
      </gen:stringAttribute>
```

Optionally, you can also visually inspect the CityGML data using a CityGML viewer like FME Data Inspector. Alternative (Open Source) CityGML viewers can be found under the following web link:

http://www.citygmlwiki.org/index.php/Freeware

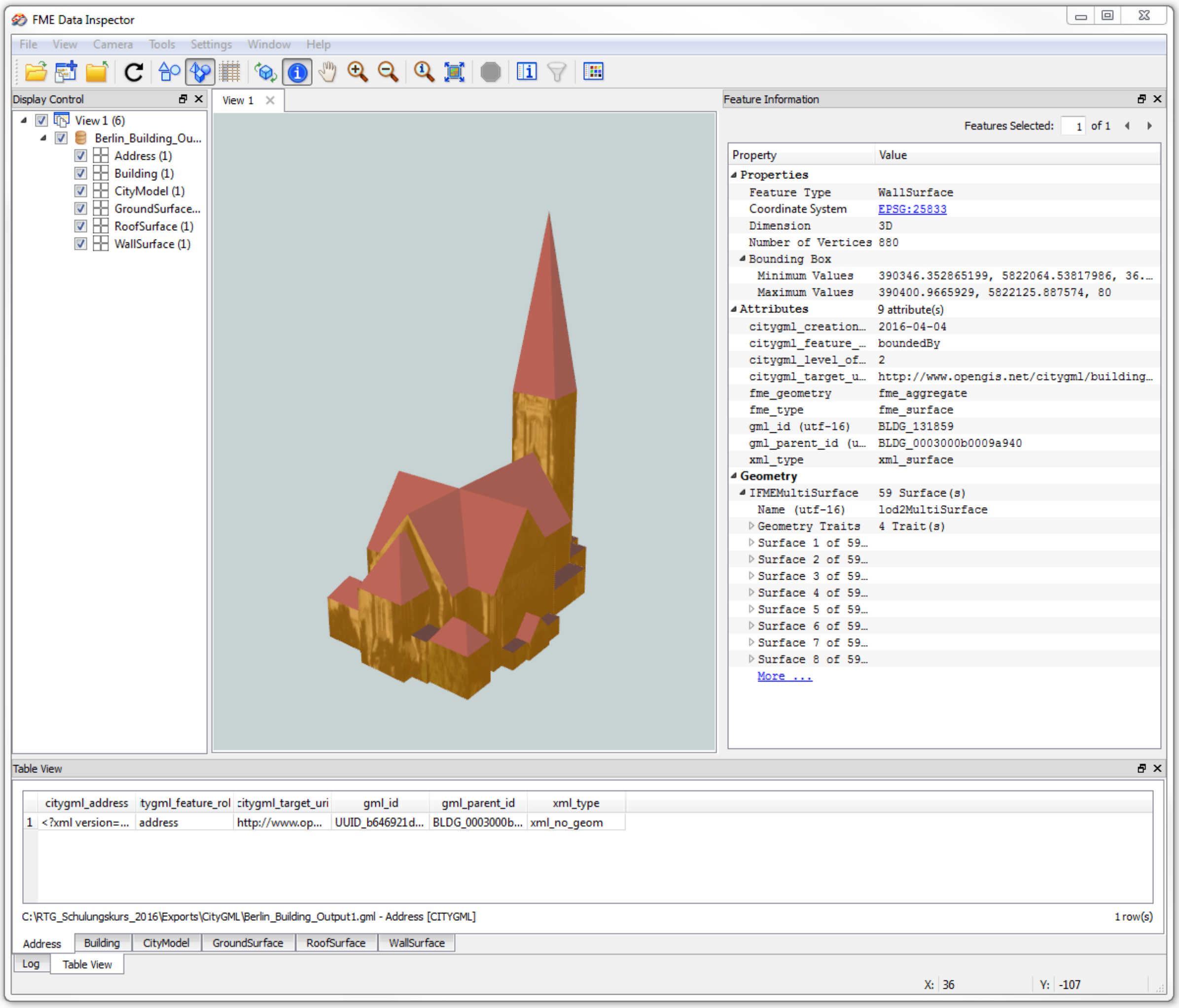
□ Click on the windows ***Start Menu*** button on the desktop

□ Search the program "**FME Data Inspector**" (see the screenshot below)



FME Data Inspector

- Run the ***FME Data Inspector*** program, and load the exported CityGML file through the menu bar ***File → Open Dataset…***



Features Selected: 1 of 1

Property	Value
Properties	
Feature Type	WallSurface
Coordinate System	EPSG:25833
Dimension	3D
Number of Vertices	880
Bounding Box	
Minimum Values	390346.352865199, 5822064.53817986, 36.…
Maximum Values	390400.9665929, 5822125.887574, 80
Attributes	9 attribute(s)
citygml_creation…	2016-04-04
citygml_feature_…	boundedBy
citygml_level_of…	2
citygml_target_u…	http://www.opengis.net/citygml/building…
fme_geometry	fme_aggregate
fme_type	fme_surface
gml_id (utf-16)	BLDG_131859
gml_parent_id (u…	BLDG_0003000b0009a940
xml_type	xml_surface
Geometry	
IFMEMultiSurface	59 Surface(s)
Name (utf-16)	lod2MultiSurface
Geometry Traits	4 Trait(s)
Surface 1 of 59…	
Surface 2 of 59…	
Surface 3 of 59…	
Surface 4 of 59…	
Surface 5 of 59…	
Surface 6 of 59…	
Surface 7 of 59…	
Surface 8 of 59…	
More ...	

	citygml_address	tygml_feature_rol	citygml_target_uri	gml_id	gml_parent_id	xml_type
1	<?xml version=...	address	http://www.op...	UUID_b646921d...	BLDG_0003000b...	xml_no_geom

C:\RTG_Schulungskurs_2016\Exports\CityGML\Berlin_Building_Output1.gml - Address [CITYGML] 1 row(s)

X: 36 Y: -107

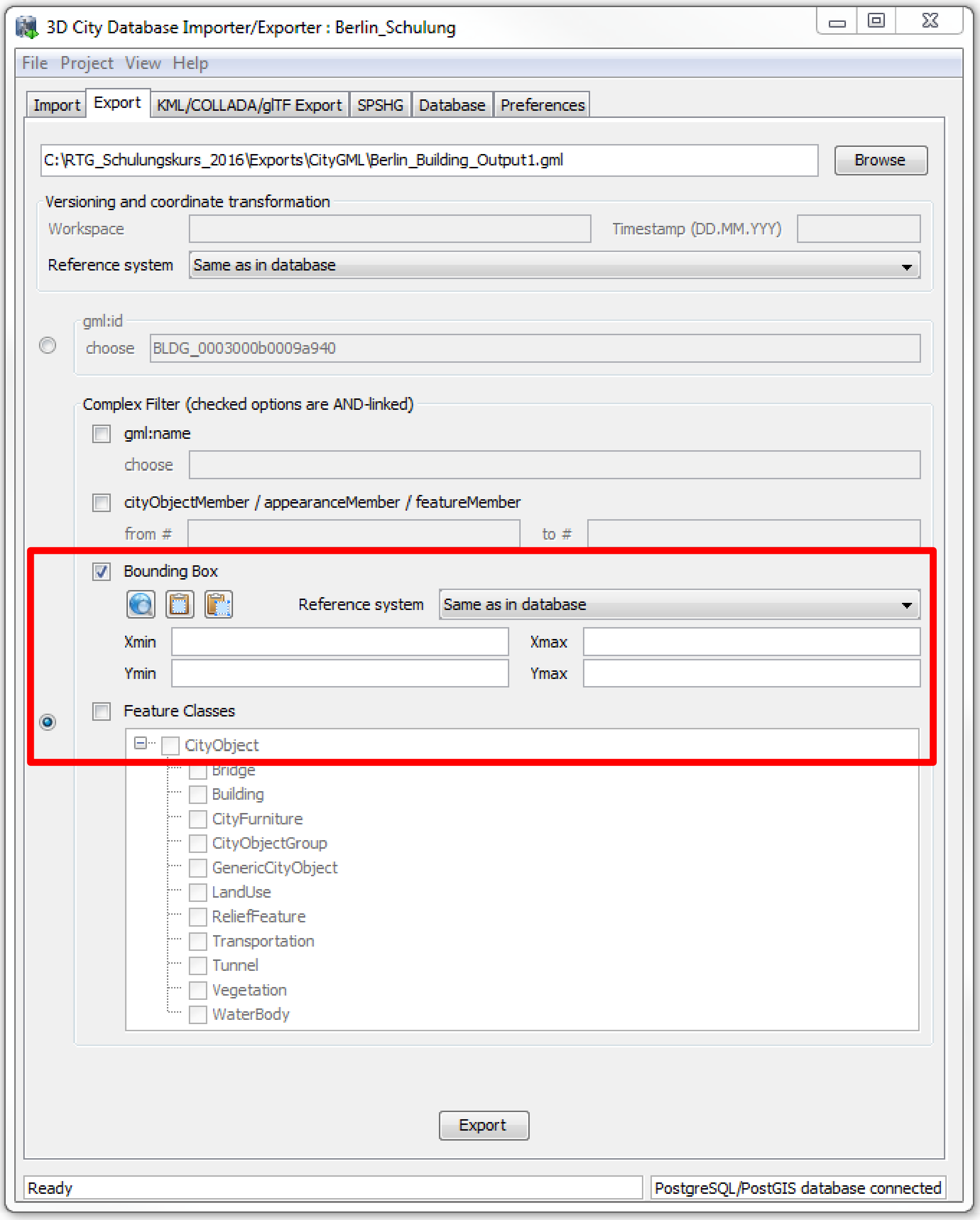
Exercise 4.2 – Export a group of CityGML features using a Bounding Box

□ Click the ***Browse*** button to open a file selection dialog and specify the path and filename of the output CityGML file:

“***C:\RTG_Schulungskurs_2016\Exports\CityGML\Berlin_Building_Output2.gml***”

□ Activate the second radio button (see the screenshot below).

□ Activate the ***Bounding Box*** checkbox (see the screenshot below).



3D City Database Importer/Exporter : Berlin_Schulung

C:\RTG_Schulungskurs_2016\Exports\CityGML\Berlin_Building_Output1.gml

Versioning and coordinate transformation

Workspace

Timestamp (DD.MM.YYY)

Reference system: Same as in database

○ gml:id

choose: BLDG_0003000b0009a940

● Complex Filter (checked options are AND-linked)

□ gml:name

choose

□ cityObjectMember / appearanceMember / featureMember

from # to #

☑ Bounding Box

Reference system: Same as in database

Xmin Xmax

Ymin Ymax

□ Feature Classes

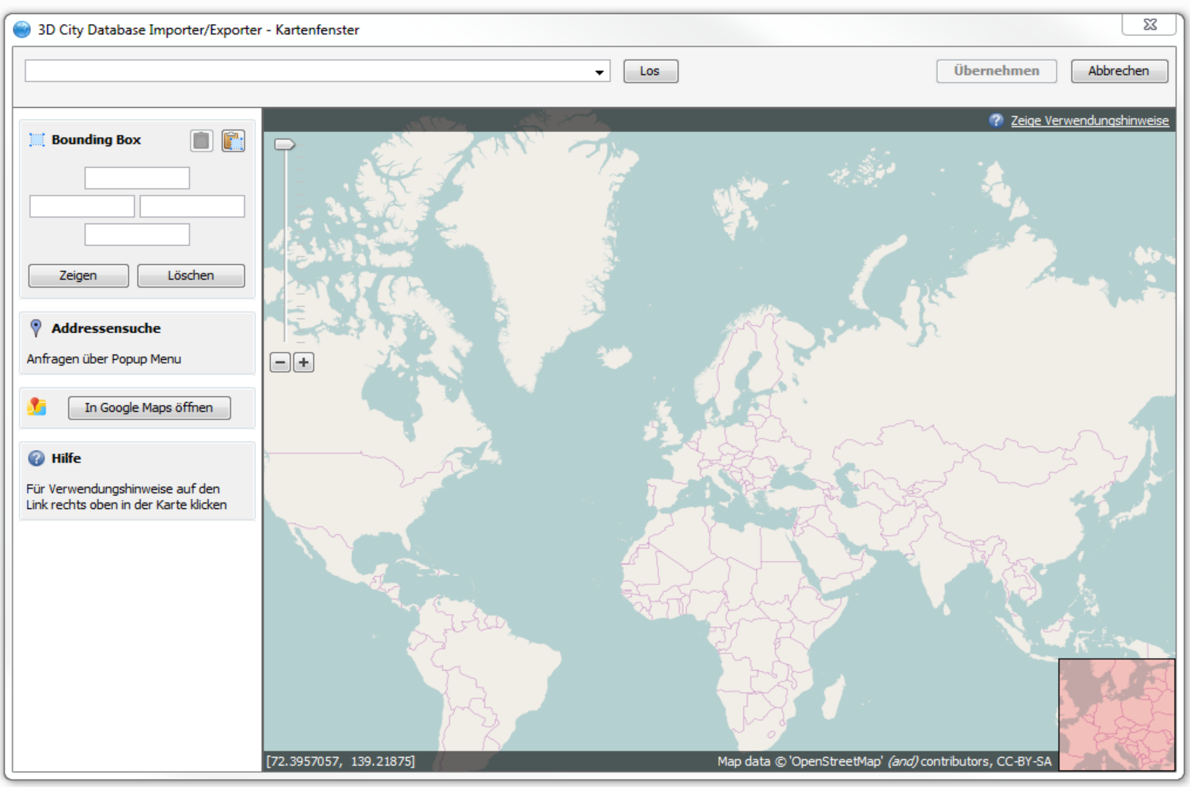
- □ CityObject
  - □ Bridge
  - □ Building
  - □ CityFurniture
  - □ CityObjectGroup
  - □ GenericCityObject
  - □ LandUse
  - □ ReliefFeature
  - □ Transportation
  - □ Tunnel
  - □ Vegetation
  - □ WaterBody

Ready

PostgreSQL/PostGIS database connected

□ Click the map button

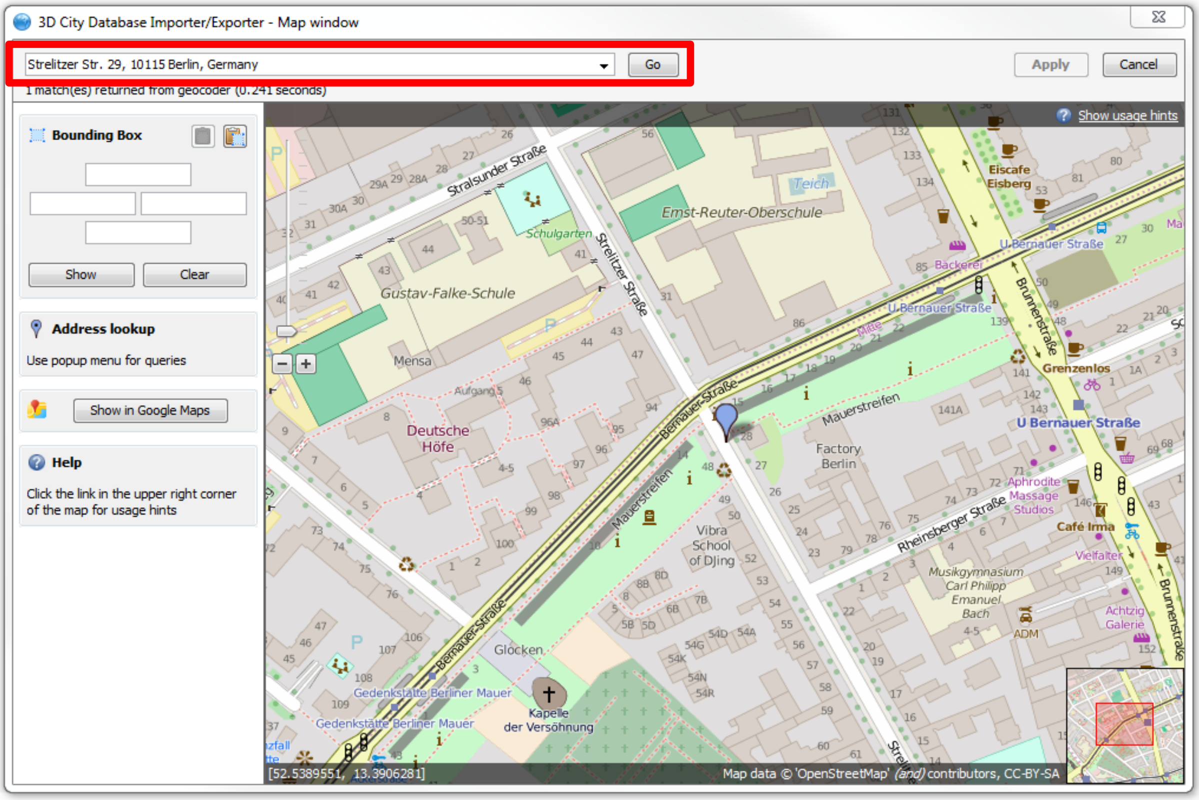
A map window (see the figure below) will appear that allows you to interactively select a 2D bounding box as a spatial filter for the CityGML export.


3D City Database Importer/Exporter - Kartenfenster
Los
Übernehmen
Abbrechen
Bounding Box
Zeigen
Löschen
Addressensuche
Anfragen über Popup Menu
In Google Maps öffnen
Hilfe
Für Verwendungshinweise auf den Link rechts oben in der Karte klicken
Zeige Verwendungshinweise
[72.3957057, 139.21875]
Map data © 'OpenStreetMap' (and) contributors, CC-BY-SA

In this exercise, we would like to export some buildings near to Berlin "*Strelitzer Straße*"

□ Enter the address "**Strelitzer Strasse 29, Berlin**" into the text field on the top left corner

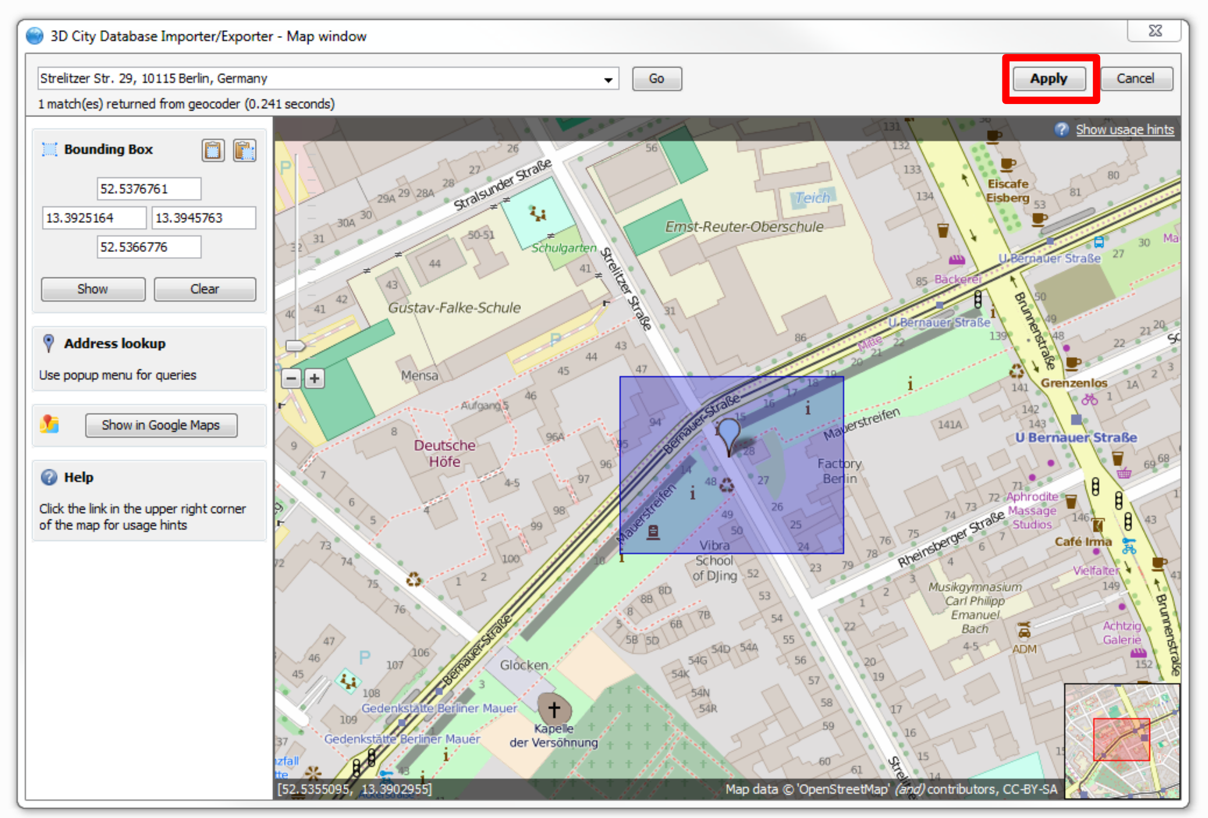
□ Click the ***Go*** button, the 2D Map will be zoomed to the target location that matches the entered address information (see the screenshot below)


Strelitzer Str. 29, 10115 Berlin, Germany

1 match(es) returned from geocoder (0.241 seconds)

3D City Database Importer/Exporter - Map window
Go
Apply
Cancel
Bounding Box
Show
Clear
Address lookup
Use popup menu for queries
Show in Google Maps
Help
Click the link in the upper right corner of the map for usage hints
Show usage hints
[52.5389551, 13.3906281]
Map data © 'OpenStreetMap' (and) contributors, CC-BY-SA

□ Hold **Alt** key and left mouse button to select a small bounding box (see the screenshot below).



3D City Database Importer/Exporter - Map window

Streltizer Str. 29, 10115 Berlin, Germany

1 match(es) returned from geocoder (0.241 seconds)

Bounding Box

	52.5376761
13.3925164	13.3945763
	52.5366776

Address lookup

Use popup menu for queries

Help

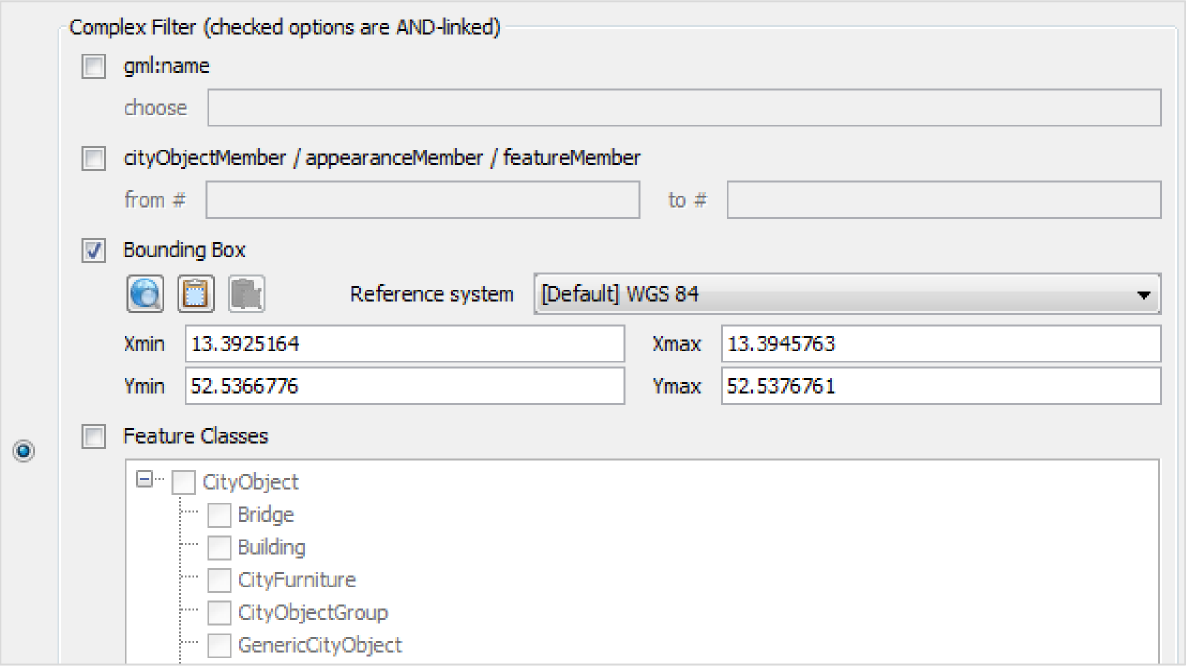
Click the link in the upper right corner of the map for usage hints

[52.5355095, 13.3902955]

Map data © 'OpenStreetMap' (and) contributors, CC-BY-SA

□ Click the ***Apply*** button on the upper right corner of the window (marked in the screenshot above)

The map window will be closed and the coordinate values of the selected bounding box will be transferred to the CityGML-Exporter window (see screenshot below).



Complex Filter (checked options are AND-linked)

□ gml:name

choose:

□ cityObjectMember / appearanceMember / featureMember

from #: to #:

☑ Bounding Box

Reference system: [Default] WGS 84

Xmin	13.3925164	Xmax	13.3945763
Ymin	52.5366776	Ymax	52.5376761

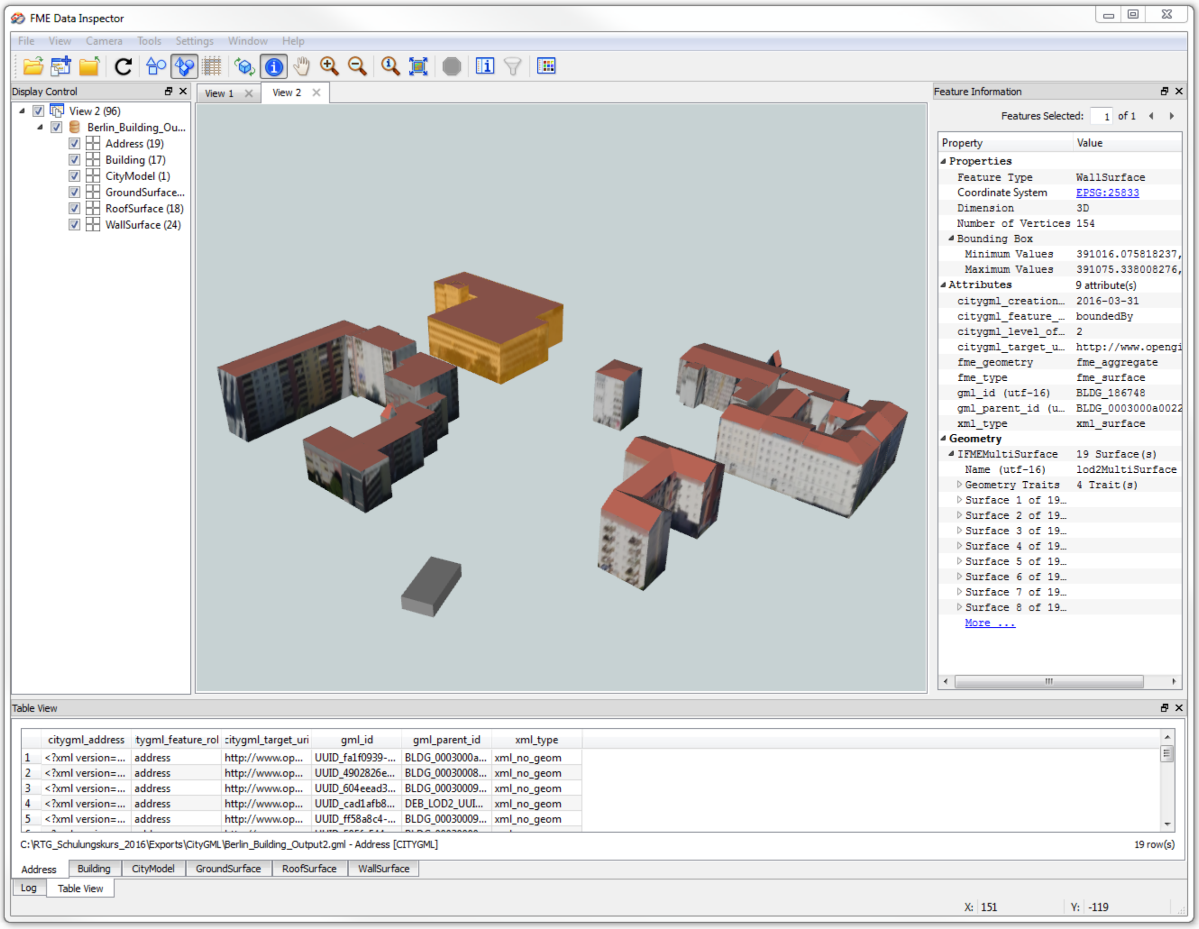
● □ Feature Classes

- □ CityObject
  - □ Bridge
  - □ Building
  - □ CityFurniture
  - □ CityObjectGroup
  - □ GenericCityObject

□ Click the ***Export*** button to start the export process.

Having completed the export process you can visually inspect the CityGML data using the CityGML Viewer FME Data Inspector.

□ Start the FME Data Inspector, and load the exported CityGML file through the menu bar ***File → Open Dataset…***



Features Selected: 1 of 1

Property	Value
Properties	
Feature Type	WallSurface
Coordinate System	EPSG:25833
Dimension	3D
Number of Vertices	154
Bounding Box	
Minimum Values	391016.075818237,
Maximum Values	391075.338008276,
Attributes	9 attribute(s)
citygml_creation…	2016-03-31
citygml_feature_…	boundedBy
citygml_level_of…	2
citygml_target_u…	http://www.opengi
fme_geometry	fme_aggregate
fme_type	fme_surface
gml_id (utf-16)	BLDG_186748
gml_parent_id (u…	BLDG_0003000a0022
xml_type	xml_surface
Geometry	
IFMEMultiSurface	19 Surface(s)
Name (utf-16)	lod2MultiSurface
Geometry Traits	4 Trait(s)
Surface 1 of 19…	
Surface 2 of 19…	
Surface 3 of 19…	
Surface 4 of 19…	
Surface 5 of 19…	
Surface 6 of 19…	
Surface 7 of 19…	
Surface 8 of 19…	
More ...	

	citygml_address	tygml_feature_rol	:itygml_target_uri	gml_id	gml_parent_id	xml_type
1	<?xml version=...	address	http://www.op...	UUID_fa1f0939-...	BLDG_0003000a...	xml_no_geom
2	<?xml version=...	address	http://www.op...	UUID_4902826e...	BLDG_00030008...	xml_no_geom
3	<?xml version=...	address	http://www.op...	UUID_604eead3...	BLDG_00030009...	xml_no_geom
4	<?xml version=...	address	http://www.op...	UUID_cad1afb8...	DEB_LOD2_UUI...	xml_no_geom
5	<?xml version=...	address	http://www.op...	UUID_ff58a8c4-...	BLDG_00030009...	xml_no_geom

C:\RTG_Schulungskurs_2016\Exports\CityGML\Berlin_Building_Output2.gml - Address [CITYGML]

19 row(s)

X: 151 Y: -119

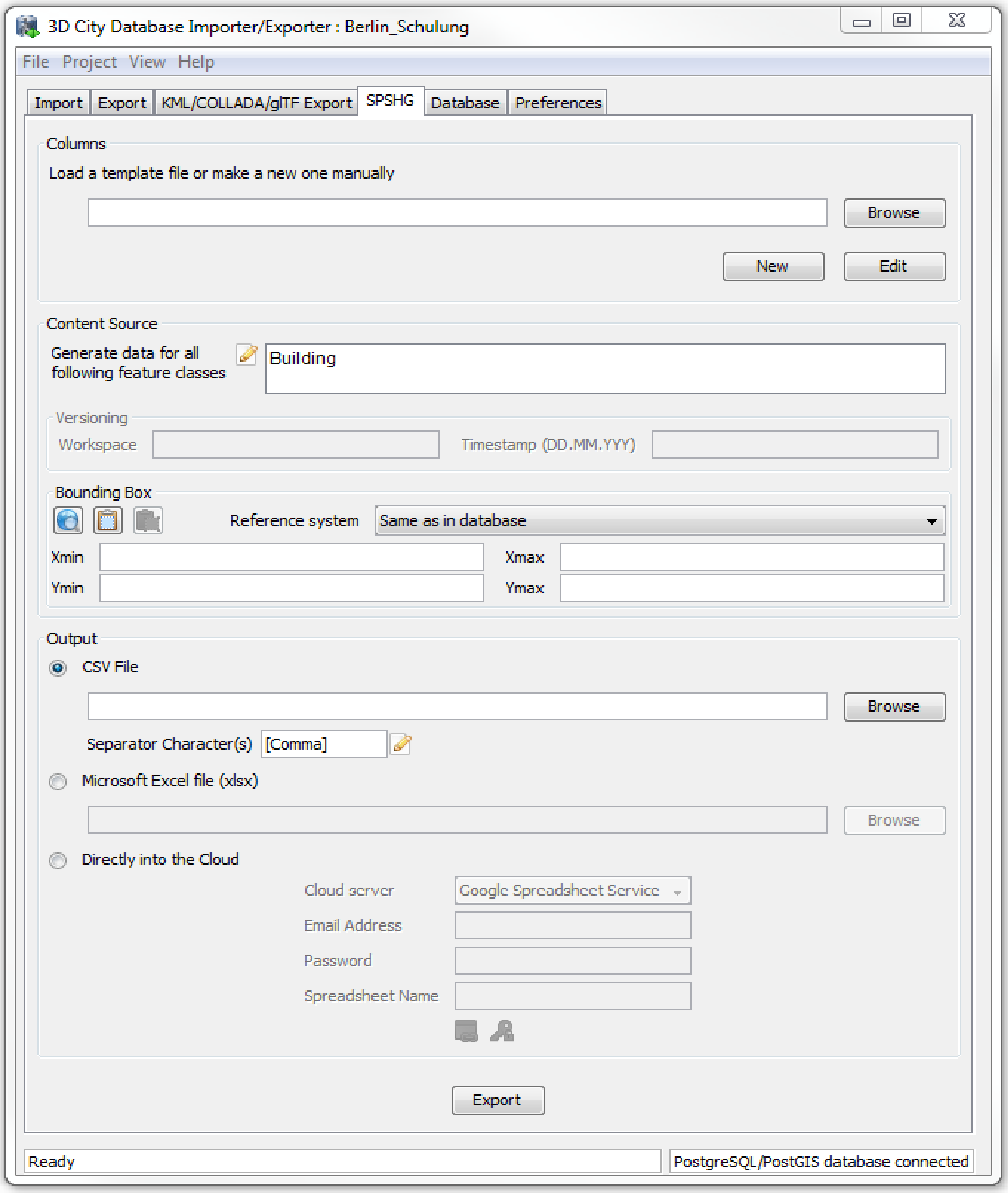
Besides the two filter functions introduced in this tutorial, The CityGML Exporter supports further filter functions to constrain the export of the 3D city model contents. For more details please refer to the 3DCityDB-Documentation.

Exercise 5: Spreadsheet Export

By using the SPSHG (Spreadsheet Generator) plugin, it is possible to export data from a 3D City Database (3DCityDB) instance into a CSV/XLSX file. The CSV or XLSX file may be imported to either a spreadsheet application (like Microsoft Excel or Open Office Calc) or to a web based online spreadsheet service (like Google Fusion Table). In addition, online spreadsheets offer features like sharing and collaborating in real-time with other users and can be easily accessed via internet.

In this exercise we will learn how to create a spreadsheet using the Spreadsheet Generator Plugin and upload the created spreadsheet to the Google Cloud Service

□ Click the ***SPSHG*** tab to open the Spreadsheet Generator window (see the figure below).



Generate data for all following feature classes: Building

Reference system: Same as in database

● CSV File

Separator Character(s): [Comma]

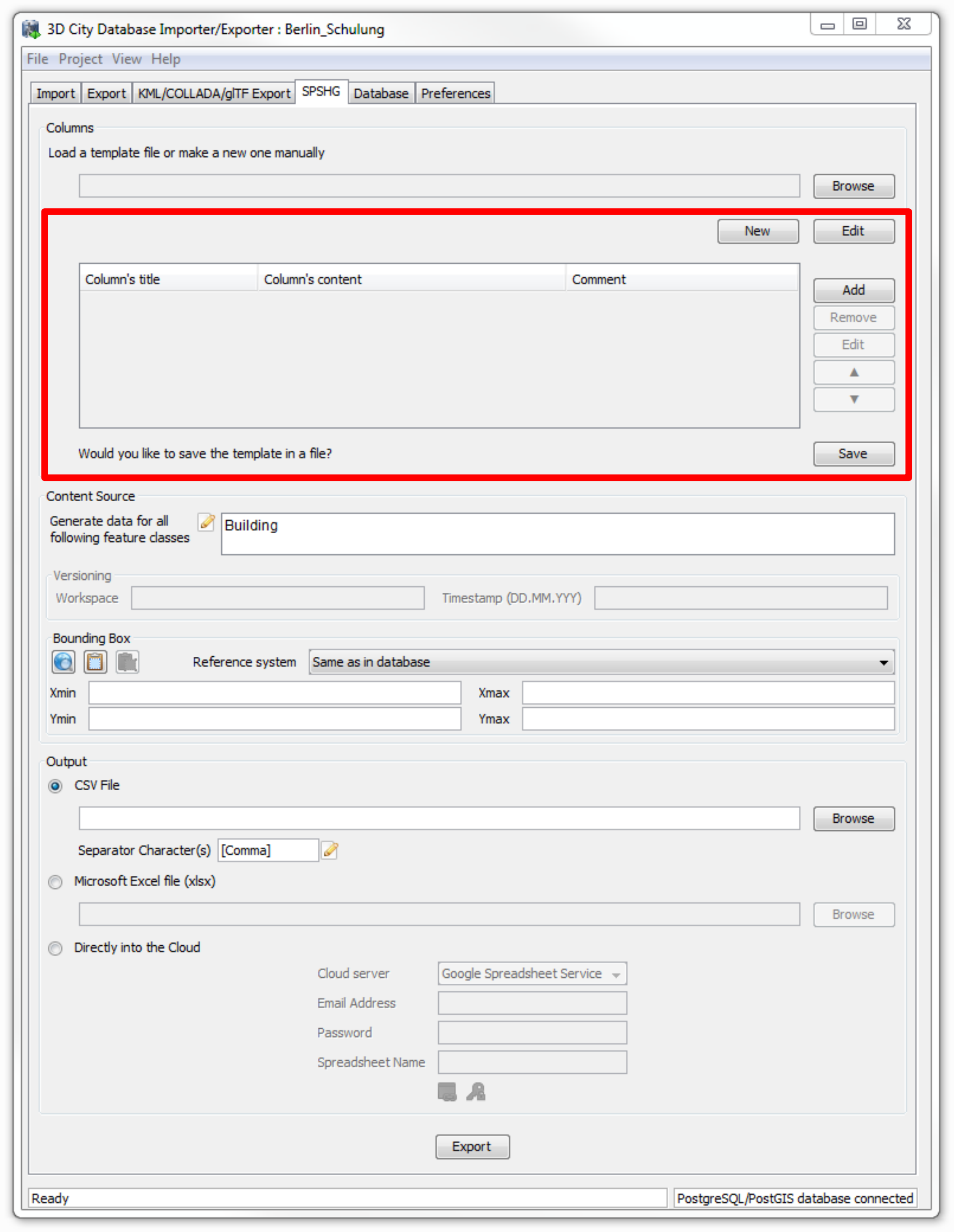
Cloud server: Google Spreadsheet Service

Ready

PostgreSQL/PostGIS database connected

First of all, the columns of your resulting spreadsheet should be defined. Now, let's create a new *template* file that defines the spreadsheet columns.

□ Click the ***New*** button to enable the template panel (marked in the figure below) for creating a template



Generate data for all following feature classes: Building

Reference system: Same as in database

● CSV File

Separator Character(s): [Comma]

○ Microsoft Excel file (xlsx)

○ Directly into the Cloud

Cloud server: Google Spreadsheet Service

Ready | PostgreSQL/PostGIS database connected

It is possible to define an arbitrary number of columns within a template file. Adding a new Column into the template is very simple:

□ Click the ***Add*** button in the template panel to open the ***New Column*** dialog (see the following screenshot)

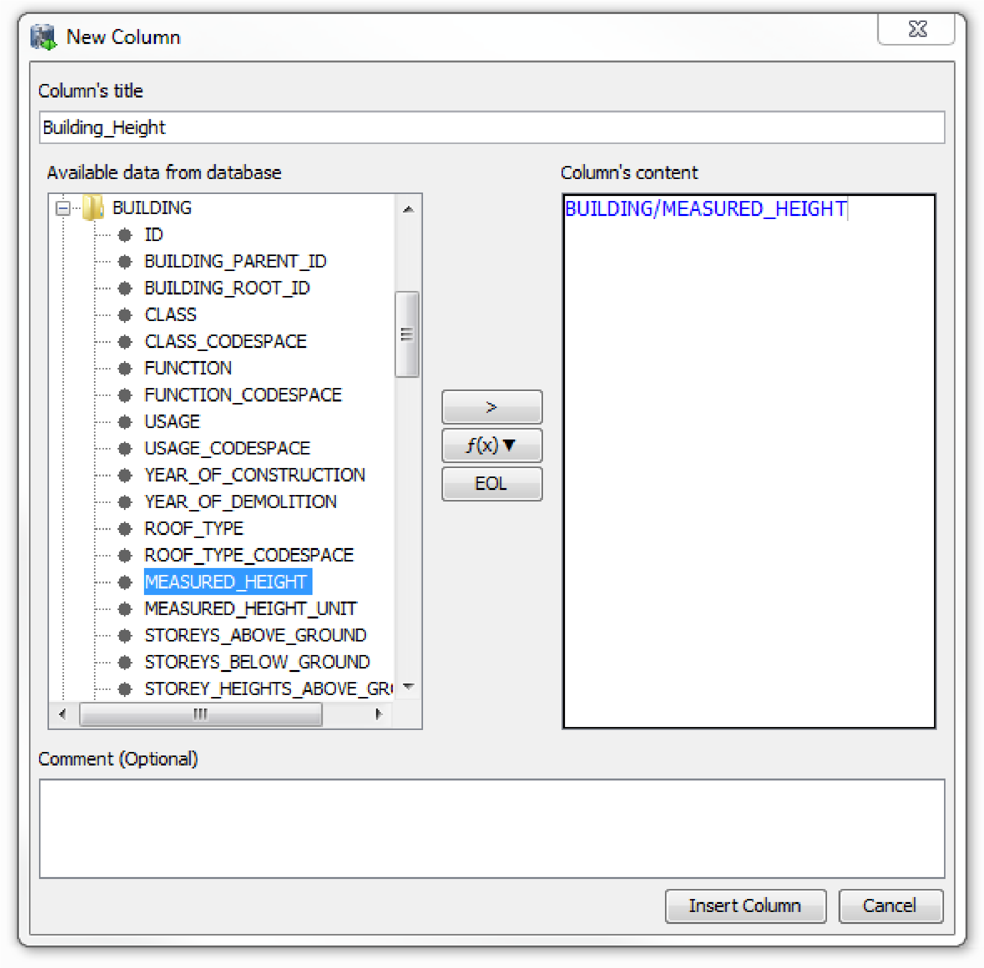
□ Fill the ***Column's title*** text field with the text value "**Building_Height**"

□ Fill the ***Column's content*** field with the text value "**BUILDING/MEASURED_HEIGHT**"

> **Note**: The expression of the ***column's content*** refers to a column (in this example is "MEASURED_HEIGHT") in a specific table (in this example is "BUILDING") of the 3D City Database.

□ Keep the ***Comment (Optional)*** field empty

The completed settings should look like the figure below



New Column

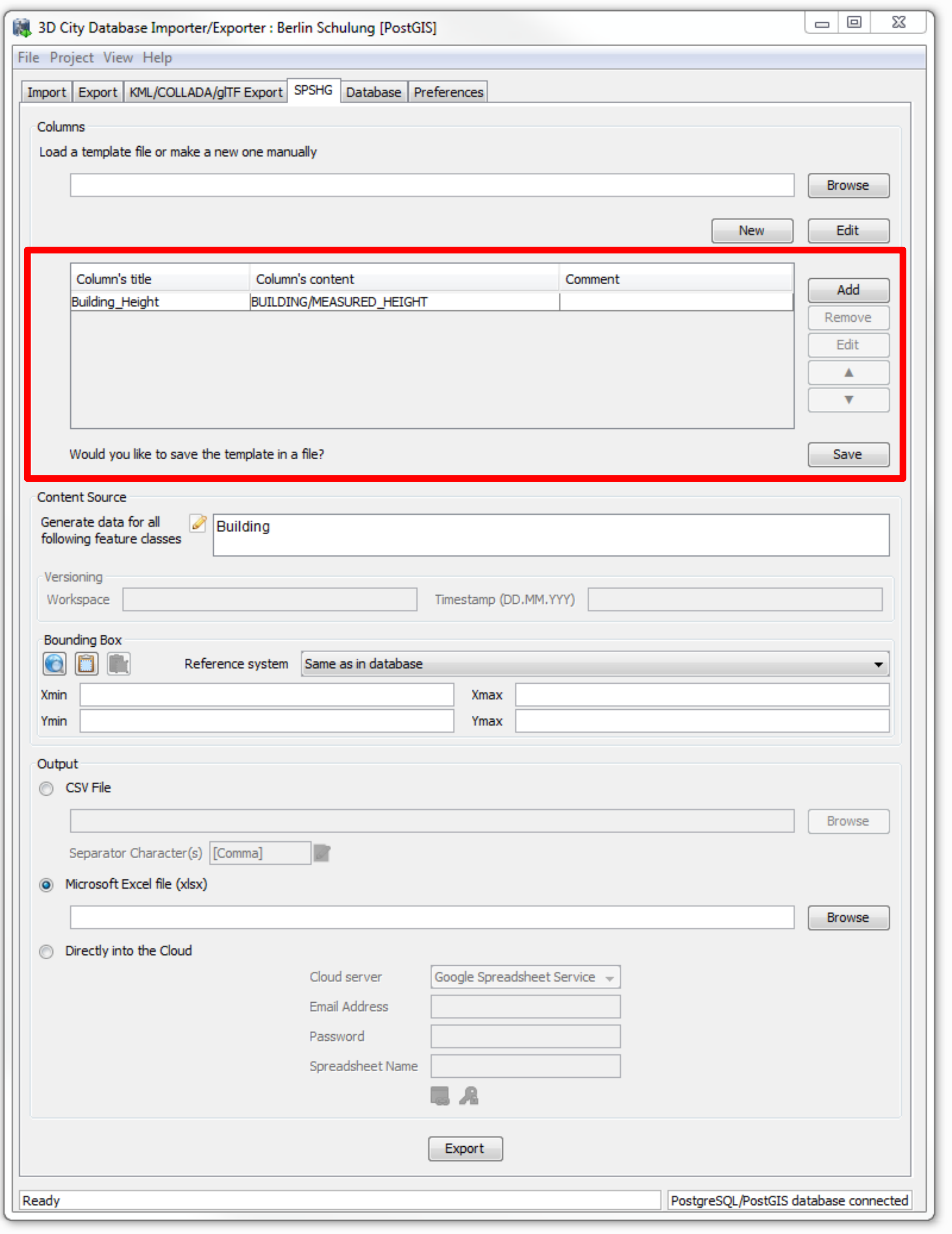
Column's title: Building_Height

Column's content: BUILDING/MEASURED_HEIGHT

Comment (Optional):

□ Click the ***Insert Column*** button.

The ***New Column*** dialog window will be closed and a new column with the title "**Building_Height**" will be created and added to the template panel (marked in the figure below).


3D City Database Importer/Exporter : Berlin Schulung [PostGIS]

Columns

Load a template file or make a new one manually

Column's title	Column's content	Comment
Building_Height	BUILDING/MEASURED_HEIGHT	

Would you like to save the template in a file?

Content Source

Generate data for all following feature classes: Building

Bounding Box

Reference system: Same as in database

Output

○ CSV File

Separator Character(s): [Comma]

● Microsoft Excel file (xlsx)

○ Directly into the Cloud

Cloud server: Google Spreadsheet Service

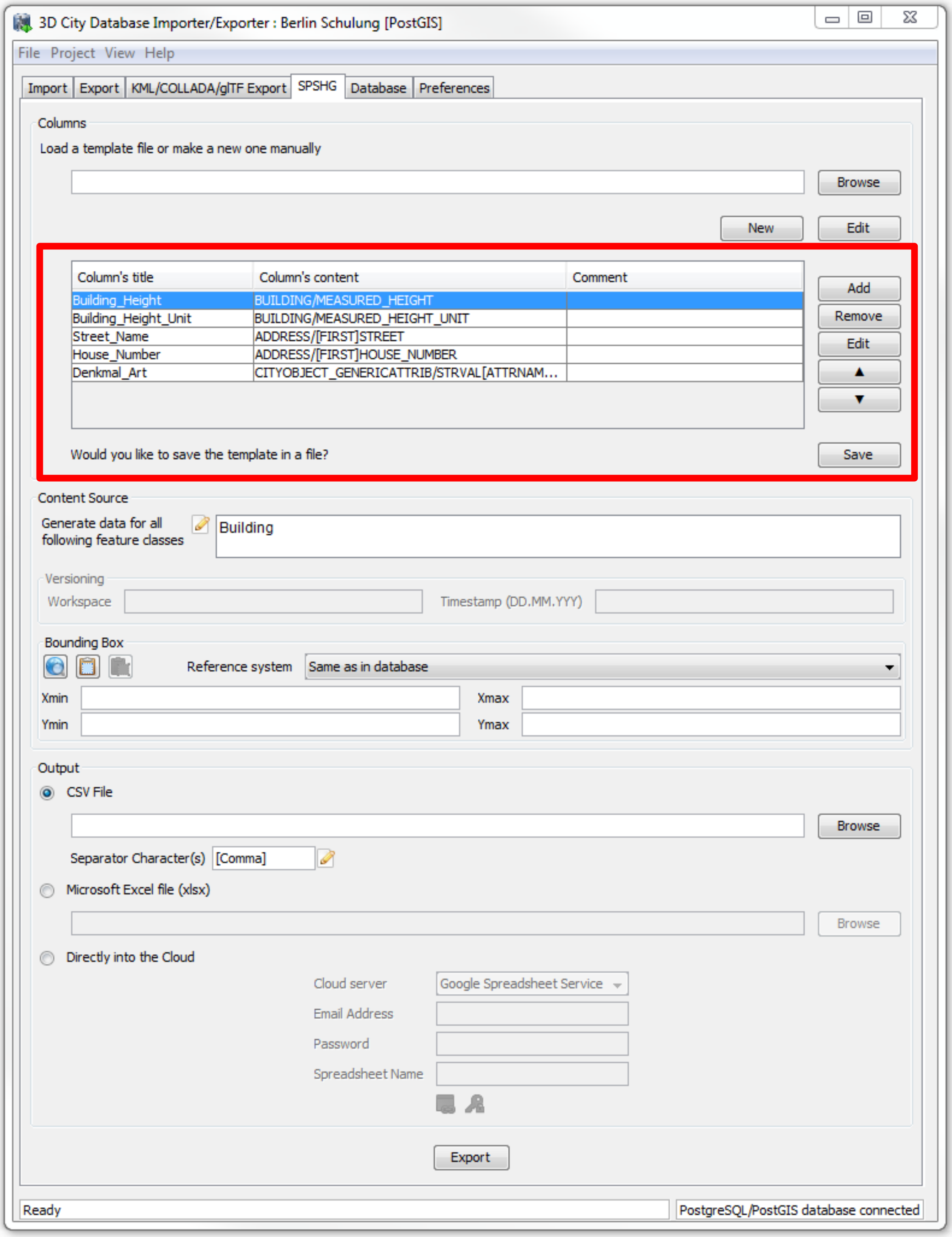
Ready

PostgreSQL/PostGIS database connected

□ Add a few additional columns (see the table below) by repeating the previous operations

Column's title	***Column's content***
Building_Height_Unit	BUILDING/MEASURED_HEIGHT_UNIT
Street_Name	ADDRESS/[FIRST]STREET
House_Number	ADDRESS/[FIRST]HOUSE_NUMBER
Denkmal_Art	CITYOBJECT_GENERICATTRIB/STRVAL[ATTRNAME='DENKMALART']

After adding the columns listed in the above table, the template panel in the Spreadsheet Generator window should look like in the following figure:



Columns

Load a template file or make a new one manually

Column's title	Column's content	Comment
Building_Height	BUILDING/MEASURED_HEIGHT	
Building_Height_Unit	BUILDING/MEASURED_HEIGHT_UNIT	
Street_Name	ADDRESS/[FIRST]STREET	
House_Number	ADDRESS/[FIRST]HOUSE_NUMBER	
Denkmal_Art	CITYOBJECT_GENERICATTRIB/STRVAL[ATTRNAM...	

Would you like to save the template in a file?

Content Source

Generate data for all following feature classes: Building

Versioning

Workspace

Timestamp (DD.MM.YYY)

Bounding Box

Reference system: Same as in database

Xmin Xmax

Ymin Ymax

Output

● CSV File

Separator Character(s): [Comma]

○ Microsoft Excel file (xlsx)

○ Directly into the Cloud

Cloud server: Google Spreadsheet Service

Email Address

Password

Spreadsheet Name

Ready

PostgreSQL/PostGIS database connected

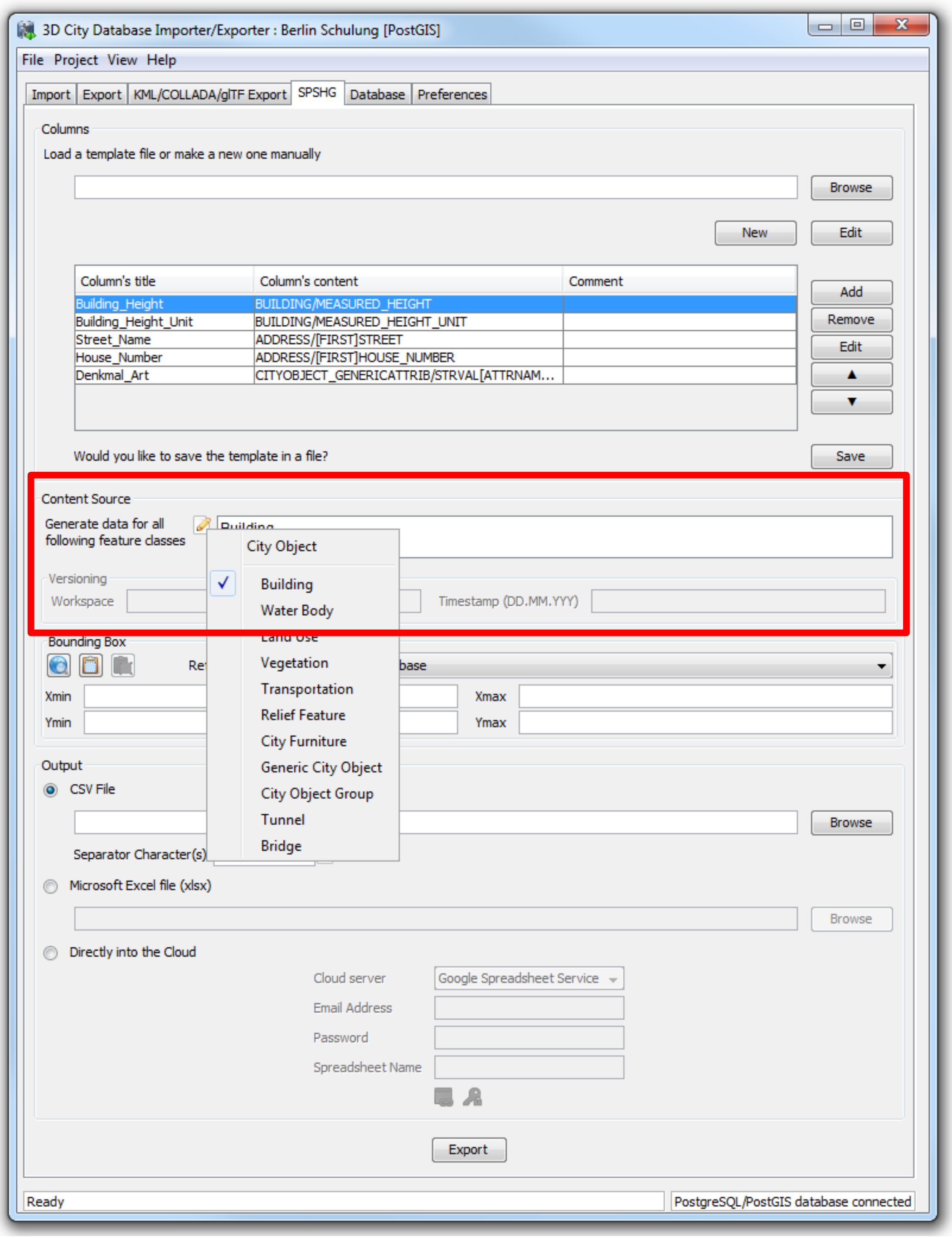
□ Click the ***Save*** button to save the template file on your computer under the path:

"***C:\RTG_Schulungskurs_2016\Exports\Spreadsheet\ Spreadsheet_Template.txt***"

> **Note**: Next time when you run the Importer/Exporter tool, you can directly load the saved template file without the need to manually create the template file again.

□ Click the ***edit*** button in the ***Content Source*** panel and select the feature class ***Building*** (see the screenshot below).

> **Note**: For CityGML datasets containing other feature types than buildings, you can specify which feature classes should be exported. You can add and remove the desired feature classes by simply clicking on the corresponding items in the feature type list.



Column's title	Column's content	Comment
Building_Height	BUILDING/MEASURED_HEIGHT	
Building_Height_Unit	BUILDING/MEASURED_HEIGHT_UNIT	
Street_Name	ADDRESS/[FIRST]STREET	
House_Number	ADDRESS/[FIRST]HOUSE_NUMBER	
Denkmal_Art	CITYOBJECT_GENERICATTRIB/STRVAL[ATTRNAM...	

In the next step we need to specify a bounding box to make sure that the concerned CityGML features lie within a given geographic area.

> **Note**: The bounding box can be interactively selected in a 2D map which can be opened by clicking on the map button 🌐 (please refer to CityGML-Export exercise).

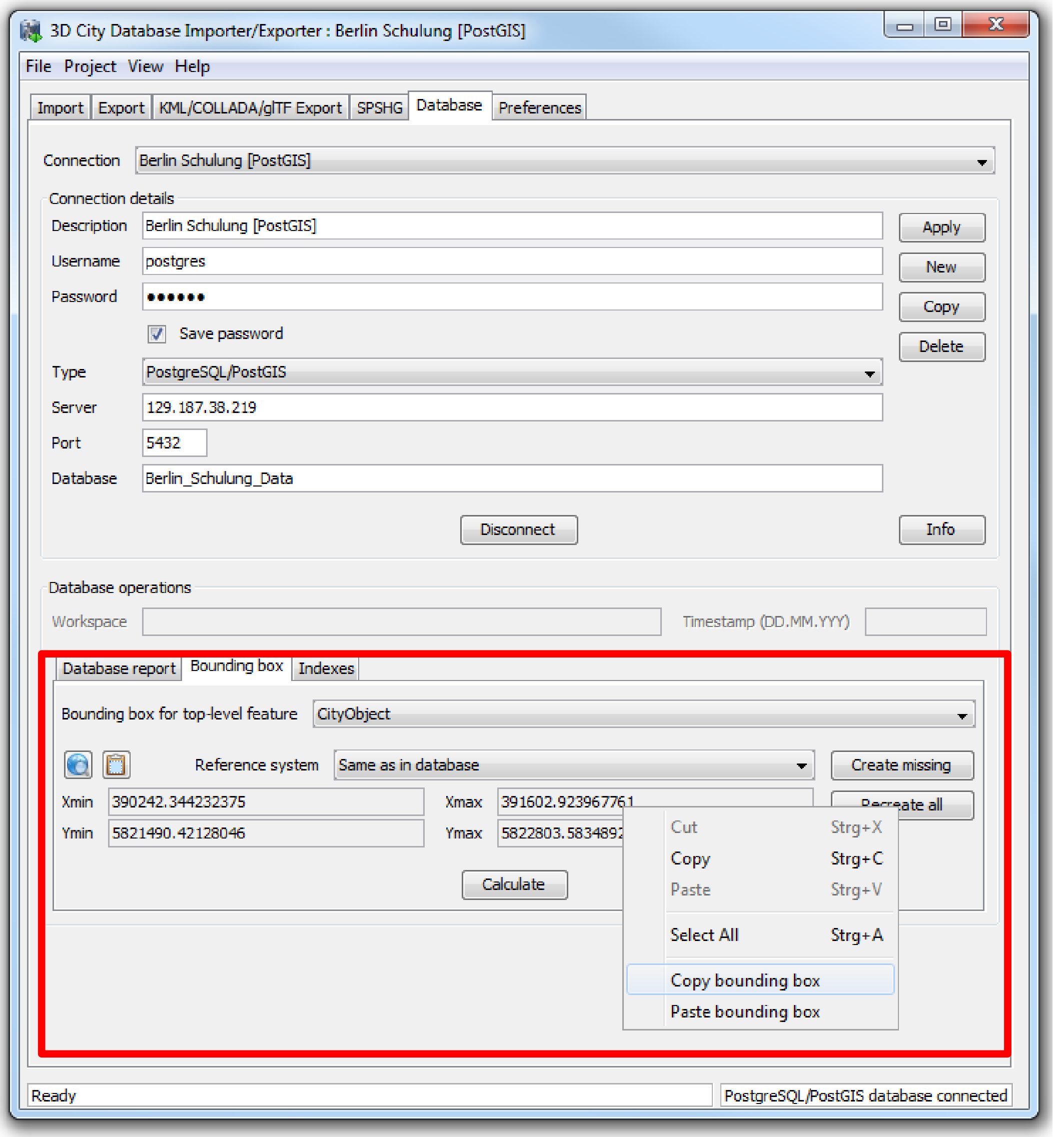
In this exercise we will export the selected content information of all CityGML features stored in the database. So, we firstly need to calculate the bounding box of the entire database contents accordingly:

□ Click on the ***Database*** tab to activate the database connection window

□ Activate the ***Bounding box*** tab, and click the ***Calculate*** button

The coordinate values of the lower left (Xmin, Ymin) and upper right (Xmax, Ymax) corner of the calculated bounding box are printed in the coordinate value fields of the dialog window (marked in the screenshot below).

□ Right-click on a coordinate value field and choose the ***Copy bounding box*** option from the context menu to copy the coordinate values to the clipboard



3D City Database Importer/Exporter : Berlin Schulung [PostGIS]

Connection: Berlin Schulung [PostGIS]

Connection details

Description: Berlin Schulung [PostGIS]

Username: postgres

Password: ••••••

☑ Save password

Type: PostgreSQL/PostGIS

Server: 129.187.38.219

Port: 5432

Database: Berlin_Schulung_Data

Database operations

Workspace:

Timestamp (DD.MM.YYY):

Bounding box for top-level feature: CityObject

Reference system: Same as in database

Xmin: 390242.344232375

Xmax: 391602.923967761

Ymin: 5821490.42128046

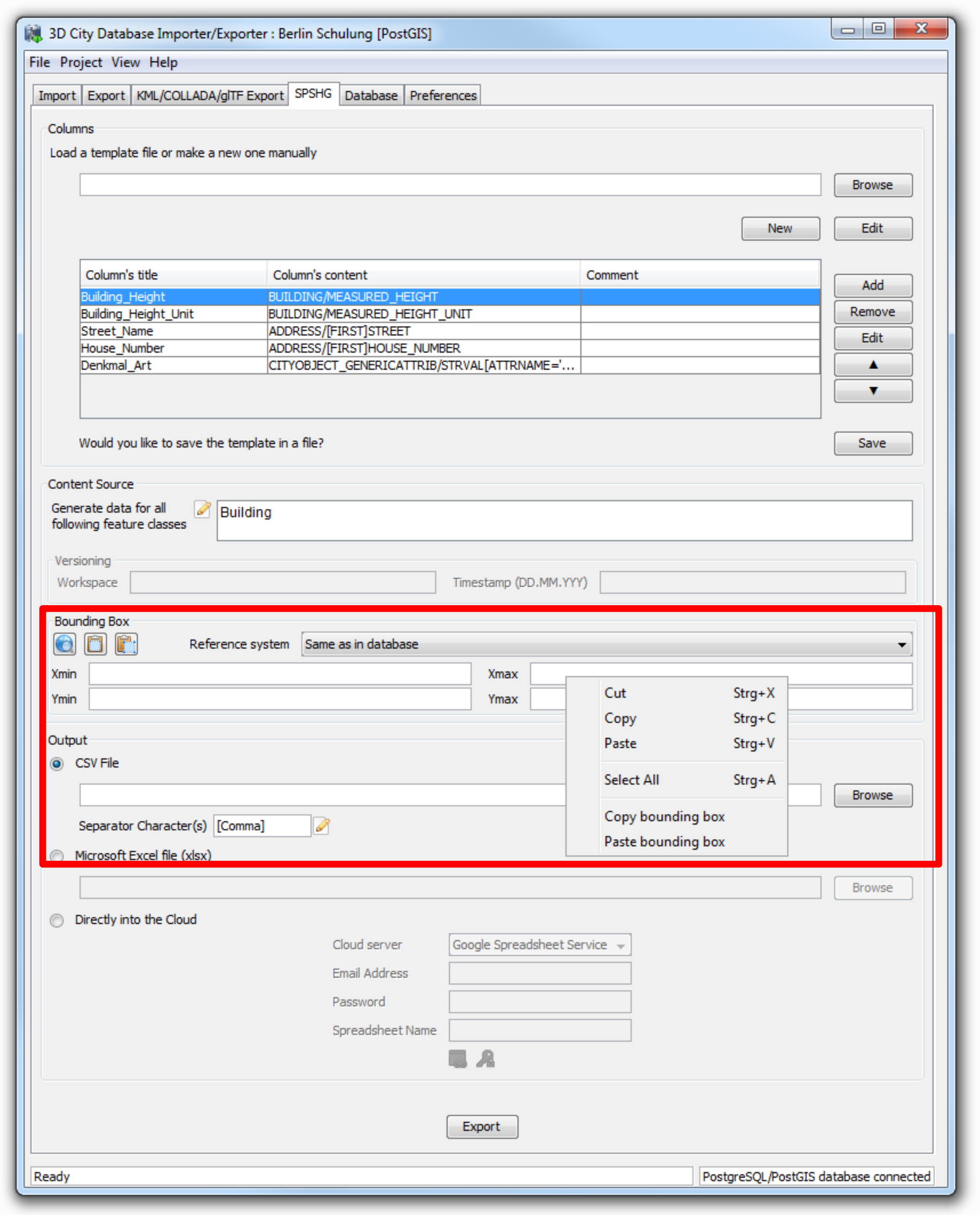
Ymax: 5822803.5834892

Ready

PostgreSQL/PostGIS database connected

□ Click the ***SPSHG*** tab to go back to the Spreadsheet Generator window

□ Right-click on a coordinate value field (marked in the figure below) and choose the ***Paste bounding box*** option from the context menu



Column's title	Column's content	Comment
Building_Height	BUILDING/MEASURED_HEIGHT	
Building_Height_Unit	BUILDING/MEASURED_HEIGHT_UNIT	
Street_Name	ADDRESS/[FIRST]STREET	
House_Number	ADDRESS/[FIRST]HOUSE_NUMBER	
Denkmal_Art	CITYOBJECT_GENERICATTRIB/STRVAL[ATTRNAME='...	

Generate data for all following feature classes: Building

Reference system: Same as in database

● CSV File

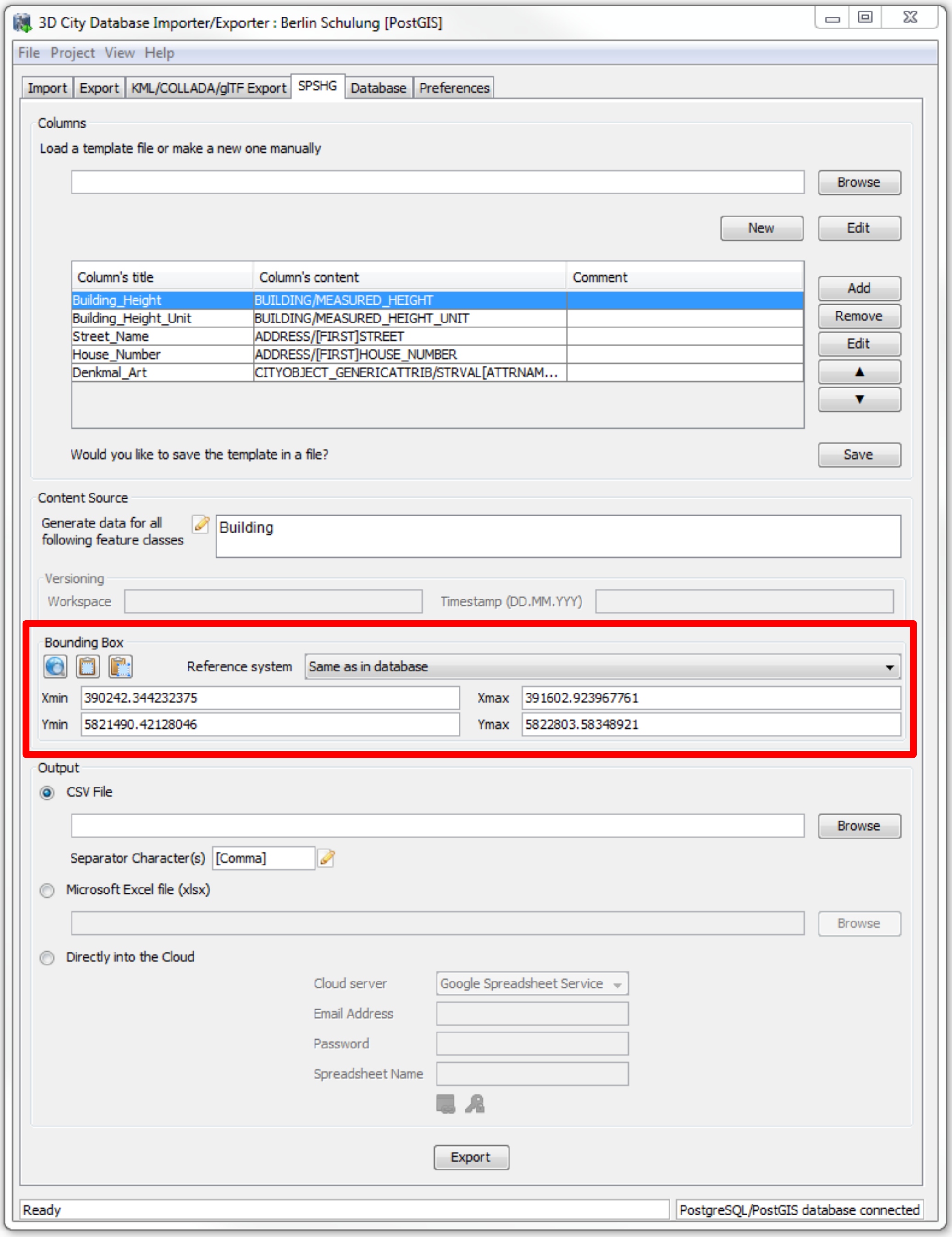
Separator Character(s): [Comma]

○ Microsoft Excel file (xlsx)

○ Directly into the Cloud

Cloud server: Google Spreadsheet Service

The calculated globe bounding box will be copied to the bounding box panel of the Spreadsheet Generator window (see the figure below).


3D City Database Importer/Exporter : Berlin Schulung [PostGIS]
File Project View Help
Import Export KML/COLLADA/glTF Export SPSHG Database Preferences
Browse
New
Edit
Add
Remove
Edit
▲
▼
Save
Browse
Browse
Export

Columns

Load a template file or make a new one manually

Column's title	Column's content	Comment
Building_Height	BUILDING/MEASURED_HEIGHT	
Building_Height_Unit	BUILDING/MEASURED_HEIGHT_UNIT	
Street_Name	ADDRESS/[FIRST]STREET	
House_Number	ADDRESS/[FIRST]HOUSE_NUMBER	
Denkmal_Art	CITYOBJECT_GENERICATTRIB/STRVAL[ATTRNAM...	

Would you like to save the template in a file?

Content Source

Generate data for all following feature classes: Building

Versioning

Workspace

Timestamp (DD.MM.YYY)

Bounding Box

Reference system: Same as in database

Xmin: 390242.344232375

Xmax: 391602.923967761

Ymin: 5821490.42128046

Ymax: 5822803.58348921

Output

● CSV File

Separator Character(s): [Comma]

○ Microsoft Excel file (xlsx)

○ Directly into the Cloud

Cloud server: Google Spreadsheet Service

Email Address

Password

Spreadsheet Name

Ready

PostgreSQL/PostGIS database connected

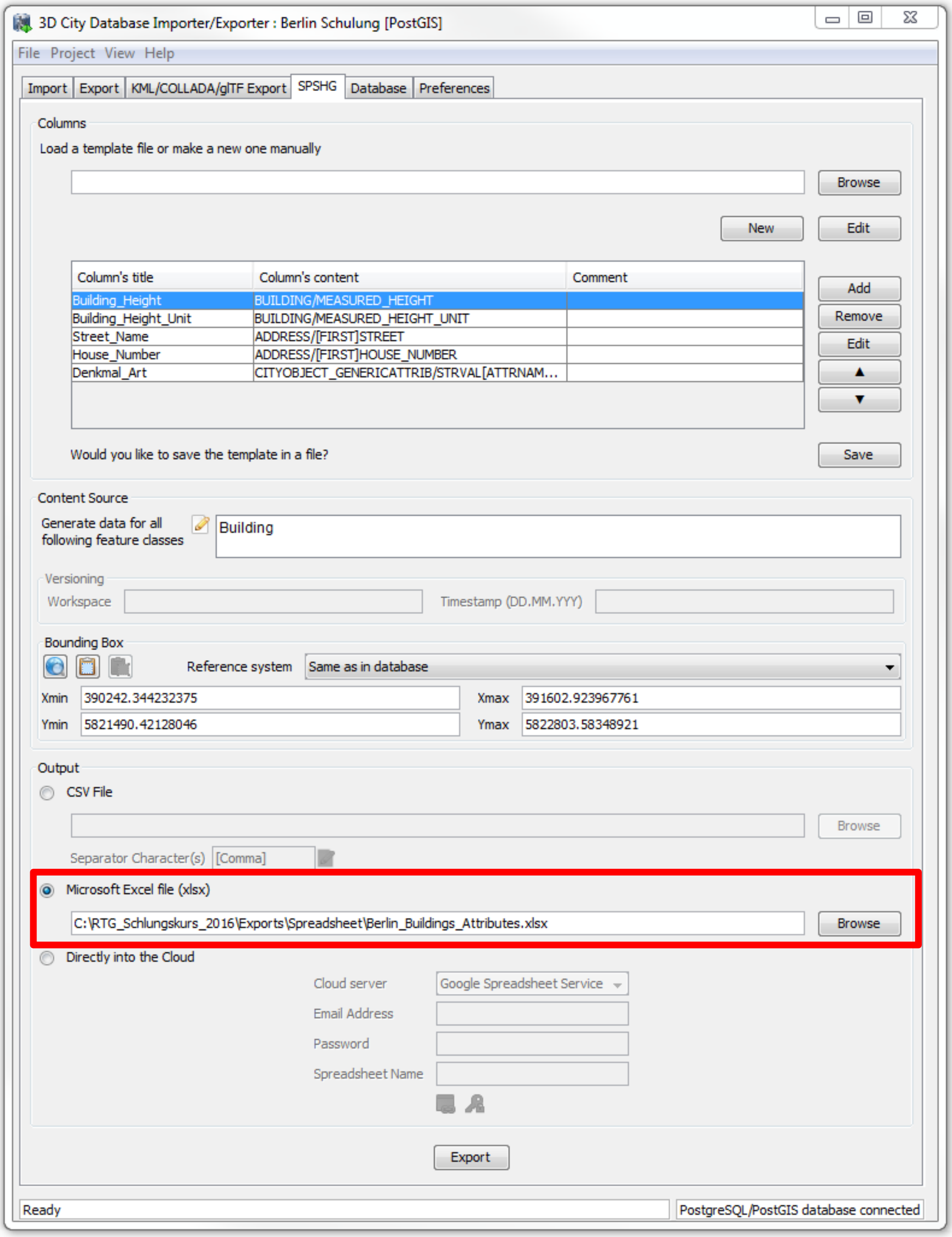
In the last step for spreadsheet export settings we have to choose the format in which the spreadsheet should be generated. In this tutorial, we choose the Microsoft Excel format.

□ Check the radio button ***Microsoft Excel file (xlsx)***.

□ Click the ***Browse*** button to open a file selection window, choose an output path and filename for the output Microsoft Excel file. We simply choose:

“C:\***RTG_Schulungskurs_2016\Exports\Spreadsheet\Berlin_Buildings_Attributes.xlsx***”

Once the above settings have been completed, the application window of the Importer/Exporter tool should look like the figure below.


3D City Database Importer/Exporter : Berlin Schulung [PostGIS]
File Project View Help
Import Export KML/COLLADA/glTF Export SPSHG Database Preferences
Columns
Load a template file or make a new one manually
Browse
New
Edit
Add
Remove
Edit
▲
▼
Would you like to save the template in a file?
Save
Content Source
Versioning
Workspace
Timestamp (DD.MM.YYY)
Bounding Box
Output
○ CSV File
Browse
○ Directly into the Cloud
Cloud server
Email Address
Password
Spreadsheet Name
Export

Column's title	Column's content	Comment
Building_Height	BUILDING/MEASURED_HEIGHT	
Building_Height_Unit	BUILDING/MEASURED_HEIGHT_UNIT	
Street_Name	ADDRESS/[FIRST]STREET	
House_Number	ADDRESS/[FIRST]HOUSE_NUMBER	
Denkmal_Art	CITYOBJECT_GENERICATTRIB/STRVAL[ATTRNAM...	

Generate data for all following feature classes: Building

Reference system: Same as in database

Xmin: 390242.344232375 Xmax: 391602.923967761

Ymin: 5821490.42128046 Ymax: 5822803.58348921

Separator Character(s): [Comma]

● Microsoft Excel file (xlsx)

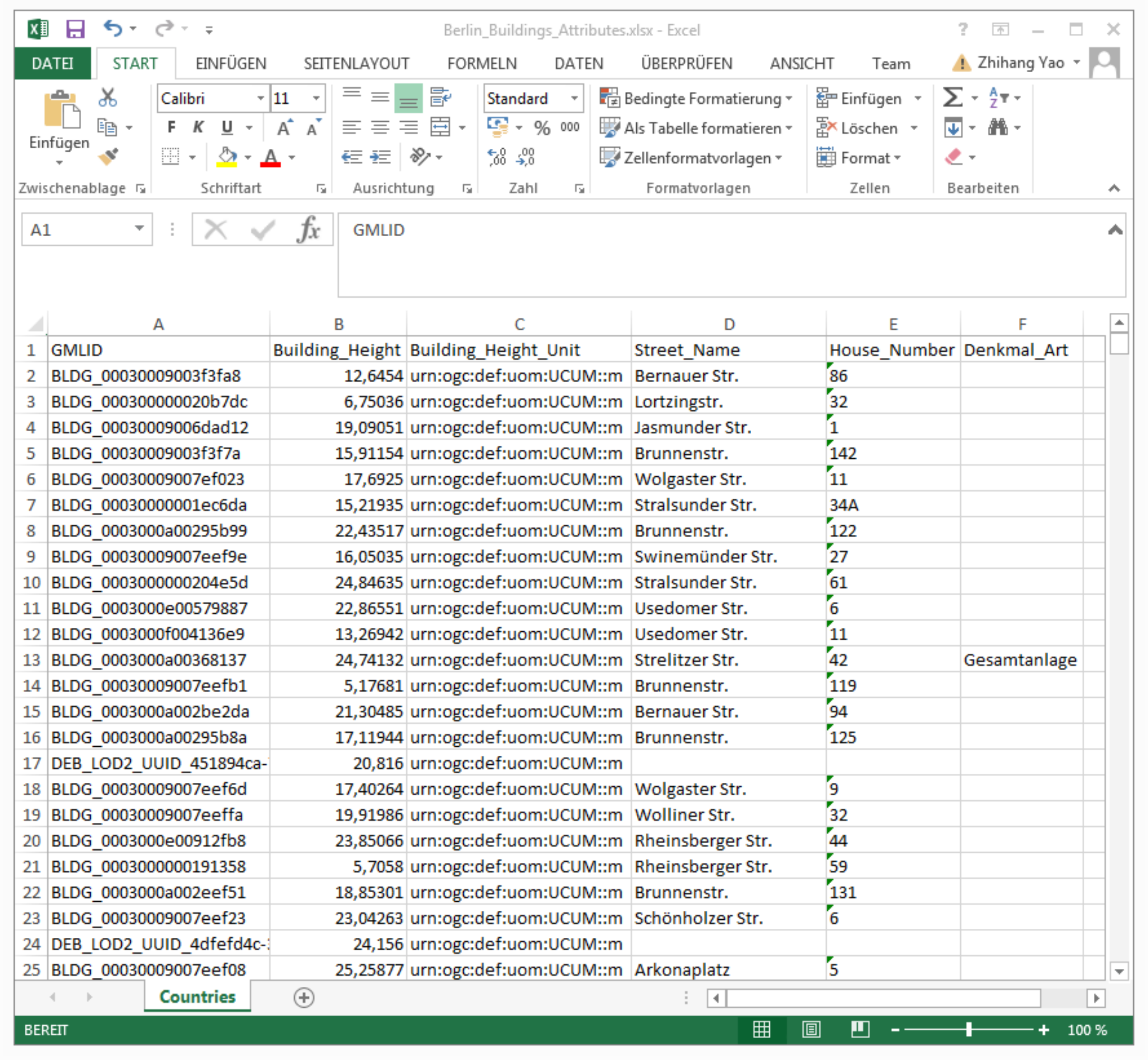
C:\RTG_Schlungskurs_2016\Exports\Spreadsheet\Berlin_Buildings_Attributes.xlsx

Cloud server: Google Spreadsheet Service

Ready | PostgreSQL/PostGIS database connected

□ Click the ***Export*** button to start the export process.

□ After having completed the export process you can open the generated Microsoft Excel file by double-clicking on it. It should look something like the followings:



	A	B	C	D	E	F
1	GMLID	Building_Height	Building_Height_Unit	Street_Name	House_Number	Denkmal_Art
2	BLDG_00030009003f3fa8	12,6454	urn:ogc:def:uom:UCUM::m	Bernauer Str.	86	
3	BLDG_000300000020b7dc	6,75036	urn:ogc:def:uom:UCUM::m	Lortzingstr.	32	
4	BLDG_00030009006dad12	19,09051	urn:ogc:def:uom:UCUM::m	Jasmunder Str.	1	
5	BLDG_00030009003f3f7a	15,91154	urn:ogc:def:uom:UCUM::m	Brunnenstr.	142	
6	BLDG_00030009007ef023	17,6925	urn:ogc:def:uom:UCUM::m	Wolgaster Str.	11	
7	BLDG_00030000001ec6da	15,21935	urn:ogc:def:uom:UCUM::m	Stralsunder Str.	34A	
8	BLDG_0003000a00295b99	22,43517	urn:ogc:def:uom:UCUM::m	Brunnenstr.	122	
9	BLDG_00030009007eef9e	16,05035	urn:ogc:def:uom:UCUM::m	Swinemünder Str.	27	
10	BLDG_000300000204e5d	24,84635	urn:ogc:def:uom:UCUM::m	Stralsunder Str.	61	
11	BLDG_0003000e00579887	22,86551	urn:ogc:def:uom:UCUM::m	Usedomer Str.	6	
12	BLDG_0003000f004136e9	13,26942	urn:ogc:def:uom:UCUM::m	Usedomer Str.	11	
13	BLDG_0003000a00368137	24,74132	urn:ogc:def:uom:UCUM::m	Strelitzer Str.	42	Gesamtanlage
14	BLDG_00030009007eefb1	5,17681	urn:ogc:def:uom:UCUM::m	Brunnenstr.	119	
15	BLDG_0003000a002be2da	21,30485	urn:ogc:def:uom:UCUM::m	Bernauer Str.	94	
16	BLDG_0003000a00295b8a	17,11944	urn:ogc:def:uom:UCUM::m	Brunnenstr.	125	
17	DEB_LOD2_UUID_451894ca-	20,816	urn:ogc:def:uom:UCUM::m			
18	BLDG_00030009007eef6d	17,40264	urn:ogc:def:uom:UCUM::m	Wolgaster Str.	9	
19	BLDG_00030009007eeffa	19,91986	urn:ogc:def:uom:UCUM::m	Wolliner Str.	32	
20	BLDG_0003000e00912fb8	23,85066	urn:ogc:def:uom:UCUM::m	Rheinsberger Str.	44	
21	BLDG_000300000191358	5,7058	urn:ogc:def:uom:UCUM::m	Rheinsberger Str.	59	
22	BLDG_0003000a002eef51	18,85301	urn:ogc:def:uom:UCUM::m	Brunnenstr.	131	
23	BLDG_00030009007eef23	23,04263	urn:ogc:def:uom:UCUM::m	Schönholzer Str.	6	
24	DEB_LOD2_UUID_4dfefd4c-	24,156	urn:ogc:def:uom:UCUM::m			
25	BLDG_00030009007eef08	25,25877	urn:ogc:def:uom:UCUM::m	Arkonaplatz	5	

The first column in the spreadsheet stores the unique identifier (GMLID) for each CityGML feature and is generated by default. More CityGML contents are contained in the other columns which we have explicitly specified in the previous steps.

In the following steps we need to upload our created spreadsheet into an online spreadsheet hosted in a cloud service. In this tutorial we use the Google Cloud Service – Goolge Fusion Table.

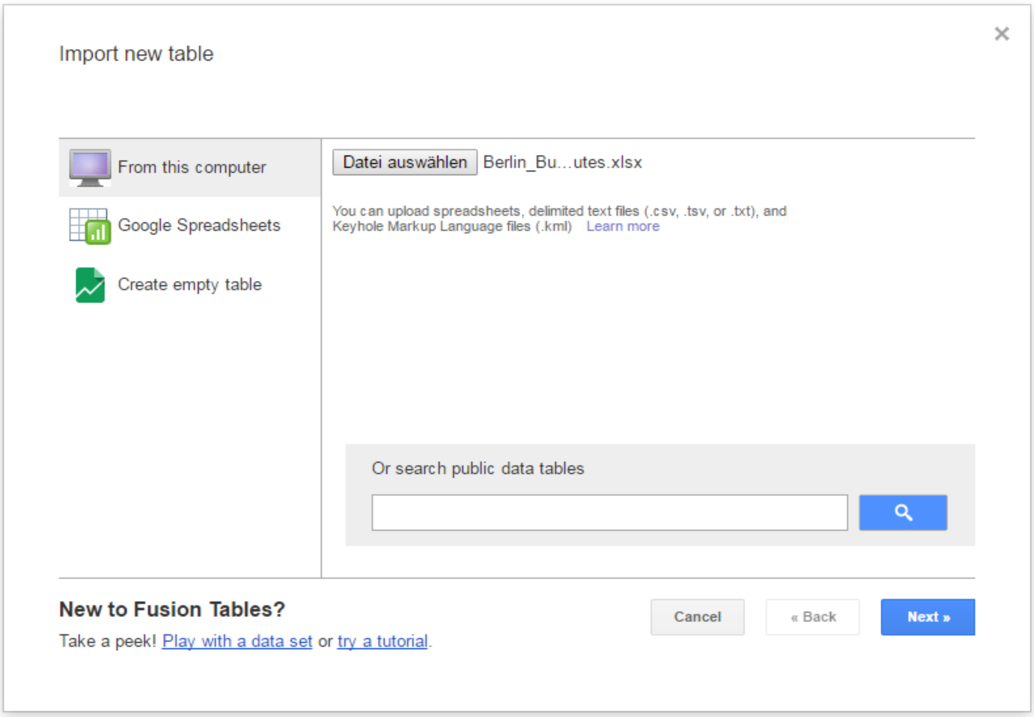
□ Open a web browser (you can use for example the Google Chrome or Mozilla Firefox**, but please never use the Microsoft Internet Explorer**) and type the following address into the address bar.

https://www.google.com/fusiontables/data?dsrcid=implicit

When you go to this page, you will be asked to log in by using your Google account.

□ Enter your Email address and the password of your Google account into the corresponding input fields

After log in, an ***Import new table*** dialog window will be displayed like in the screenshot below:



Berlin_Bu...utes.xlsx

You can upload spreadsheets, delimited text files (.csv, .tsv, or .txt), and Keyhole Markup Language files (.kml) Learn more

New to Fusion Tables?

Take a peek! Play with a data set or try a tutorial.

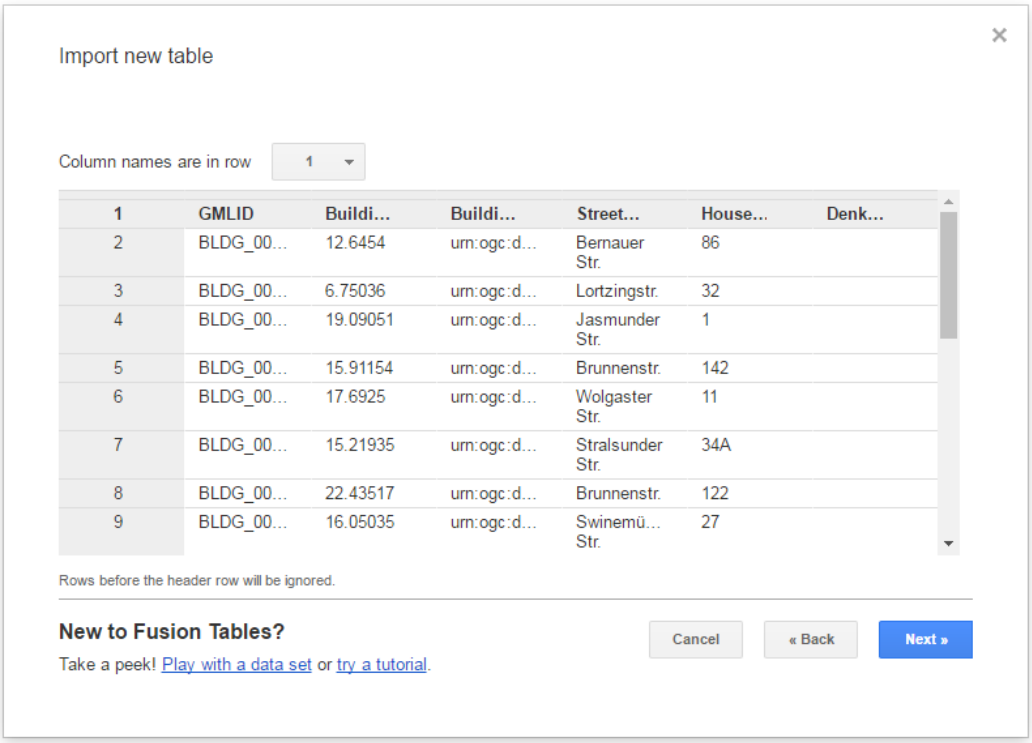
□ Click the ***Datei auswählen*** button to open a file selection window

□ Navigate to the system path of the your created Excel file and select it

□ After selecting the Excel file, click the ***Next*** button to continue

The contents of the selected table is displayed in the dialog window (see the screenshot below)

□ Briefly Check the for errors again and then click the ***Next*** button



Import new table

Column names are in row 1

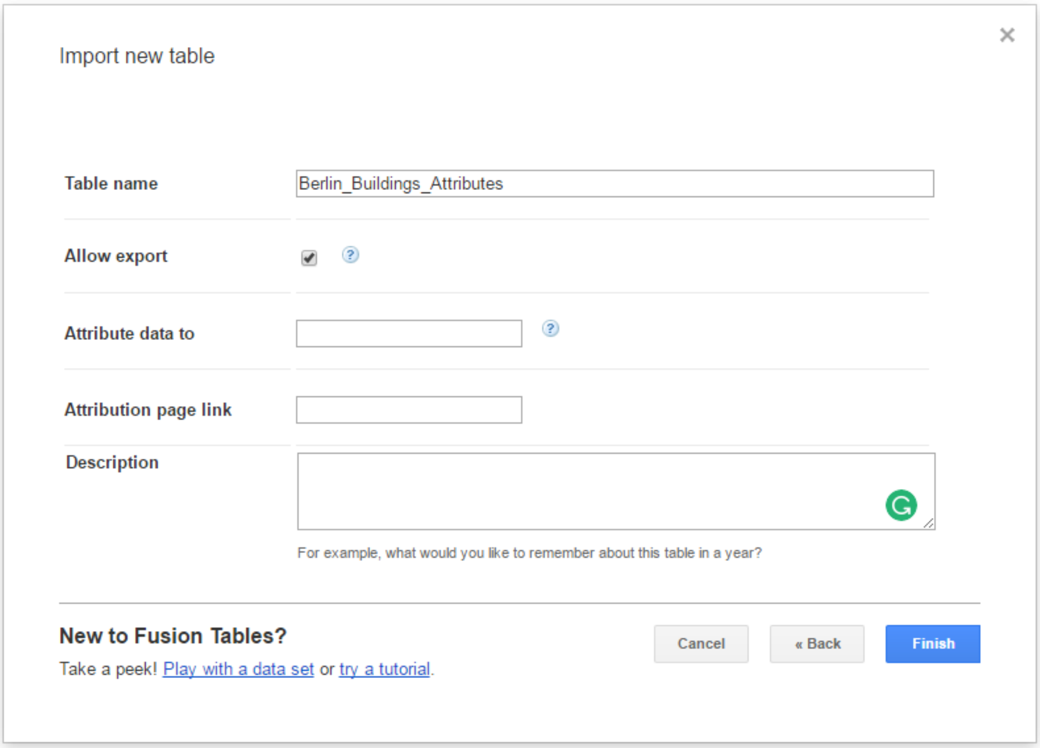
1	GMLID	Buildi...	Buildi...	Street...	House...	Denk...
2	BLDG_00...	12.6454	urn:ogc:d...	Bernauer Str.	86	
3	BLDG_00...	6.75036	urn:ogc:d...	Lortzingstr.	32	
4	BLDG_00...	19.09051	urn:ogc:d...	Jasmunder Str.	1	
5	BLDG_00...	15.91154	urn:ogc:d...	Brunnenstr.	142	
6	BLDG_00...	17.6925	urn:ogc:d...	Wolgaster Str.	11	
7	BLDG_00...	15.21935	urn:ogc:d...	Stralsunder Str.	34A	
8	BLDG_00...	22.43517	urn:ogc:d...	Brunnenstr.	122	
9	BLDG_00...	16.05035	urn:ogc:d...	Swinemü... Str.	27	

Rows before the header row will be ignored.

New to Fusion Tables?

Take a peek! Play with a data set or try a tutorial.

- In the following dialog window (see the screenshot below), keep the table name as the default value "***Berlin_Buildings_Attributes***" and click the ***Finish*** button



Import new table

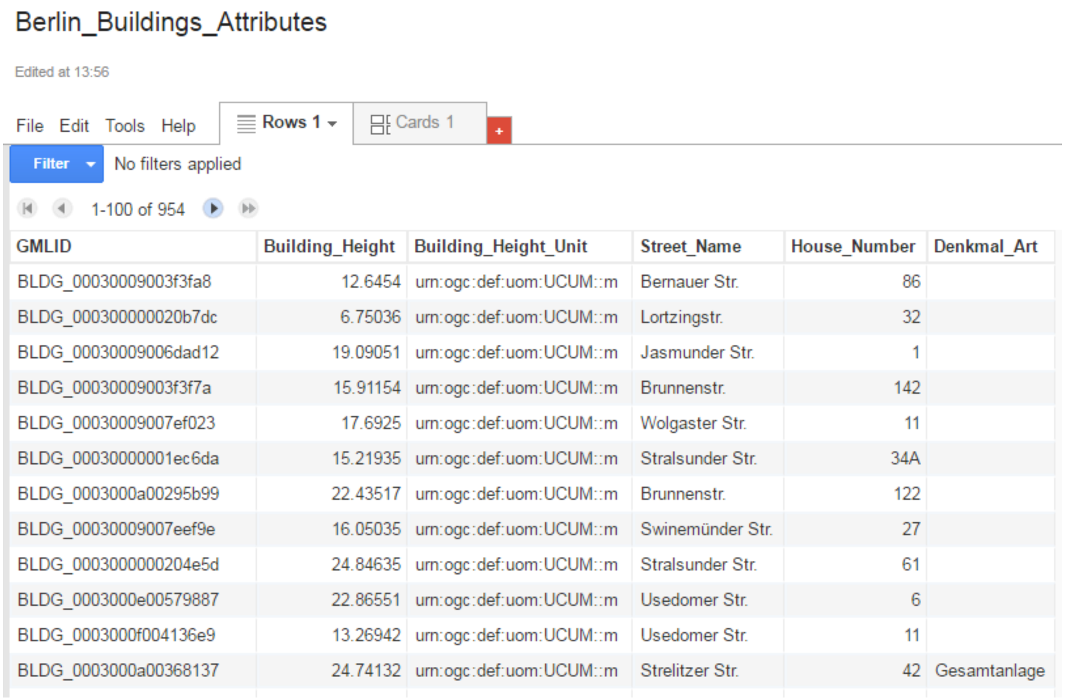
Table name	Berlin_Buildings_Attributes
Allow export	☑
Attribute data to	
Attribution page link	
Description	

For example, what would you like to remember about this table in a year?

New to Fusion Tables?

Take a peek! Play with a data set or try a tutorial.

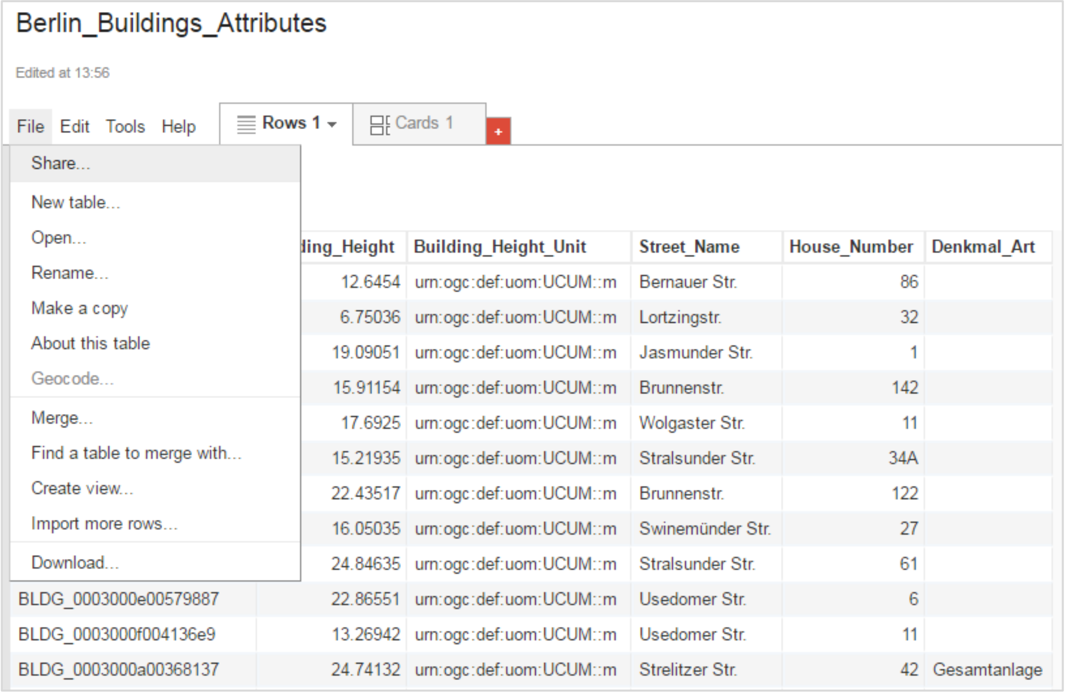
Now, your Excel file has been successfully uploaded to the Google Cloud Service and a Google Fusion Table instance has been created (see the screenshot below).



GMLID	Building_Height	Building_Height_Unit	Street_Name	House_Number	Denkmal_Art
BLDG_00030009003f3fa8	12.6454	urn:ogc:def:uom:UCUM::m	Bernauer Str.	86	
BLDG_000300000020b7dc	6.75036	urn:ogc:def:uom:UCUM::m	Lortzingstr.	32	
BLDG_00030009006dad12	19.09051	urn:ogc:def:uom:UCUM::m	Jasmunder Str.	1	
BLDG_00030009003f3f7a	15.91154	urn:ogc:def:uom:UCUM::m	Brunnenstr.	142	
BLDG_00030009007ef023	17.6925	urn:ogc:def:uom:UCUM::m	Wolgaster Str.	11	
BLDG_00030000001ec6da	15.21935	urn:ogc:def:uom:UCUM::m	Stralsunder Str.	34A	
BLDG_0003000a00295b99	22.43517	urn:ogc:def:uom:UCUM::m	Brunnenstr.	122	
BLDG_00030009007eef9e	16.05035	urn:ogc:def:uom:UCUM::m	Swinemünder Str.	27	
BLDG_0003000000204e5d	24.84635	urn:ogc:def:uom:UCUM::m	Stralsunder Str.	61	
BLDG_0003000e00579887	22.86551	urn:ogc:def:uom:UCUM::m	Usedomer Str.	6	
BLDG_0003000f004136e9	13.26942	urn:ogc:def:uom:UCUM::m	Usedomer Str.	11	
BLDG_0003000a00368137	24.74132	urn:ogc:def:uom:UCUM::m	Strelitzer Str.	42	Gesamtanlage

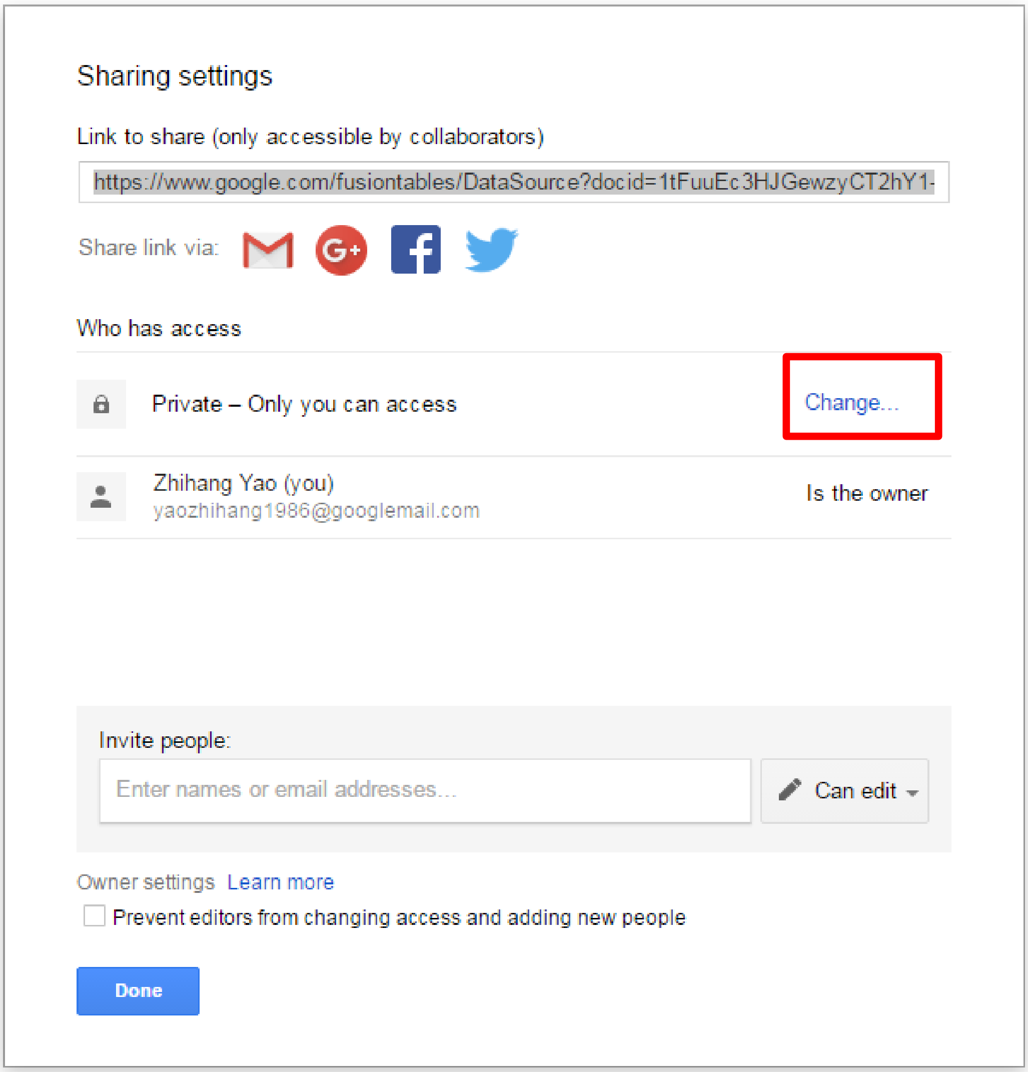
We would like to share our created online spreadsheet with other people. Here we need to change the sharing settings of the Google Fusion Table by completing the following steps:

□ Choose the **File → Share…** from the menu bar at the top of the online spreadsheet window



	ding_Height	Building_Height_Unit	Street_Name	House_Number	Denkmal_Art
	12.6454	urn:ogc:def:uom:UCUM::m	Bernauer Str.	86	
	6.75036	urn:ogc:def:uom:UCUM::m	Lortzingstr.	32	
	19.09051	urn:ogc:def:uom:UCUM::m	Jasmunder Str.	1	
	15.91154	urn:ogc:def:uom:UCUM::m	Brunnenstr.	142	
	17.6925	urn:ogc:def:uom:UCUM::m	Wolgaster Str.	11	
	15.21935	urn:ogc:def:uom:UCUM::m	Stralsunder Str.	34A	
	22.43517	urn:ogc:def:uom:UCUM::m	Brunnenstr.	122	
	16.05035	urn:ogc:def:uom:UCUM::m	Swinemünder Str.	27	
	24.84635	urn:ogc:def:uom:UCUM::m	Stralsunder Str.	61	
BLDG_0003000e00579887	22.86551	urn:ogc:def:uom:UCUM::m	Usedomer Str.	6	
BLDG_0003000f004136e9	13.26942	urn:ogc:def:uom:UCUM::m	Usedomer Str.	11	
BLDG_0003000a00368137	24.74132	urn:ogc:def:uom:UCUM::m	Strelitzer Str.	42	Gesamtanlage

In the ***Sharing settings*** window, click on ***Change…*** button (see the screenshot below)



Sharing settings

Link to share (only accessible by collaborators)

https://www.google.com/fusiontables/DataSource?docid=1tFuuEc3HJGewzyCT2hY1-

Share link via:

Who has access

Private – Only you can access

Change...

Zhihang Yao (you)
yaozhihang1986@googlemail.com

Is the owner

Invite people:

Enter names or email addresses...

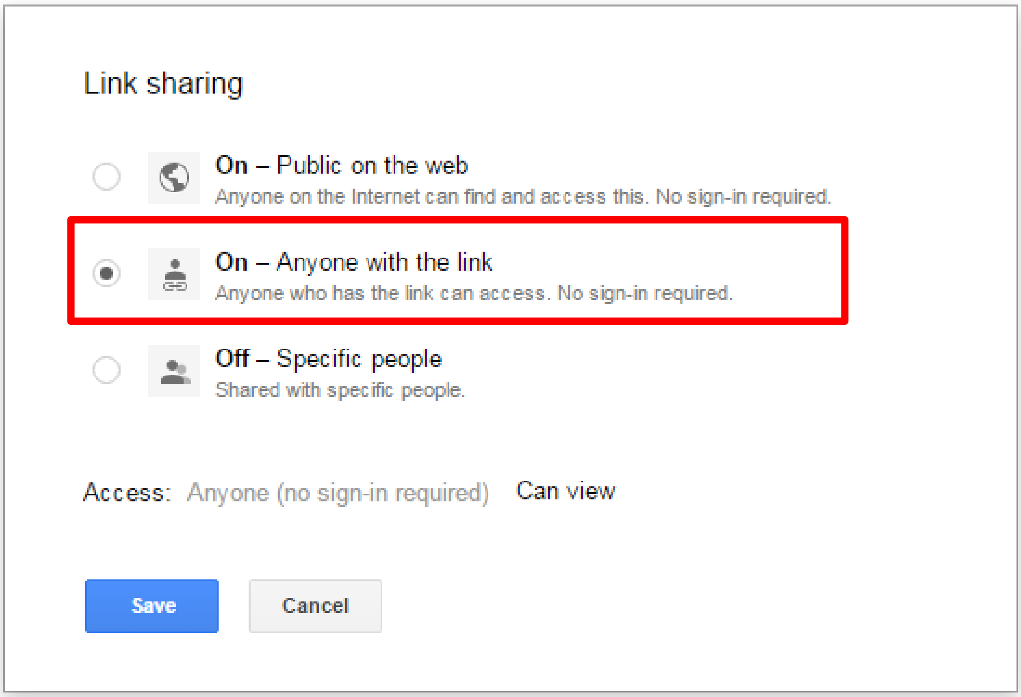
Can edit

Owner settings Learn more

□ Prevent editors from changing access and adding new people

Done

□ In the ***Link sharing*** window (see the figure below), choose the second radio button ***On – Anyone with the link***



Link sharing

○ On – Public on the web
Anyone on the Internet can find and access this. No sign-in required.

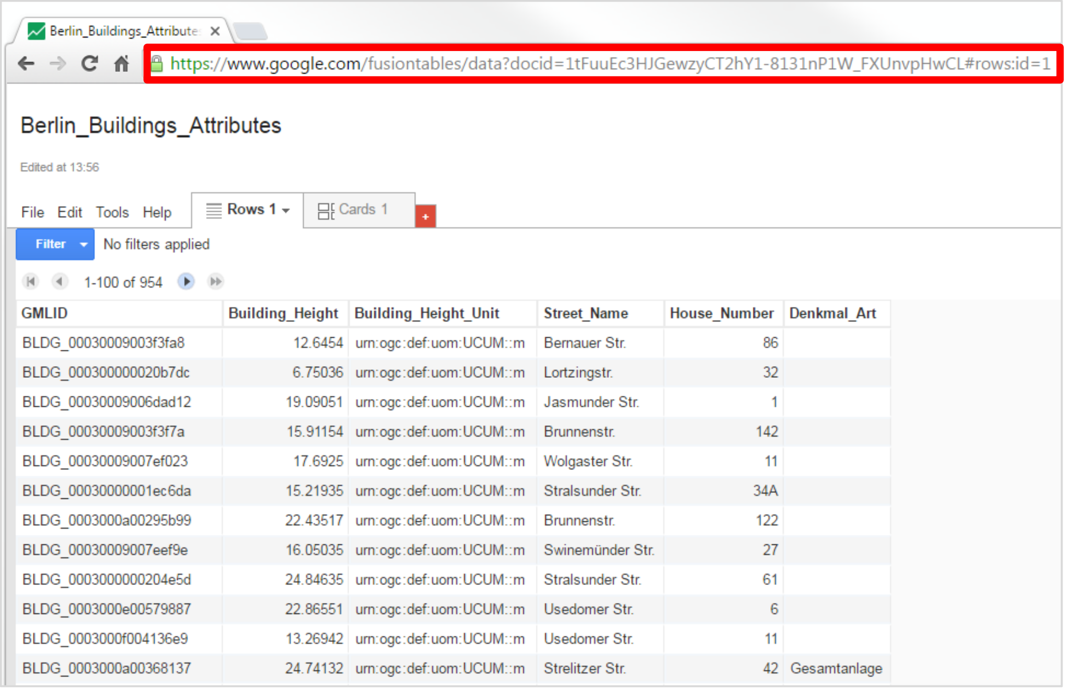
● On – Anyone with the link
Anyone who has the link can access. No sign-in required.

○ Off – Specific people
Shared with specific people.

Access: Anyone (no sign-in required) Can view

- □ Click the ***Save*** button to save the settings and close the share settings window

Now, the uploaded spreadsheet is being shared and can be accessed by anybody who has the URL of the spreadsheet. The URL of the spreadsheet can be obtained from the address bar of the web browser (marked in the screenshot below). We'll use this spreadsheet URL in the over next exercise (exercise 7). So**, please don't close this browser tab, we will need it later on.**



https://www.google.com/fusiontables/data?docid=1tFuuEc3HJGewzyCT2hY1-8131nP1W_FXUnvpHwCL#rows:id=1

Berlin_Buildings_Attributes

Edited at 13:56

No filters applied

1-100 of 954

GMLID	Building_Height	Building_Height_Unit	Street_Name	House_Number	Denkmal_Art
BLDG_00030009003f3fa8	12.6454	urn:ogc:def:uom:UCUM::m	Bernauer Str.	86	
BLDG_000300000020b7dc	6.75036	urn:ogc:def:uom:UCUM::m	Lortzingstr.	32	
BLDG_00030009006dad12	19.09051	urn:ogc:def:uom:UCUM::m	Jasmunder Str.	1	
BLDG_00030009003f3f7a	15.91154	urn:ogc:def:uom:UCUM::m	Brunnenstr.	142	
BLDG_00030009007ef023	17.6925	urn:ogc:def:uom:UCUM::m	Wolgaster Str.	11	
BLDG_00030000001ec6da	15.21935	urn:ogc:def:uom:UCUM::m	Stralsunder Str.	34A	
BLDG_0003000a00295b99	22.43517	urn:ogc:def:uom:UCUM::m	Brunnenstr.	122	
BLDG_00030009007eef9e	16.05035	urn:ogc:def:uom:UCUM::m	Swinemünder Str.	27	
BLDG_0003000000204e5d	24.84635	urn:ogc:def:uom:UCUM::m	Stralsunder Str.	61	
BLDG_0003000e00579887	22.86551	urn:ogc:def:uom:UCUM::m	Usedomer Str.	6	
BLDG_0003000f004136e9	13.26942	urn:ogc:def:uom:UCUM::m	Usedomer Str.	11	
BLDG_0003000a00368137	24.74132	urn:ogc:def:uom:UCUM::m	Strelitzer Str.	42	Gesamtanlage

Exercise 6: KML/COLLADA/glTF Export

The spatial data stored in the 3D City Database can be directly exported in KML, COLLADA, and glTF formats for presentation, viewing, and visual inspection in a broad range of applications such as virtual globes like Google Earth, ESRI ArcGIS Explorer and Cesium WebGL Virtual Globe.

In this exercise you will learn how to export the CityGML Berlin data in KML, COLLADA, and glTF formats.

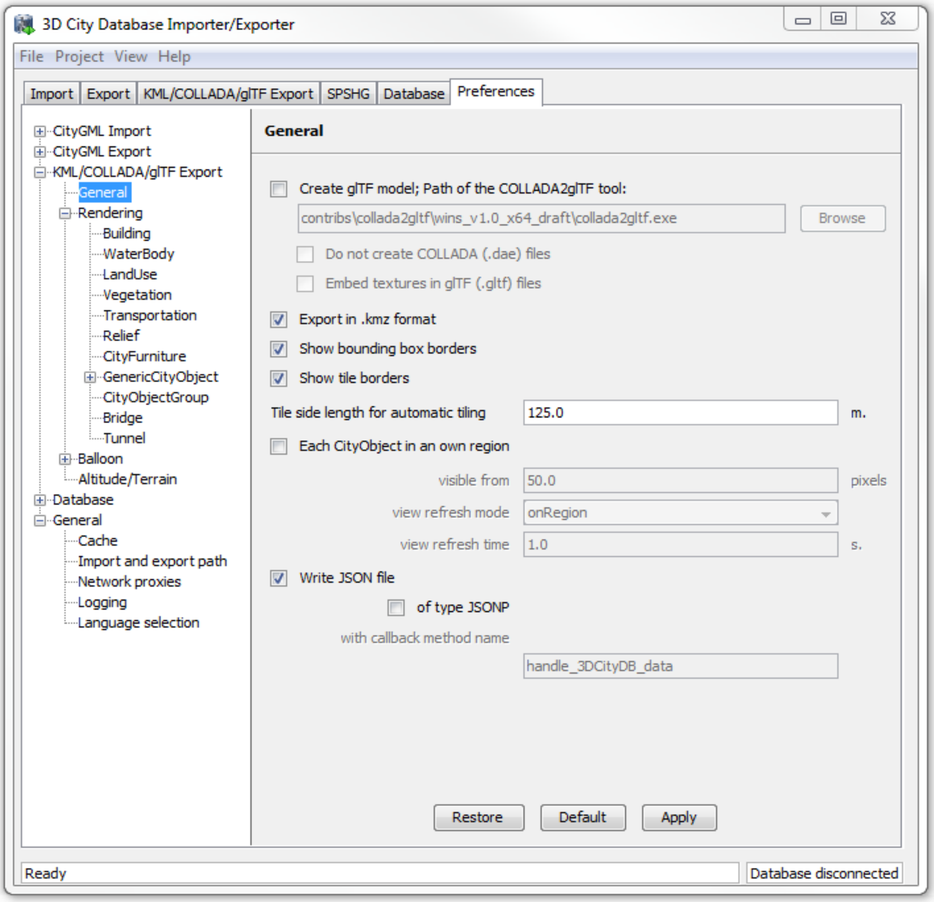
Exercise 6.1 – Export the Berlin data in KML format as extruded geometry

In this exercise we will export the Berlin CityGML data given in (LoD2) in KML format represented as generalized blocks models (LoD1) by extruding their footprint according to their height value. With its simplified geometries, the “Extruded” display form is very useful for some use cases that require high rendering performance but less geometry details.

First, we need to configure a few necessary settings for a proper KML export:

□ Click the ***Preferences*** tab in the dialog window of the Importer/Exporter tool to open the preferences settings window.

□ Expand the ***KML/COLLADA/glTF Export*** tree node and then click on the **General** tree node to open the general settings window (see the screenshot below)



General

□ Create glTF model; Path of the COLLADA2glTF tool:

contribs\collada2gltf\wins_v1.0_x64_draft\collada2gltf.exe

□ Do not create COLLADA (.dae) files

□ Embed textures in glTF (.gltf) files

☑ Export in .kmz format

☑ Show bounding box borders

☑ Show tile borders

Tile side length for automatic tiling 125.0 m.

□ Each CityObject in an own region

visible from 50.0 pixels

view refresh mode onRegion

view refresh time 1.0 s.

☑ Write JSON file

□ of type JSONP

with callback method name handle_3DCityDB_data

□ Uncheck the option ***Create glTF model; Path of the COLLADA2glTF tools***

□ Check the option ***Export in .kmz format***

□ Check the option ***Show bounding box borders***

□ Check the *option* ***Show tile borders***

□ Uncheck the option ***Each CityObject in an own region***

□ Check the option ***Write JSON file***

□ Uncheck the option ***of type JSONP***

The completed settings should look like in the figure above.

□ Click the ***Apply*** button to save the settings

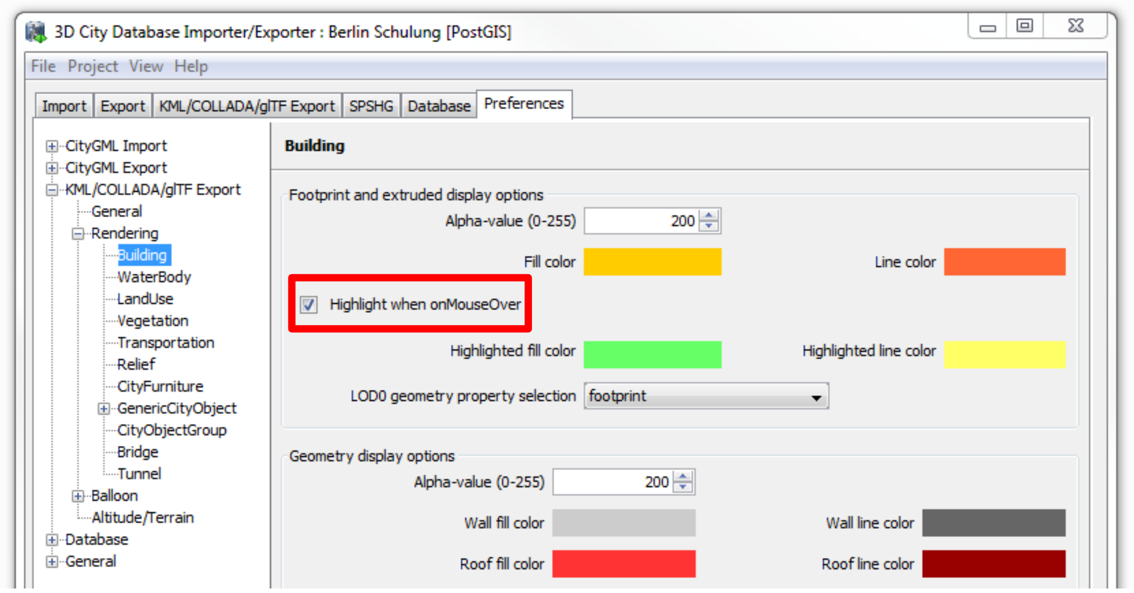
The appearance (such as surface color, border color, highlighting color etc.) of the exported KML data can also be configured:

□ Expand the ***Rendering*** tree node and click the ***Building*** tree node to open the rendering settings window for building feature type (see the screenshot below)

□ Check the option ***Highlighting when onMouseOver***

□ Keep all the other settings as default

□ Click ***Apply*** button to save the settings



Building

Footprint and extruded display options

Alpha-value (0-255) 200

Fill color 🟨 Line color 🟧

☑ Highlight when onMouseOver

Highlighted fill color 🟩 Highlighted line color 🟨

LOD0 geometry property selection footprint

Geometry display options

Alpha-value (0-255) 200

Wall fill color ⬜ Wall line color ⬛

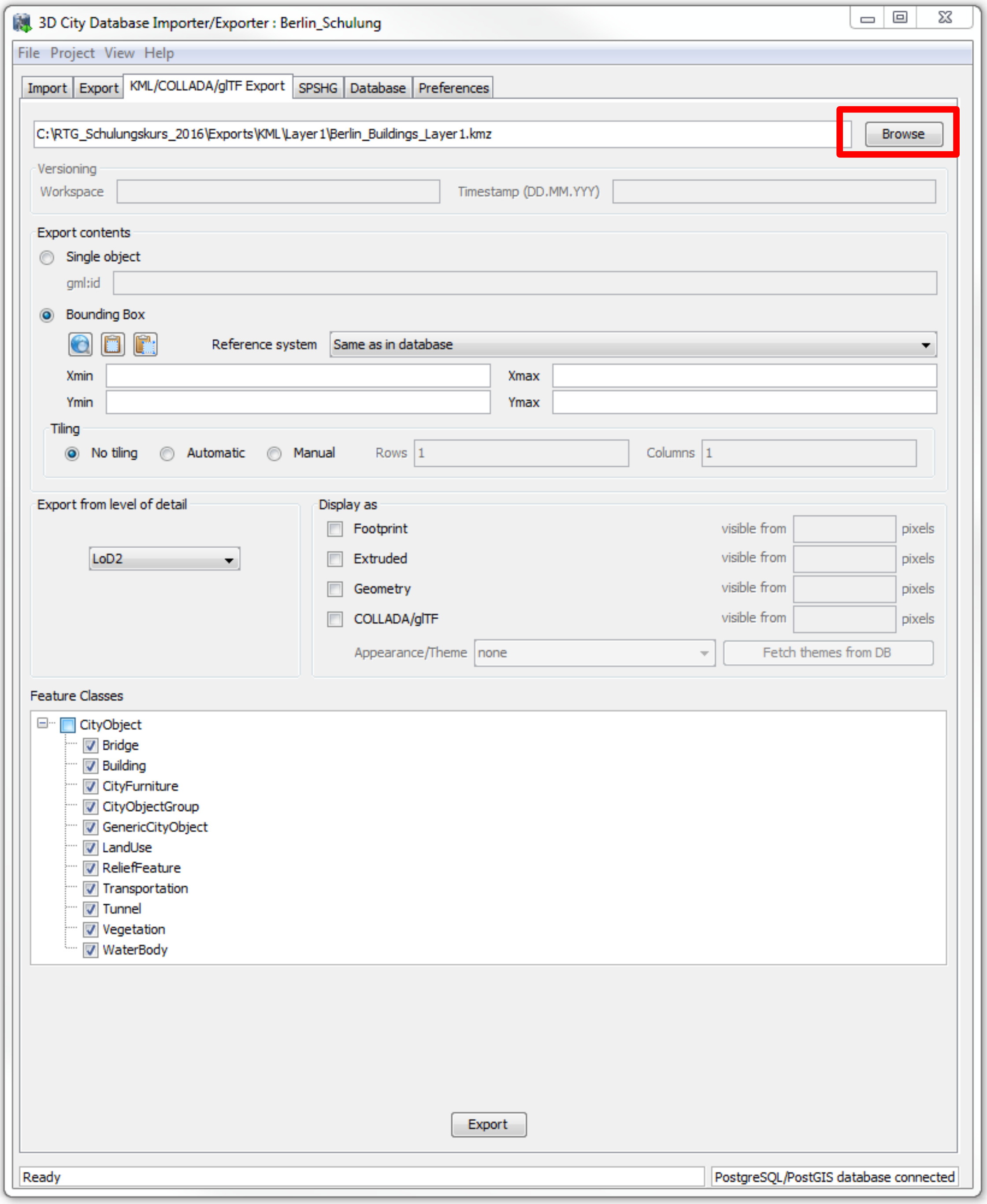
Roof fill color 🟥 Roof line color 🟥

By now, the preference settings have been done. We still need to complete some additional settings for the KML export:

□ Click the ***KML/COLLADA/glTF*** tab to open the KML/COLLADA/glTF-Exporter window (see the following figure)

□ Click the ***Browse*** button (marked in the screenshot below) to open the file selection window to choose an output path and filename for the output KML file. The path and filename should be:

“C:\***RTG_Schulungskurs_2016\Exports\KML\Layer1\Berlin_Buildings_Layer1.kml***”


3D City Database Importer/Exporter : Berlin_Schulung
File Project View Help
Import | Export | KML/COLLADA/glTF Export | SPSHG | Database | Preferences
Browse
Export

C:\RTG_Schulungskurs_2016\Exports\KML\Layer1\Berlin_Buildings_Layer1.kmz

Versioning
Workspace
Timestamp (DD.MM.YYY)

Export contents
○ Single object
gml:id
● Bounding Box
Reference system: Same as in database
Xmin
Xmax
Ymin
Ymax

Tiling
● No tiling ○ Automatic ○ Manual
Rows: 1
Columns: 1

Export from level of detail: LoD2

Display as
□ Footprint visible from pixels
□ Extruded visible from pixels
□ Geometry visible from pixels
□ COLLADA/glTF visible from pixels
Appearance/Theme: none
Fetch themes from DB

Feature Classes
□ CityObject
- ☑ Bridge
- ☑ Building
- ☑ CityFurniture
- ☑ CityObjectGroup
- ☑ GenericCityObject
- ☑ LandUse
- ☑ ReliefFeature
- ☑ Transportation
- ☑ Tunnel
- ☑ Vegetation
- ☑ WaterBody

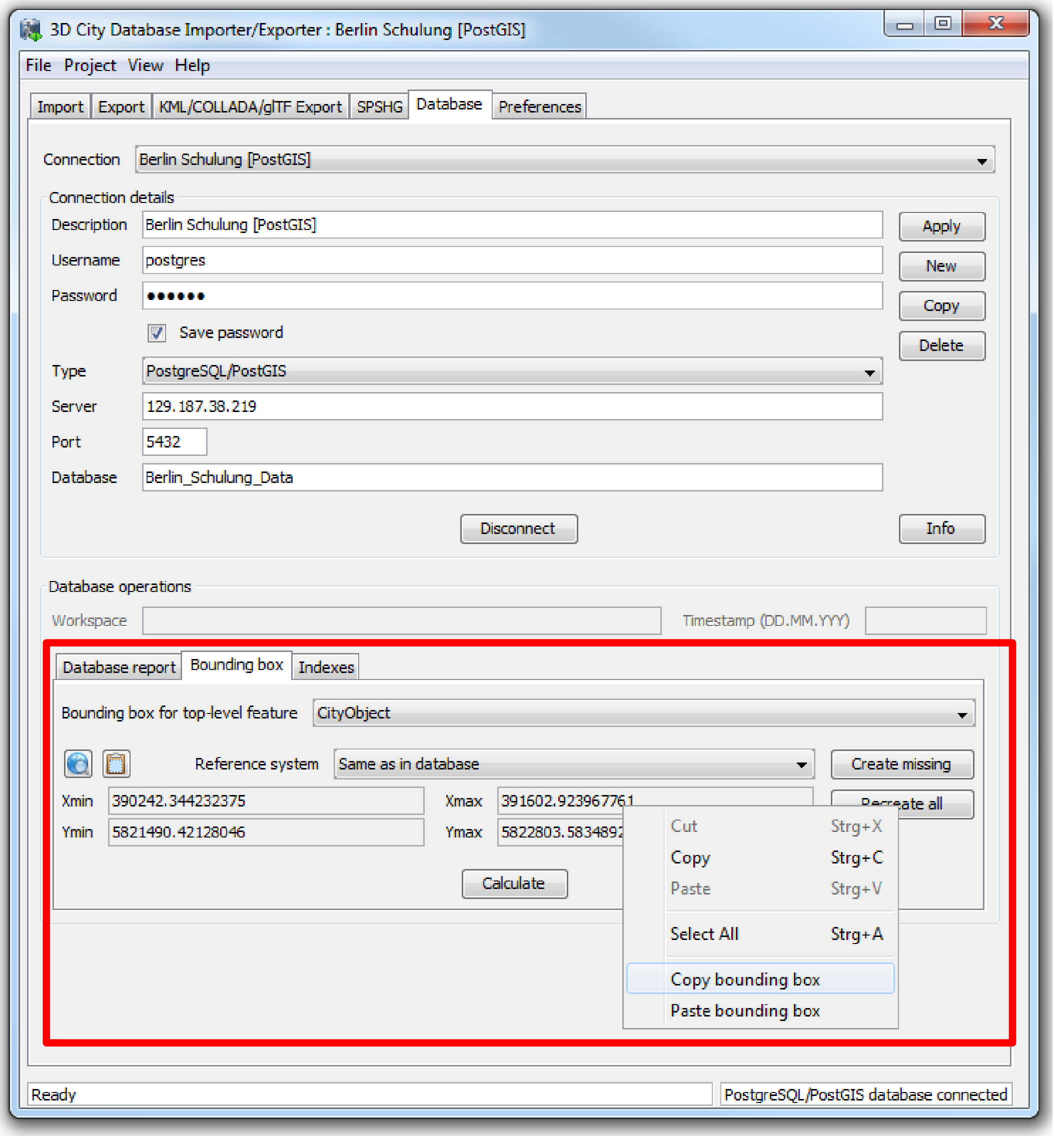
Ready
PostgreSQL/PostGIS database connected

Similar to the Spreadsheet export, we need to specify a bounding box that encloses the concerned CityGML features. Let’s repeat the steps for the calculation of the bounding box again:

□ Click on the ***Database*** tab to go back to the database connection window (see the following figure)

□ Activate the ***Bounding box*** tab, and click the ***Calculate*** button.

□ Right-click on a coordinate value field and choose the ***Copy bounding box*** option from the context menu to copy the coordinate values to the clipboard.



3D City Database Importer/Exporter : Berlin Schulung [PostGIS]

Connection: Berlin Schulung [PostGIS]

Connection details

Description: Berlin Schulung [PostGIS]

Username: postgres

Password: ••••••

☑ Save password

Type: PostgreSQL/PostGIS

Server: 129.187.38.219

Port: 5432

Database: Berlin_Schulung_Data

Database operations

Workspace:

Timestamp (DD.MM.YYY):

Bounding box for top-level feature: CityObject

Reference system: Same as in database

Xmin: 390242.344232375

Xmax: 391602.92396776̶1

Ymin: 5821490.42128046

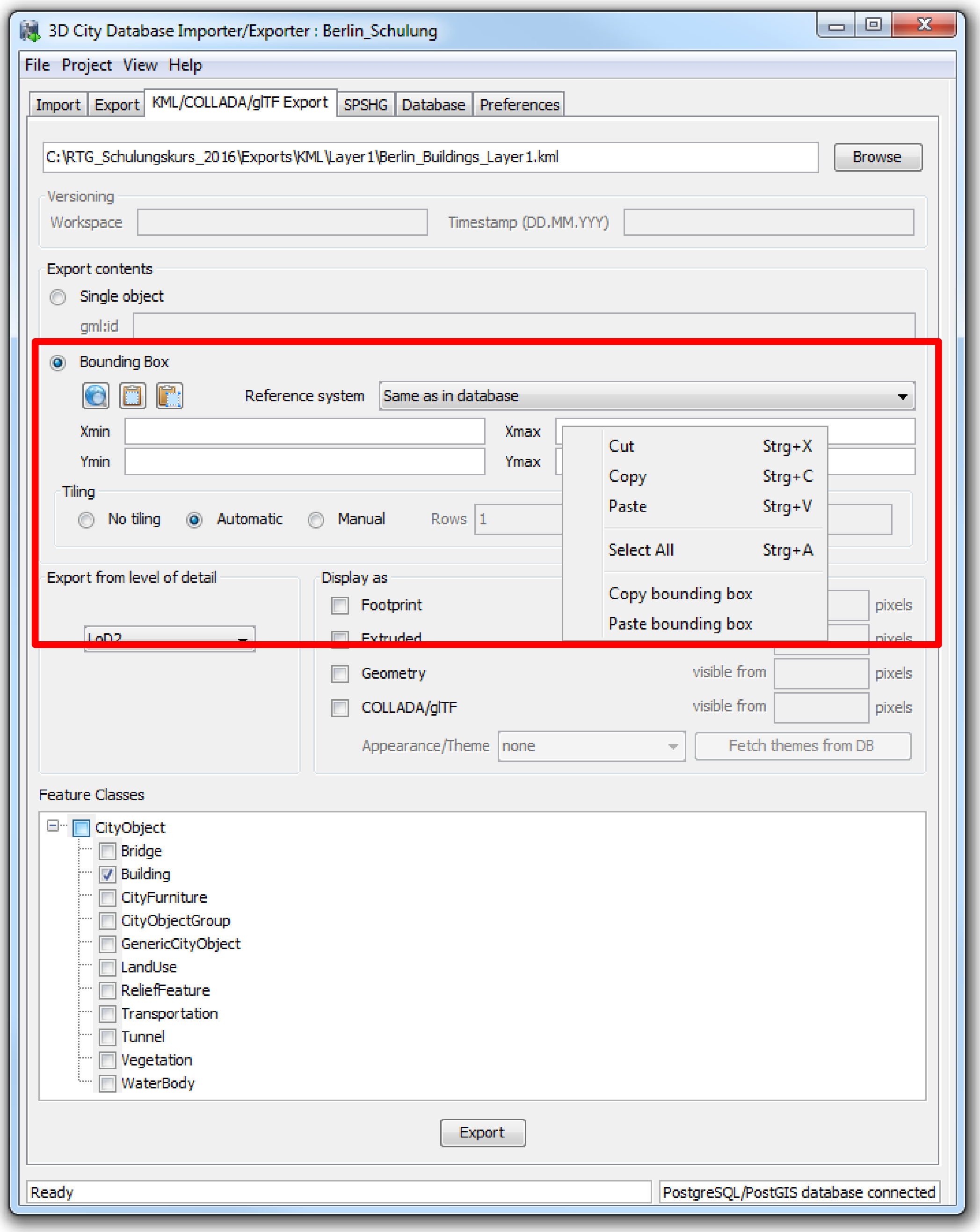
Ymax: 5822803.5834892

Ready

PostgreSQL/PostGIS database connected

□ Click the ***KML/COLLADA/glTF*** tab to go back to the KML/COLLADA/glTF-Exporter window

□ Right-click on a coordinate value field and choose the ***Paste bounding box*** option from the context menu (see the figure below)



3D City Database Importer/Exporter : Berlin_Schulung

C:\RTG_Schulungskurs_2016\Exports\KML\Layer1\Berlin_Buildings_Layer1.kml

Export contents: ● Bounding Box

Reference system: Same as in database

Tiling: ● Automatic, Rows 1

Export from level of detail: LoD2

Appearance/Theme: none

Feature Classes: ☑ Building

Ready | PostgreSQL/PostGIS database connected

By now, the calculated bounding box has been copied to the bounding box panel of the KML/COLLADA/glTF-Exporter window.

Let's continue to complete the remains settings for the KML export (marked in the following figure)

□ Check the ***Automatic*** radio button

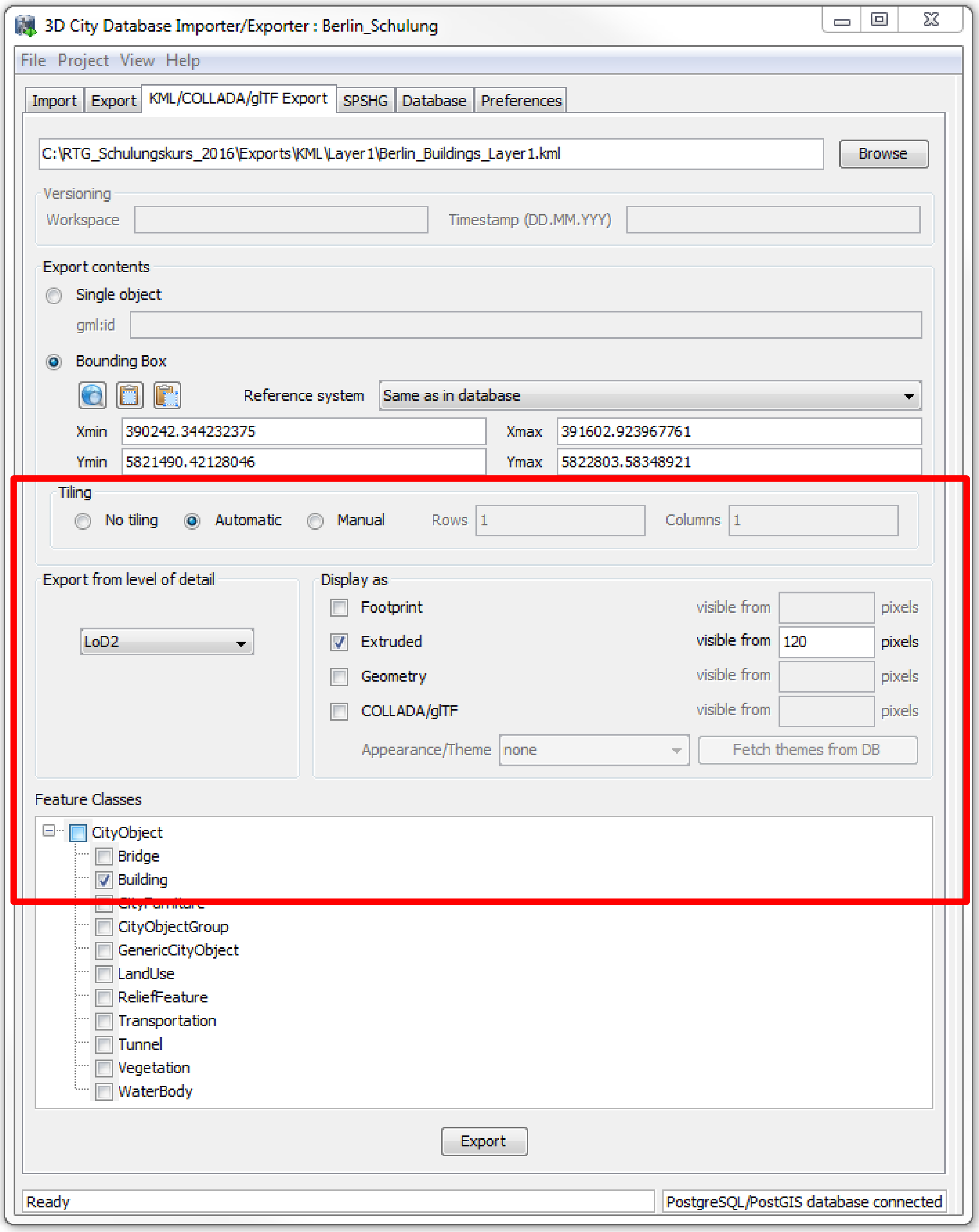
□ Click the Dropdown list ***Export from level of detail***, and select the option ***LoD2***

□ Mark the ***Extruded*** checkbox

□ Fill the ***visible from*** text field with the number value "**120**"

□ In the ***Feature Classes*** panel, we just check the ***Building*** feature type only and uncheck all the other feature types

Once the above settings have been completed, the application window of the Importer/Exporter tool should look like the figure below.



C:\RTG_Schulungskurs_2016\Exports\KML\Layer1\Berlin_Buildings_Layer1.kml

Versioning

Workspace

Timestamp (DD.MM.YYY)

Export contents

○ Single object

gml:id

● Bounding Box

Reference system: Same as in database

Xmin: 390242.344232375

Xmax: 391602.923967761

Ymin: 5821490.42128046

Ymax: 5822803.58348921

Tiling

○ No tiling ● Automatic ○ Manual Rows 1 Columns 1

Export from level of detail: LoD2

Display as

□ Footprint visible from pixels

☑ Extruded visible from 120 pixels

□ Geometry visible from pixels

□ COLLADA/glTF visible from pixels

Appearance/Theme: none

Feature Classes

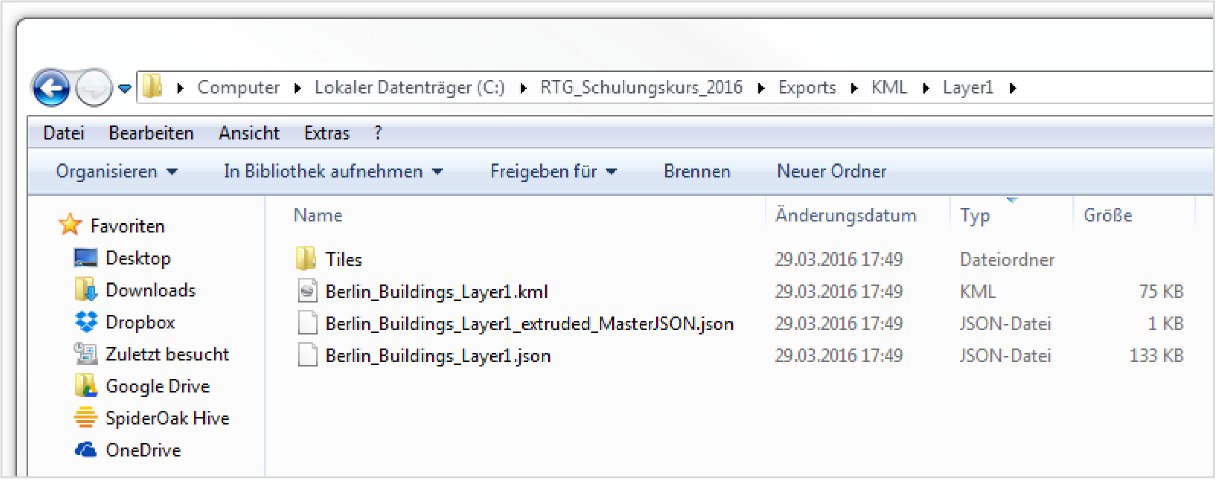
□ CityObject
- □ Bridge
- ☑ Building
- □ CityFurniture
- □ CityObjectGroup
- □ GenericCityObject
- □ LandUse
- □ ReliefFeature
- □ Transportation
- □ Tunnel
- □ Vegetation
- □ WaterBody

Ready

PostgreSQL/PostGIS database connected

□ Click the ***Export*** button to start the export process.

□ After having completed the export process, navigate to the exported KML file by using Windows Explorer. The exported files should like the following figure:


Computer ▸ Lokaler Datenträger (C:) ▸ RTG_Schulungskurs_2016 ▸ Exports ▸ KML ▸ Layer1 ▸
Datei Bearbeiten Ansicht Extras ?
Organisieren ▾ In Bibliothek aufnehmen ▾ Freigeben für ▾ Brennen Neuer Ordner
Favoriten
Desktop
Downloads
Dropbox
Zuletzt besucht
Google Drive
SpiderOak Hive
OneDrive

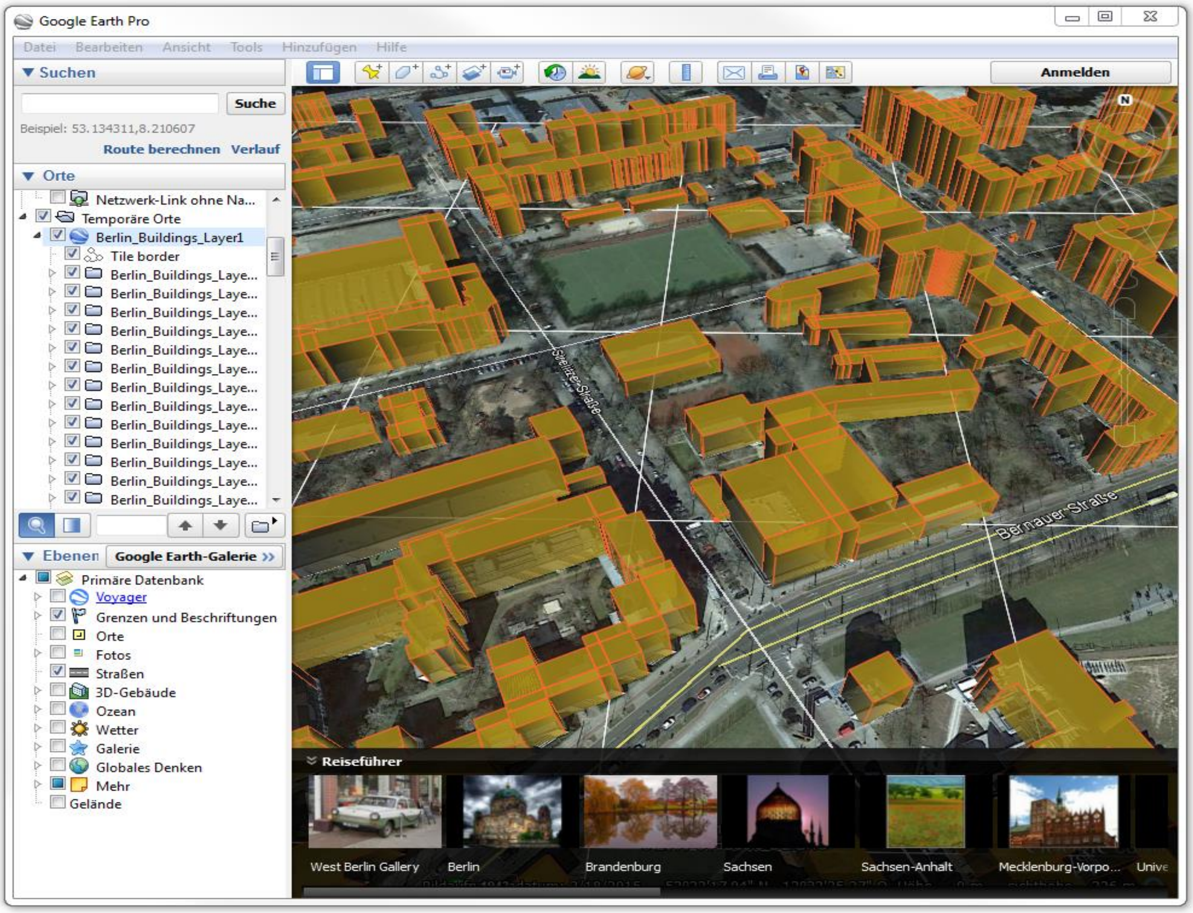
Name	Änderungsdatum	Typ	Größe
Tiles	29.03.2016 17:49	Dateiordner	
Berlin_Buildings_Layer1.kml	29.03.2016 17:49	KML	75 KB
Berlin_Buildings_Layer1_extruded_MasterJSON.json	29.03.2016 17:49	JSON-Datei	1 KB
Berlin_Buildings_Layer1.json	29.03.2016 17:49	JSON-Datei	133 KB

□ Start the *Google Earth Pro* application (you may use the windows ***Start-Button*** to search it)

> **Note**: A pop-up window may appear to ask for typing an email address and license code. Just enter your favourite email address and the license code “**GEPFREE**” into the corresponding input fields.

□ Drag and drop the KML file (***Berlin_Buildings_Layer1.kml***) into the Google Earth Pro application window

The Google Earth view will automatically zoom to the loaded KML data layer which should look like in the following screenshot.



Exercise 6.2 – Export Berlin data in COLLADA/glTF format with textures

In this exercise we will export the Berlin data with higher geometry details as well as with attached textures for realistic rendering of the building objects.

Similar to the last exercise, we firstly need to complete some necessary preference settings.

□ Click the ***Preferences*** tab in the dialog window of the Importer/Exporter tool to open the preferences settings window.

□ Expand the ***KML/COLLADA/glTF Export*** tree node and then click the **General** tree node to open the general settings window (see the screenshot below)

□ Uncheck the option ***Export in .kmz format***

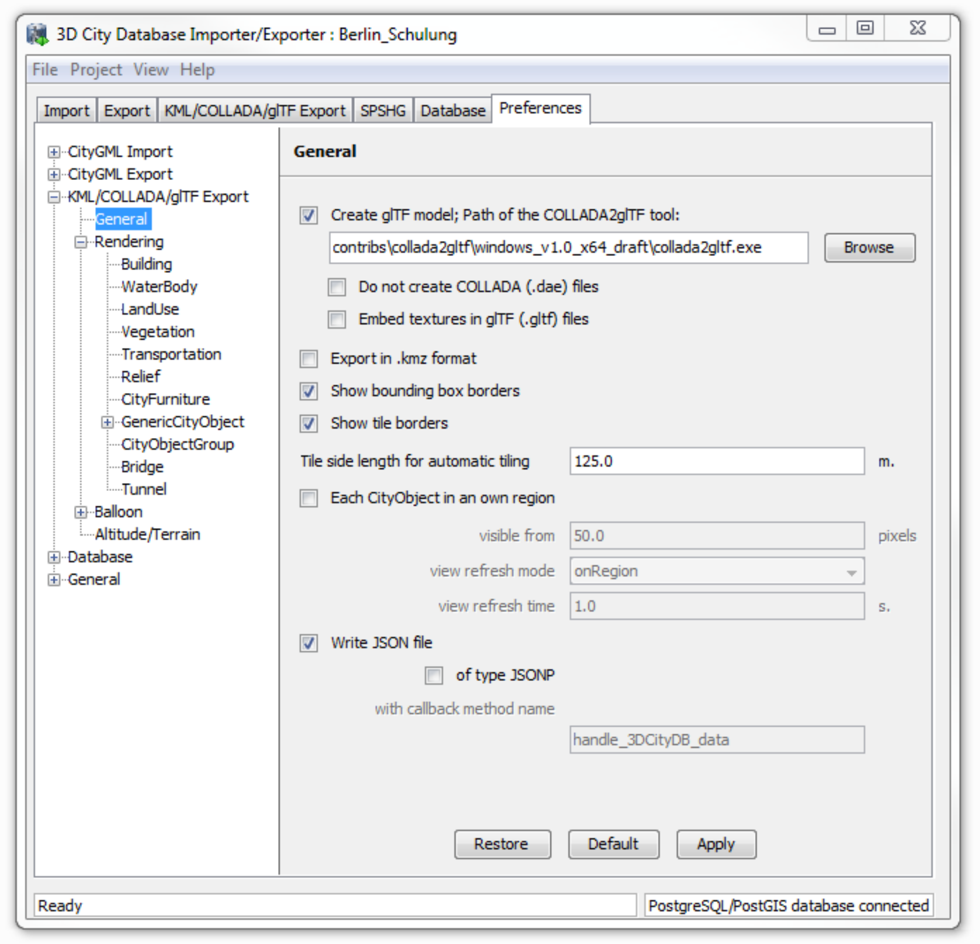
□ Check the option ***Creat glTF model; Path of the COLLADA2glTF tools***

□ Check the option ***Show bounding box borders***

□ Check the option ***Write JSON file***

□ Uncheck the option ***of type JSONP***

□ Click the ***Apply*** button to save the settings which look like the screenshot below:



General

☑ Create glTF model; Path of the COLLADA2glTF tool:

contribs\collada2gltf\windows_v1.0_x64_draft\collada2gltf.exe

☐ Do not create COLLADA (.dae) files

☐ Embed textures in glTF (.gltf) files

☐ Export in .kmz format

☑ Show bounding box borders

☑ Show tile borders

Tile side length for automatic tiling 125.0 m.

☐ Each CityObject in an own region

visible from 50.0 pixels

view refresh mode onRegion

view refresh time 1.0 s.

☑ Write JSON file

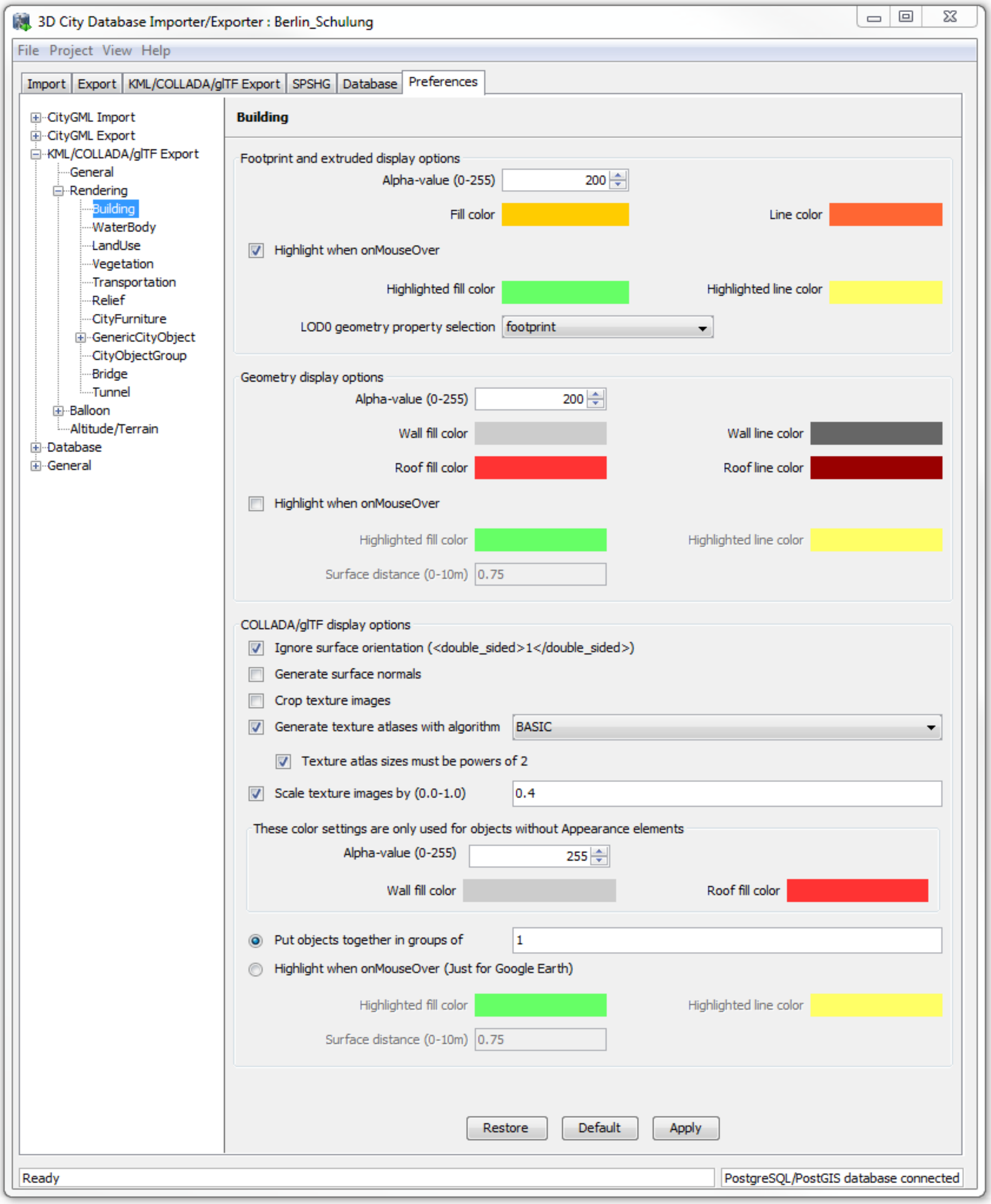
☐ of type JSONP

with callback method name

handle_3DCityDB_data

We also need to configure the appearance settings for COLLADA/glTF export:

□ Expand the ***Rendering*** tree node and click the ***Building*** tree node to open the rendering settings window for the buildings (see the screenshot below)



Building

Footprint and extruded display options

Alpha-value (0-255) 200

Fill color 🟨 Line color 🟧

☑ Highlight when onMouseOver

Highlighted fill color 🟩 Highlighted line color 🟨

LOD0 geometry property selection footprint

Geometry display options

Alpha-value (0-255) 200

Wall fill color ⬜ Wall line color ⬛

Roof fill color 🟥 Roof line color 🟥

□ Highlight when onMouseOver

Highlighted fill color 🟩 Highlighted line color 🟨

Surface distance (0-10m) 0.75

COLLADA/glTF display options

☑ Ignore surface orientation (<double_sided>1</double_sided>)

□ Generate surface normals

□ Crop texture images

☑ Generate texture atlases with algorithm BASIC

☑ Texture atlas sizes must be powers of 2

☑ Scale texture images by (0.0-1.0) 0.4

These color settings are only used for objects without Appearance elements

Alpha-value (0-255) 255

Wall fill color ⬜ Roof fill color 🟥

● Put objects together in groups of 1

○ Highlight when onMouseOver (Just for Google Earth)

Highlighted fill color 🟩 Highlighted line color 🟨

Surface distance (0-10m) 0.75

□ In the ***COLLADA/glTF display*** options panel (see the following figure), Check the option ***Ignore surface orientation (<double_sided>1</double_sided>)***

□ Uncheck the ***Generate surface normals***

□ Check the option ***Generate texture atlases with algorithm***, and click the dropdown list to select the option ***BASIC***

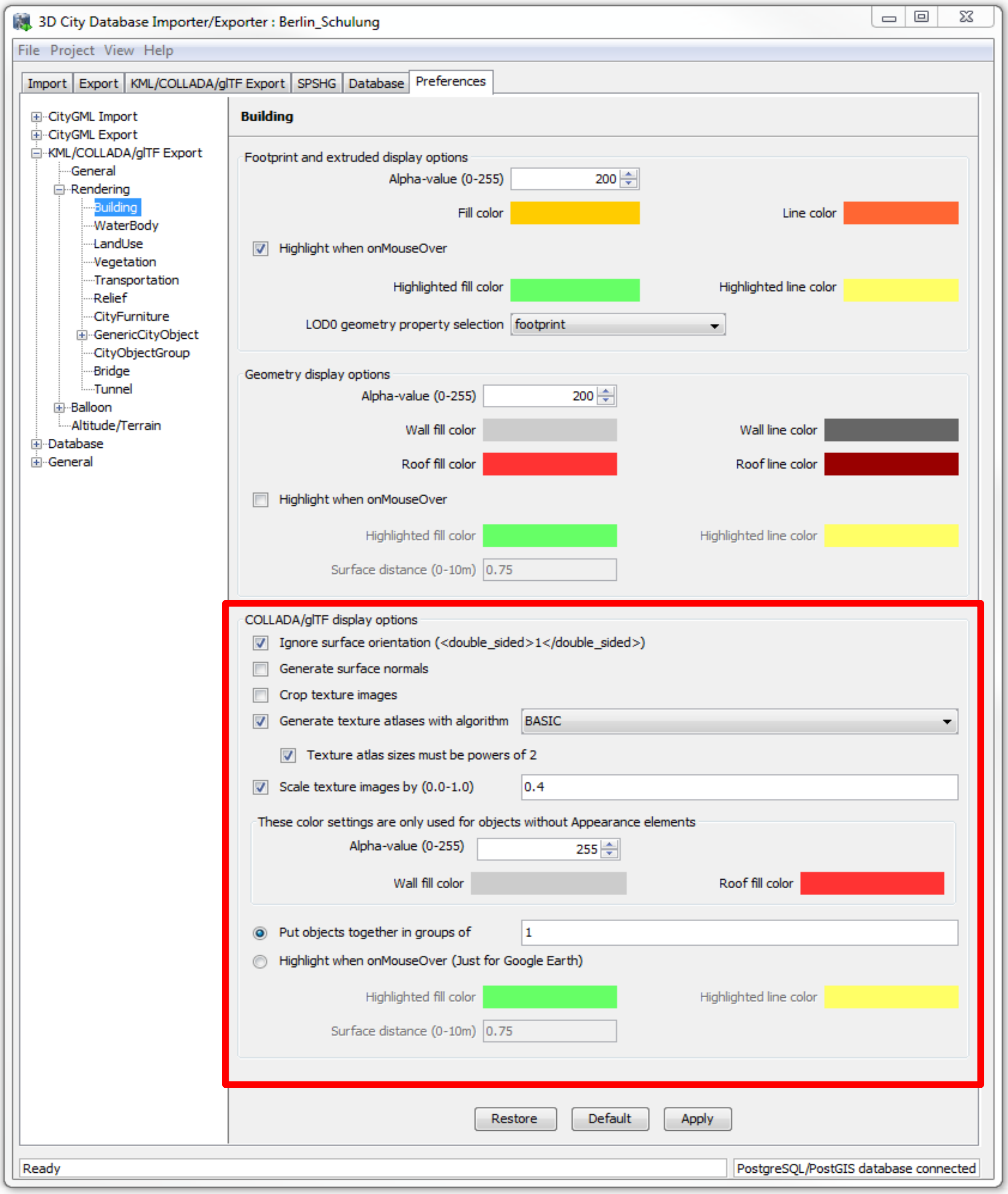
□ Check the option ***Texture atlases sizes must be power of 2***

□ Check the option ***Scale texture images by (0.0-1.0)***, and file the text field on right side with the number value **“0.4”** (which means that texture images should be scaled down to 40% regarding their width and height)

> **Note:** Scaling texture images can significantly reduce the exported file size and increase the loading speed, but scale down the image quality. A scale factor of **0.4** often offers a good balance between these two aspects. Default value is **1.0** (no scaling)

□ Check the radio button ***put objects together in group of***, and enter “**1**” into the text filed on the right side

□ Click ***Apply*** button to save the settings which should look like in the following figure



Building

Footprint and extruded display options

Alpha-value (0-255) 200

Fill color 🟧 Line color 🟧

☑ Highlight when onMouseOver

Highlighted fill color 🟩 Highlighted line color 🟨

LOD0 geometry property selection footprint

Geometry display options

Alpha-value (0-255) 200

Wall fill color ⬜ Wall line color ⬛

Roof fill color 🟥 Roof line color 🟥

☐ Highlight when onMouseOver

Highlighted fill color 🟩 Highlighted line color 🟨

Surface distance (0-10m) 0.75

COLLADA/glTF display options

☑ Ignore surface orientation (<double_sided>1</double_sided>)

☐ Generate surface normals

☐ Crop texture images

☑ Generate texture atlases with algorithm BASIC

☑ Texture atlas sizes must be powers of 2

☑ Scale texture images by (0.0-1.0) 0.4

These color settings are only used for objects without Appearance elements

Alpha-value (0-255) 255

Wall fill color ⬜ Roof fill color 🟥

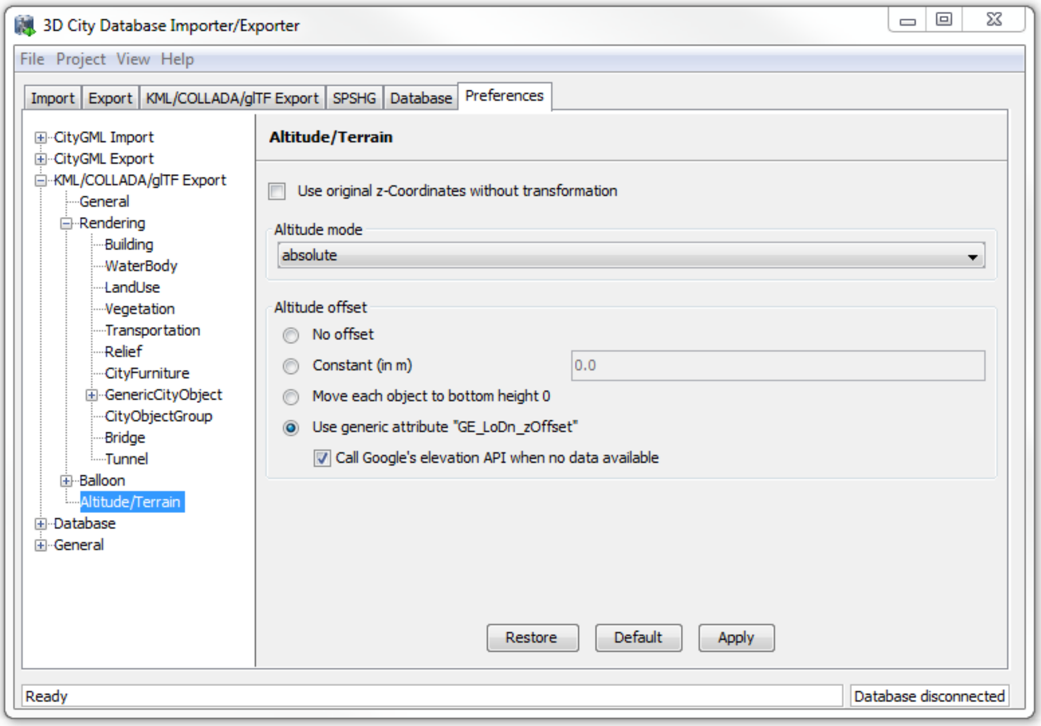
● Put objects together in groups of 1

○ Highlight when onMouseOver (Just for Google Earth)

Highlighted fill color 🟩 Highlighted line color 🟨

Surface distance (0-10m) 0.75

□ Click the **Altitude/Terrain** tree node to open the Altitude and Terrain settings window (see the screenshot below)



Altitude/Terrain

□ Use original z-Coordinates without transformation

Altitude mode: absolute

Altitude offset

○ No offset

○ Constant (in m) 0.0

○ Move each object to bottom height 0

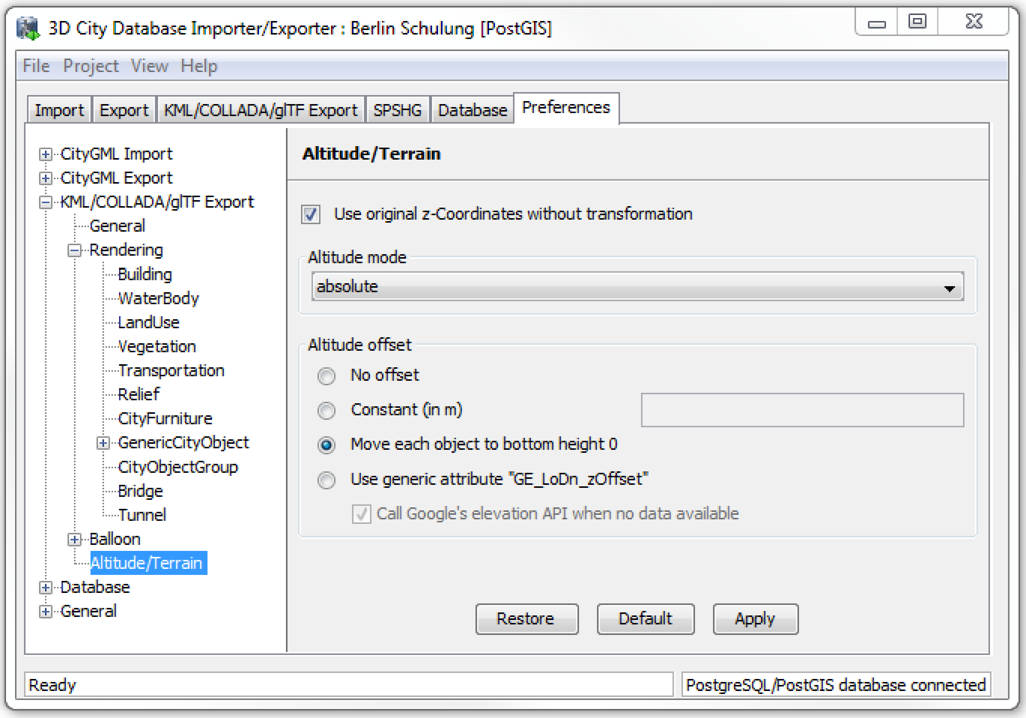
● Use generic attribute "GE_LoDn_zOffset"

☑ Call Google's elevation API when no data available

□ Check the option ***Use original z-Coordinates without transformation***

□ Check the radio button ***Move each object to bottom height 0***

□ Click the ***Apply*** button to save the settings which should look like in the screenshot below



Altitude/Terrain

☑ Use original z-Coordinates without transformation

Altitude mode: absolute

Altitude offset

○ No offset

○ Constant (in m)

● Move each object to bottom height 0

○ Use generic attribute "GE_LoDn_zOffset"

☑ Call Google's elevation API when no data available

□ Click the ***KML/COLLADA/glTF*** tab to go back to the KML/COLLADA/glTF-Exporter window

□ Click the ***Browse*** button to open the file selection window to choose an output path and filename for the output KML file. We simply specify it using the following path and filename:

"***C:\RTG_Schulungskurs_2016\Exports\KML\Layer2\Berlin_Buildings_Layer2.kml***"

Since you have already specified the bounding box for the KML export in the previous exercise 5.1, you don't need to change the bounding box value in this step.

□ Check the ***Automatic*** radio button

□ Click the Dropdown list ***Export from level of detail***, and select the option ***LoD2***

□ Check the ***COLLADA/glTF*** checkbox,

□ Uncheck the other display options like ***Footprint***, ***Extruded***, and ***Geometry***

□ Fill the ***visible from*** text field next to the ***COLLADA/glTF*** display option with the number value "**120**"

□ Click the ***Fetch themes from DB*** button

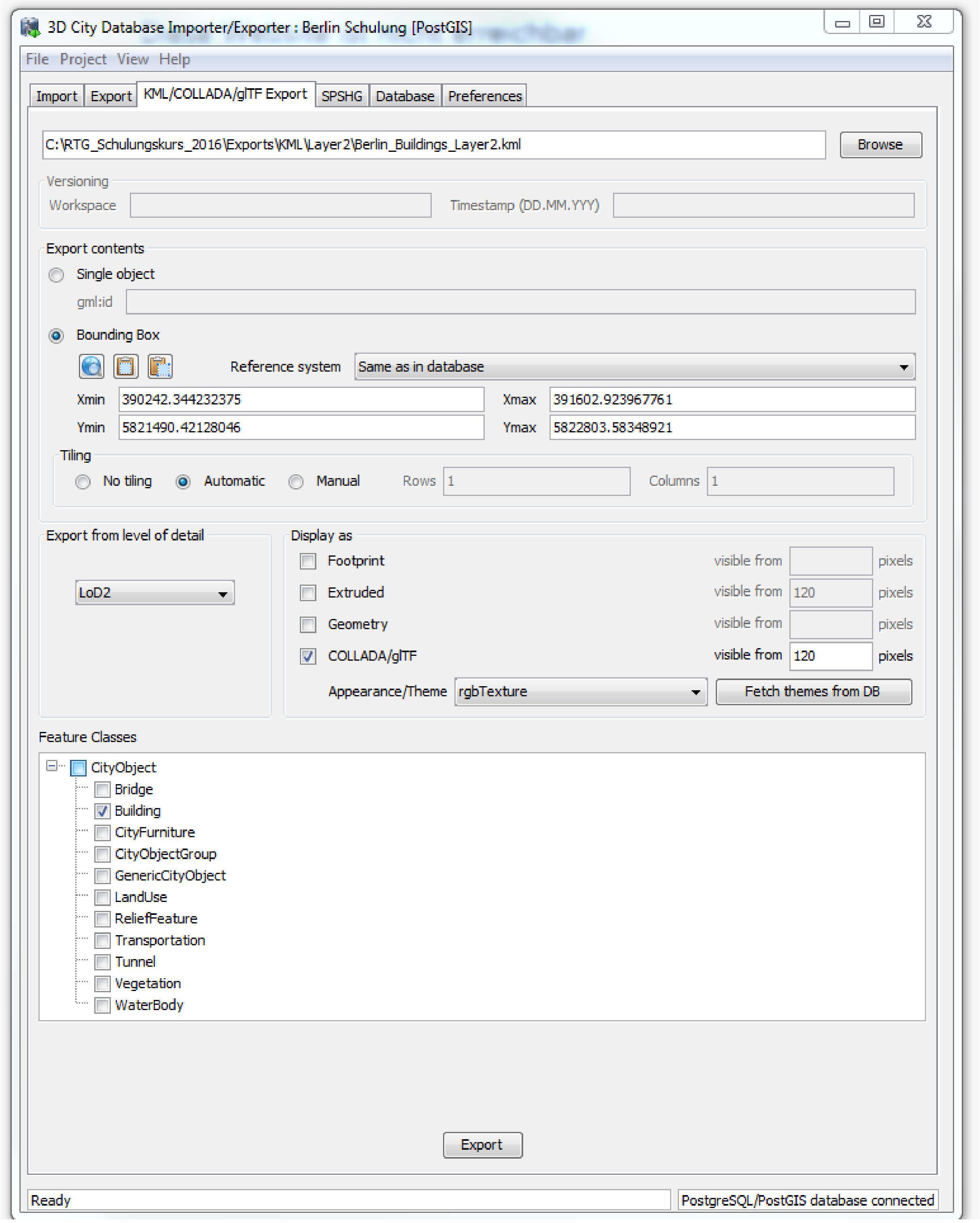
The dropdown list ***Appearance/Theme*** will be enabled

□ Click the dropdown list ***Appearance/Theme*** and select the option ***rgbTexture***

> **Note**: There are two additional appearance themes (***Eignung Photovoltaik*** and ***Eignung Solarthermie***) available in the Berlin data. You may try these two options later.

□ In the ***Feature Classes*** panel we just check the ***Building*** feature type and uncheck all the other feature types

Once the above settings have been completed, the application window of the Importer/Exporter tool should look like in the figure below.



3D City Database Importer/Exporter : Berlin Schulung [PostGIS]

C:\RTG_Schulungskurs_2016\Exports\KML\Layer2\Berlin_Buildings_Layer2.kml

Versioning

Workspace:

Timestamp (DD.MM.YYY):

Export contents

○ Single object

gml:id:

● Bounding Box

Reference system: Same as in database

Xmin: 390242.344232375 Xmax: 391602.923967761

Ymin: 5821490.42128046 Ymax: 5822803.58348921

Tiling

○ No tiling ● Automatic ○ Manual Rows: 1 Columns: 1

Export from level of detail: LoD2

Display as

☐ Footprint visible from pixels

☐ Extruded visible from 120 pixels

☐ Geometry visible from pixels

☑ COLLADA/glTF visible from 120 pixels

Appearance/Theme: rgbTexture

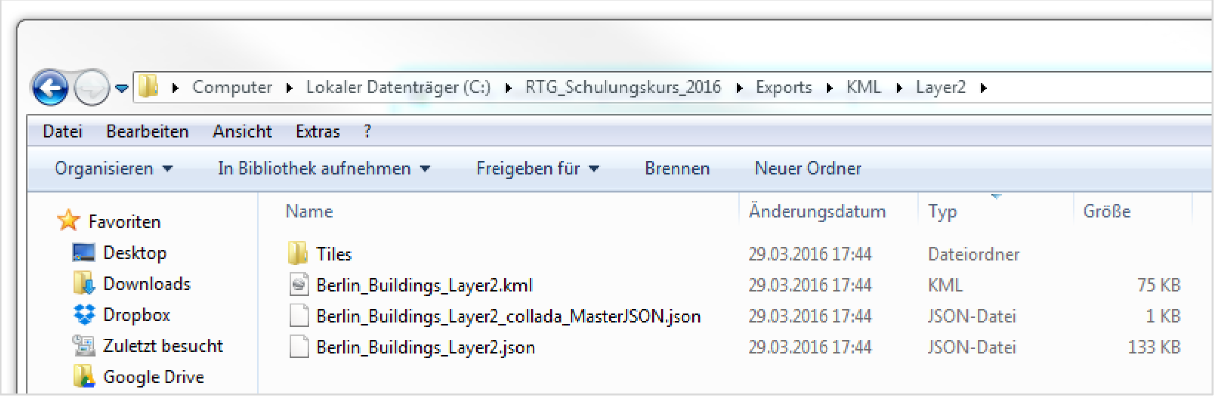
Feature Classes

- ☐ CityObject
  - ☐ Bridge
  - ☑ Building
  - ☐ CityFurniture
  - ☐ CityObjectGroup
  - ☐ GenericCityObject
  - ☐ LandUse
  - ☐ ReliefFeature
  - ☐ Transportation
  - ☐ Tunnel
  - ☐ Vegetation
  - ☐ WaterBody

Ready | PostgreSQL/PostGIS database connected

□ Click the ***Export*** button to start the export process.

□ After the export process, navigate to the exported KML file by using Windows Explorer. The exported files should look like the following figure:



Computer ▸ Lokaler Datenträger (C:) ▸ RTG_Schulungskurs_2016 ▸ Exports ▸ KML ▸ Layer2 ▸

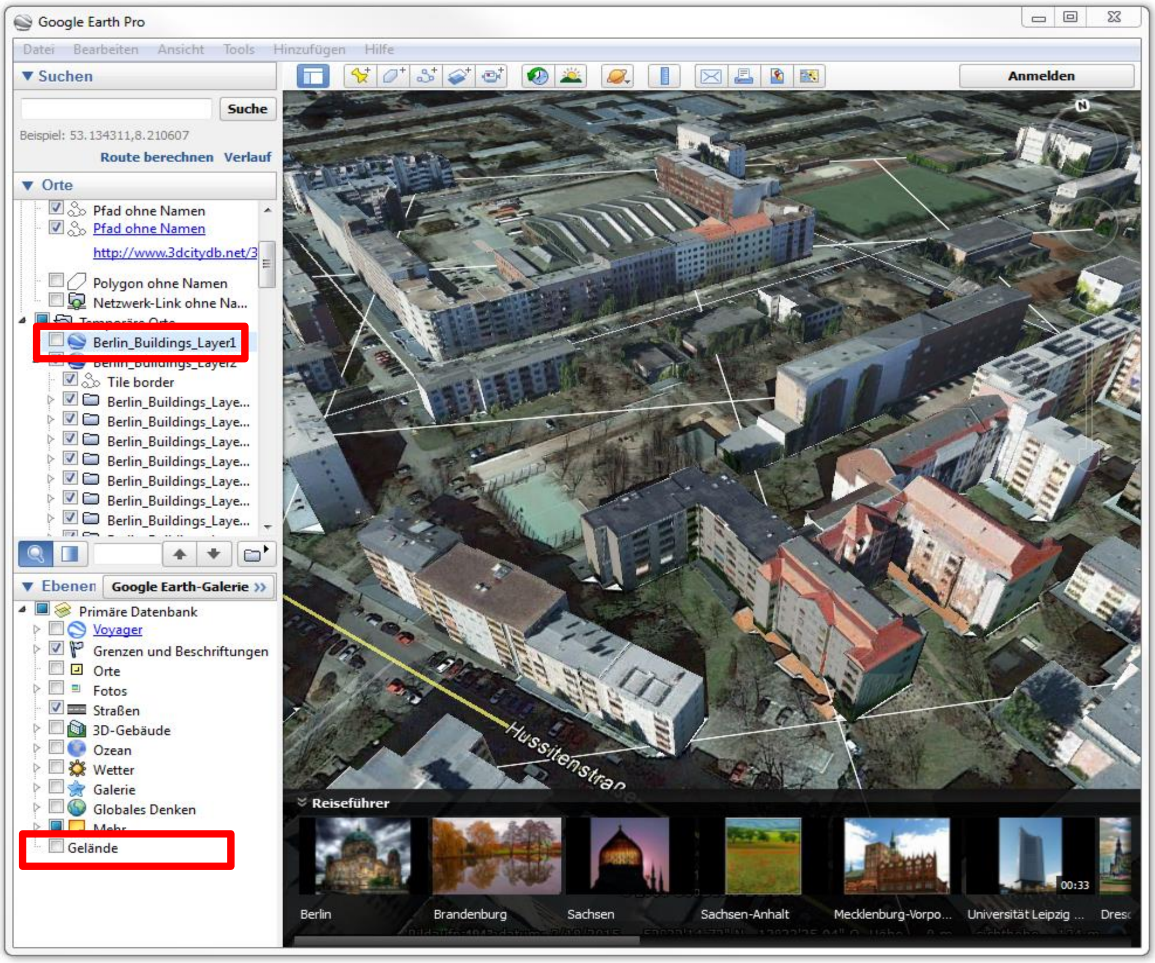
Name	Änderungsdatum	Typ	Größe
Tiles	29.03.2016 17:44	Dateiordner	
Berlin_Buildings_Layer2.kml	29.03.2016 17:44	KML	75 KB
Berlin_Buildings_Layer2_collada_MasterJSON.json	29.03.2016 17:44	JSON-Datei	1 KB
Berlin_Buildings_Layer2.json	29.03.2016 17:44	JSON-Datei	133 KB

□ Start the *Google Earth Pro* application

□ Uncheck the ***Gelände*** option on the lower-left corner of the Google Earth Pro application window (marked in the screenshot below)

□ Drag and drop the .KML file (***Berlin_Buildings_Layer2.kml***) into the Google Earth Pro application window

The exported KML data will be visualized in the Google Earth Globe, you may need to uncheck the Google Earth layer tree node ***Berlin_Buildings_Layer1*** (marked in the screenshot below) to hide the earlier opened layer from exercise 6.1.



Exercise 6.3 – Export Berlin data in COLLADA/glTF format without textures

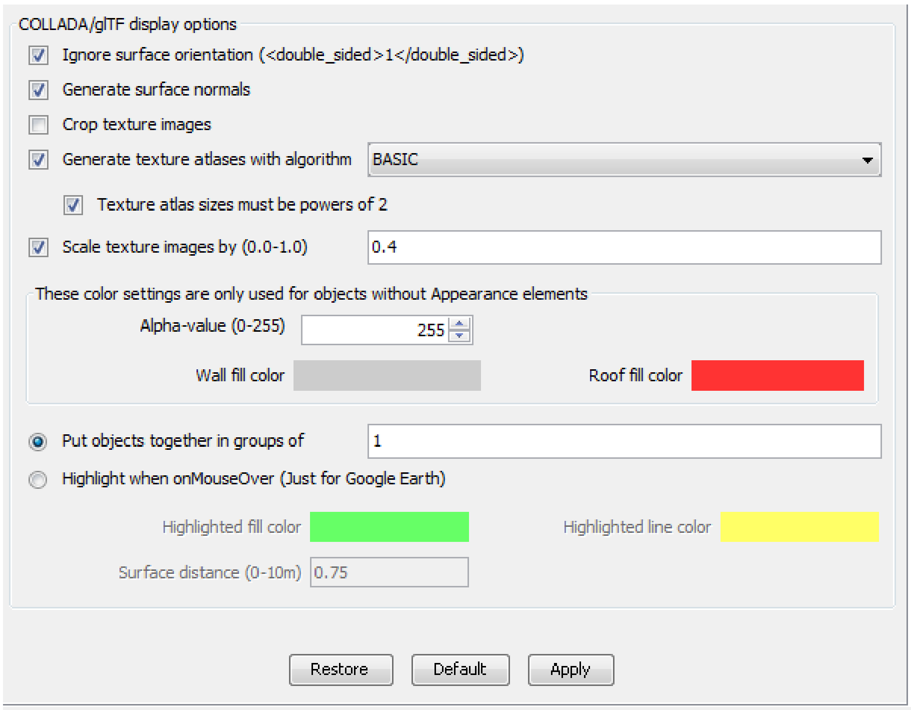
It is also possible to export the Berlin data in COLLADA/glTF format without textures. For this purpose, we just need to change a few settings mentioned in the last exercises.

□ Click the ***Preferences*** tab in the application window of the Importer/Exporter tool to open the preferences settings window.

□ Expand the ***Rendering*** tree node and click the ***Building*** tree node to open the rendering settings window for the buildings

□ Check the ***Generate surface normal***

The completed settings should look like the figure below:



COLLADA/glTF display options

☑ Ignore surface orientation (<double_sided>1</double_sided>)

☑ Generate surface normals

□ Crop texture images

☑ Generate texture atlases with algorithm: BASIC

☑ Texture atlas sizes must be powers of 2

☑ Scale texture images by (0.0-1.0): 0.4

These color settings are only used for objects without Appearance elements

Alpha-value (0-255): 255

Wall fill color ⬜ Roof fill color 🟥

● Put objects together in groups of: 1

○ Highlight when onMouseOver (Just for Google Earth)

Highlighted fill color 🟩 Highlighted line color 🟨

Surface distance (0-10m): 0.75

□ Click the ***Apply*** button to save the settings

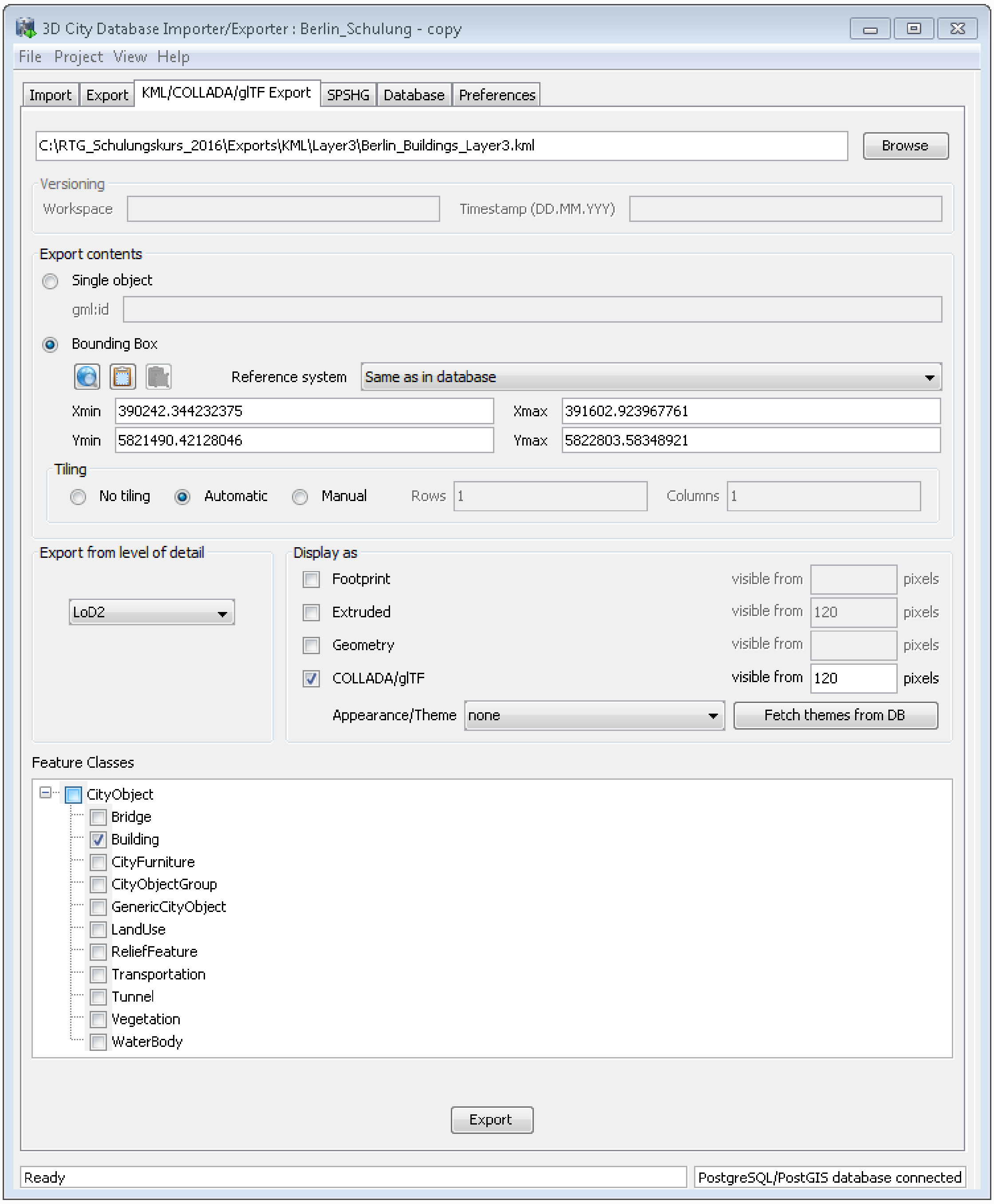
□ Click the ***KML/COLLADA/glTF*** tab to go back to the KML/COLLADA/glTF-Exporter window

□ Click the ***Browse*** button to open the file selection window to choose an output path and filename for the output KML file. We simply specify it using the following path and filename:

"C:\***RTG_Schulungskurs_2016\Exports\KML\Layer3\Berlin_Buildings_Layer3.kml***"

□ Click the dropdown list ***Appearance/Theme*** and select the option ***none***

The completed settings should look like in the following figure:



C:\RTG_Schulungskurs_2016\Exports\KML\Layer3\Berlin_Buildings_Layer3.kml

Versioning
Workspace Timestamp (DD.MM.YYY)

Export contents
○ Single object
gml:id
● Bounding Box
Reference system: Same as in database
Xmin 390242.344232375 Xmax 391602.923967761
Ymin 5821490.42128046 Ymax 5822803.58348921
Tiling
○ No tiling ● Automatic ○ Manual Rows 1 Columns 1

Export from level of detail
LoD2

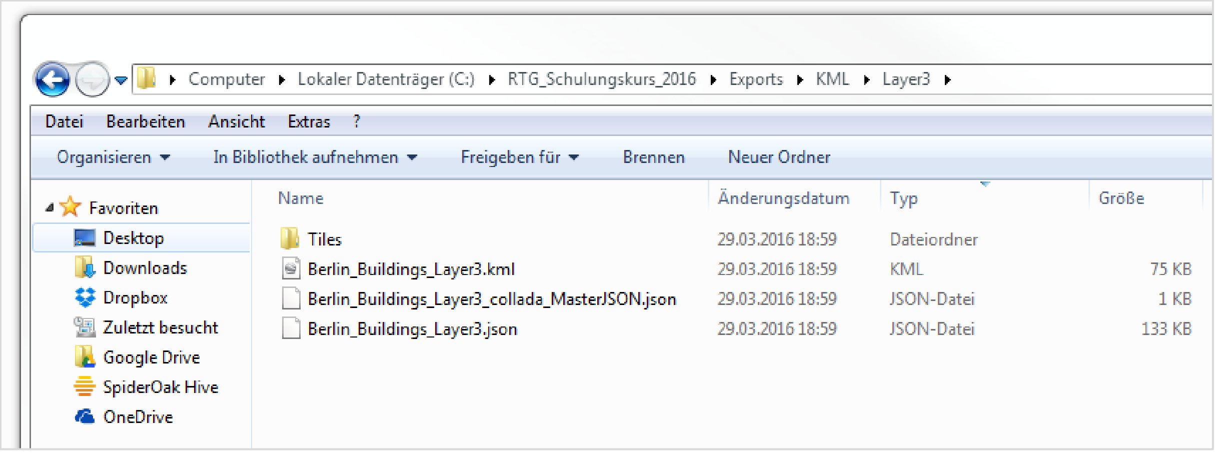
Display as
☐ Footprint visible from pixels
☐ Extruded visible from 120 pixels
☐ Geometry visible from pixels
☑ COLLADA/glTF visible from 120 pixels
Appearance/Theme none

Feature Classes
- ☐ CityObject
  - ☐ Bridge
  - ☑ Building
  - ☐ CityFurniture
  - ☐ CityObjectGroup
  - ☐ GenericCityObject
  - ☐ LandUse
  - ☐ ReliefFeature
  - ☐ Transportation
  - ☐ Tunnel
  - ☐ Vegetation
  - ☐ WaterBody

Ready | PostgreSQL/PostGIS database connected

□ Click the ***Export*** button to start the export process

□ After the export process, navigate to the exported KML file by using Windows Explorer. The exported files should like in the following figure:



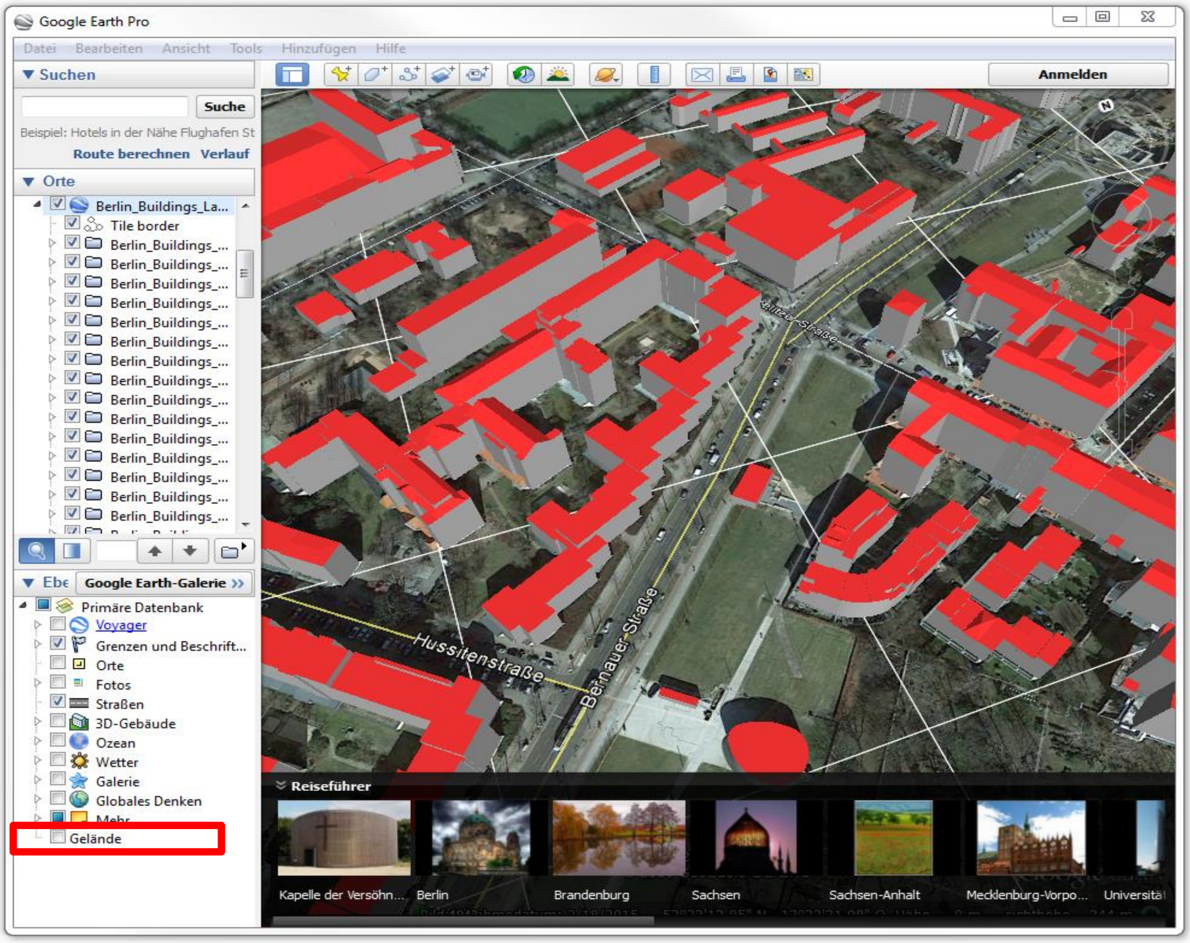
Name	Änderungsdatum	Typ	Größe
Tiles	29.03.2016 18:59	Dateiordner	
Berlin_Buildings_Layer3.kml	29.03.2016 18:59	KML	75 KB
Berlin_Buildings_Layer3_collada_MasterJSON.json	29.03.2016 18:59	JSON-Datei	1 KB
Berlin_Buildings_Layer3.json	29.03.2016 18:59	JSON-Datei	133 KB

□ Start the *Google Earth Pro* application

□ Uncheck the ***Gelände (or terrain)*** option on the lower-left corner of the Google Earth Pro application window (marked in the screenshot below)

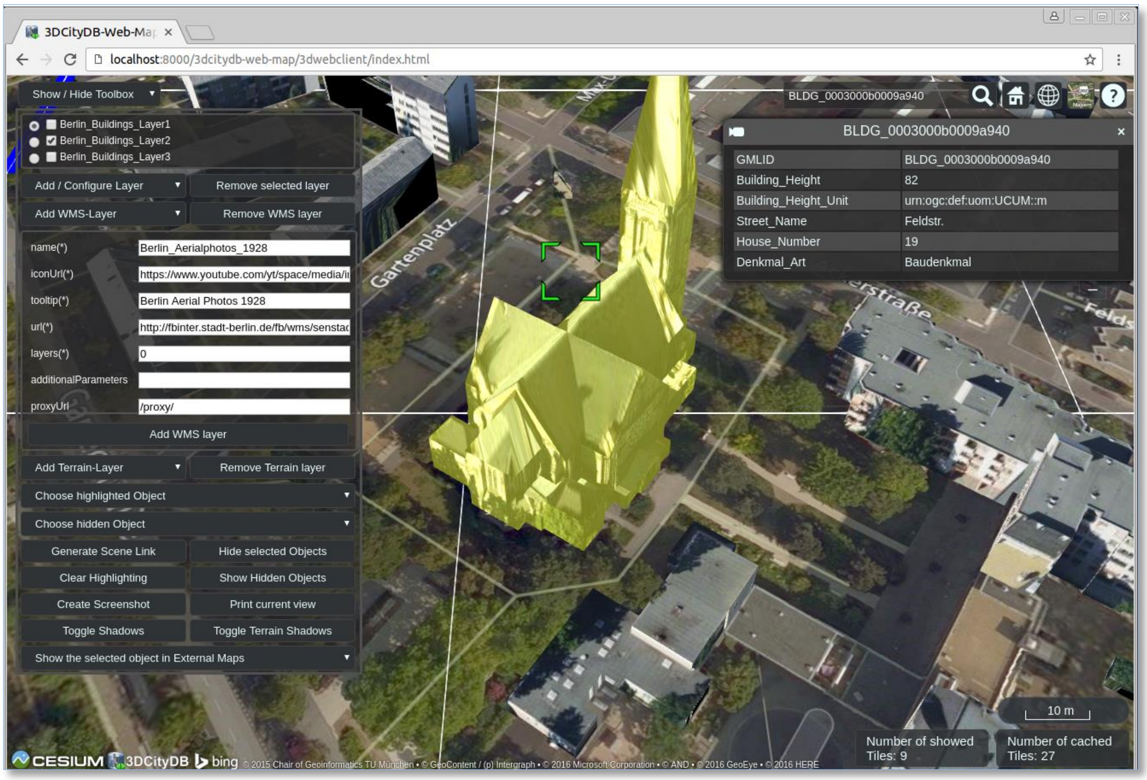
□ Drag and drop the created KML file (***Berlin_Buildings_Layer3.kml***) into the Google Earth Pro application window

The exported KML data will be visualized in the Google Earth Globe, you may need to uncheck the Google Earth layer tree node ***Berlin_Buildings_Layer2*** to hide the earlier opened layer from exercise 6.2.



Exercise 7: 3D Visualization using the 3D Web Client

The 3D City Database software package provides an interactive **3D web client (3DCityDB-Web-Map-Client)** using the Cesium Virtual Globe framework which is based on HTML5 and WebGL technology and which runs in the most modern browsers and on different operating system. The 3D web client links 3D visualization models exported in KML/COLLADA/glTF format with online spreadsheets exported using the 3DCityDB Spreadsheet Generator and allows interactive viewing and querying objects and their thematic data. The following image shows a screenshot of the 3D web client displaying the Berlin data.



localhost:8000/3dcitydb-web-map/3dwebclient/index.html

○ □ Berlin_Buildings_Layer1
● ☑ Berlin_Buildings_Layer2
○ □ Berlin_Buildings_Layer3

name(*): Berlin_Aerialphotos_1928
iconUrl(*): https://www.youtube.com/yt/space/media/i
tooltip(*): Berlin Aerial Photos 1928
url(*): http://fbinter.stadt-berlin.de/fb/wms/senstad
layers(*): 0
additionalParameters:
proxyUrl: /proxy/

BLDG_0003000b0009a940

GMLID	BLDG_0003000b0009a940
Building_Height	82
Building_Height_Unit	urn:ogc:def:uom:UCUM::m
Street_Name	Feldstr.
House_Number	19
Denkmal_Art	Baudenkmal

Number of showed Tiles: 9
Number of cached Tiles: 27

In this exercise you will learn how to use the 3D web client to visualize and explore the KML/glTF Berlin data you've created in the last exercise.

Exercise 7.1 – Setting up the 3D web client on your computer

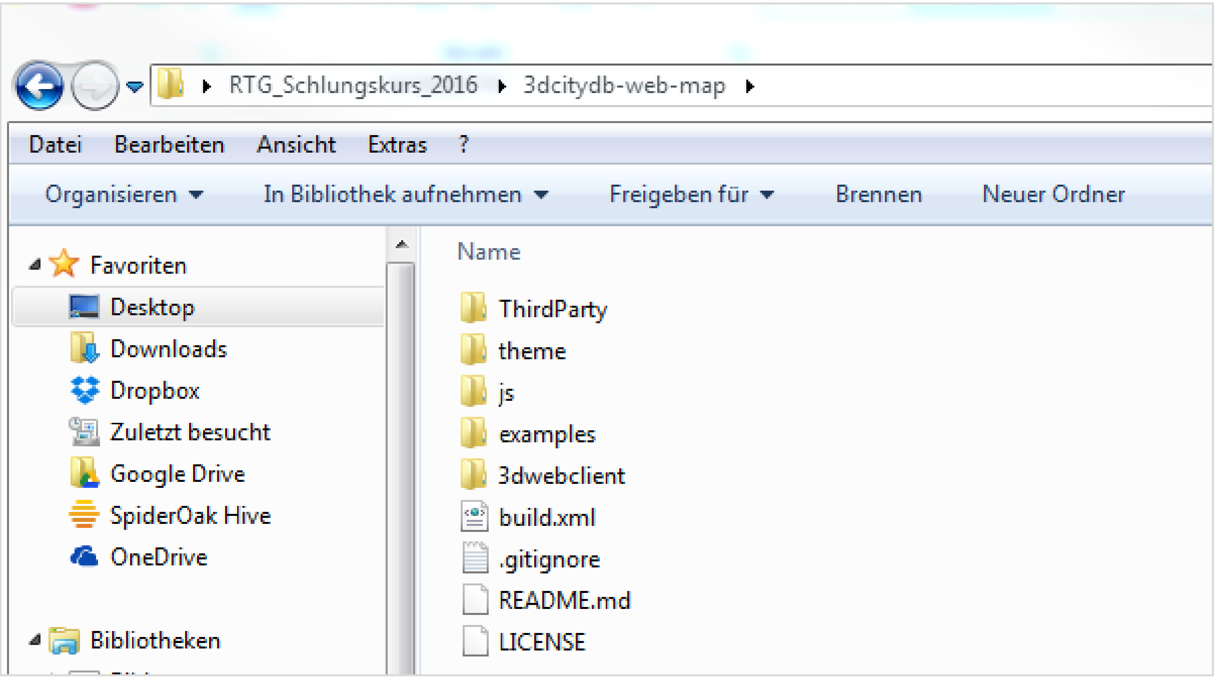
□ A zip file for the 3D web client can be found in the installation directory of the Import/Export tool within the subfolder ***3d-web-map-client*** or downloaded via the following GitHub link:

https://github.com/3dcitydb/3dcitydb-web-map/archive/v1.1.0.zip

□ Extract the zip file into the following directory:

"C:\RTG_Schulungskurs_2016 \3dcitydb-web-map"

The contents in the ***3dcitydb-web-map*** directory should look like in the following figure:

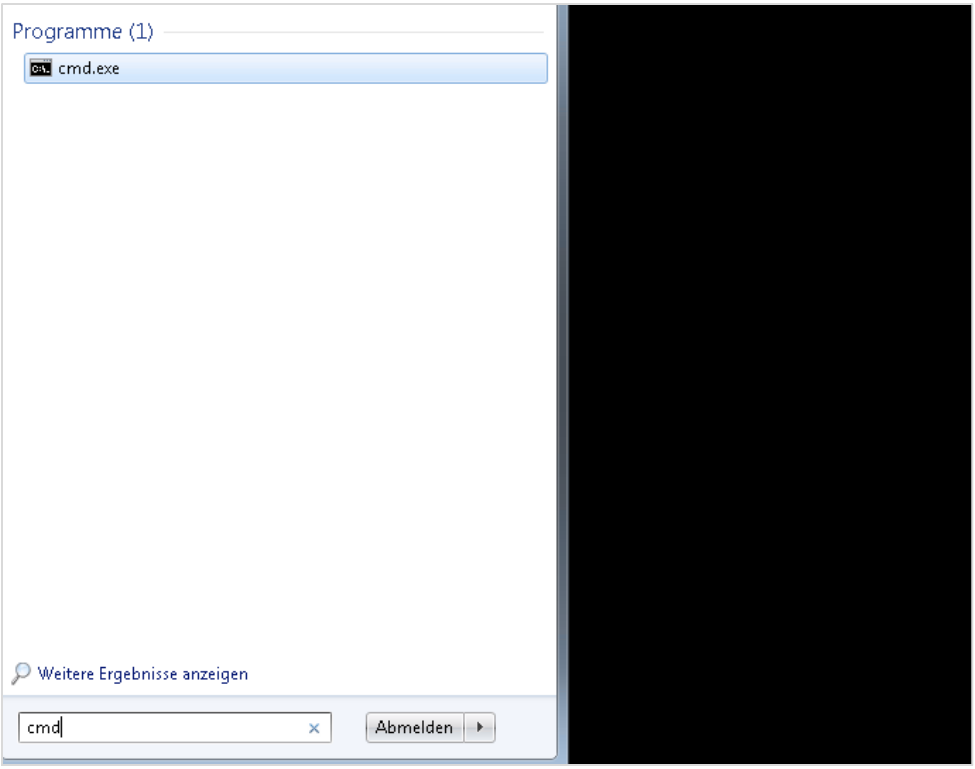


In order to run the 3D web client, you need to set up your own minimal web server to host the web client files. In this tutorial, we'll use a lightweight local web server running on top of the ***Node.js*** runtime environment.

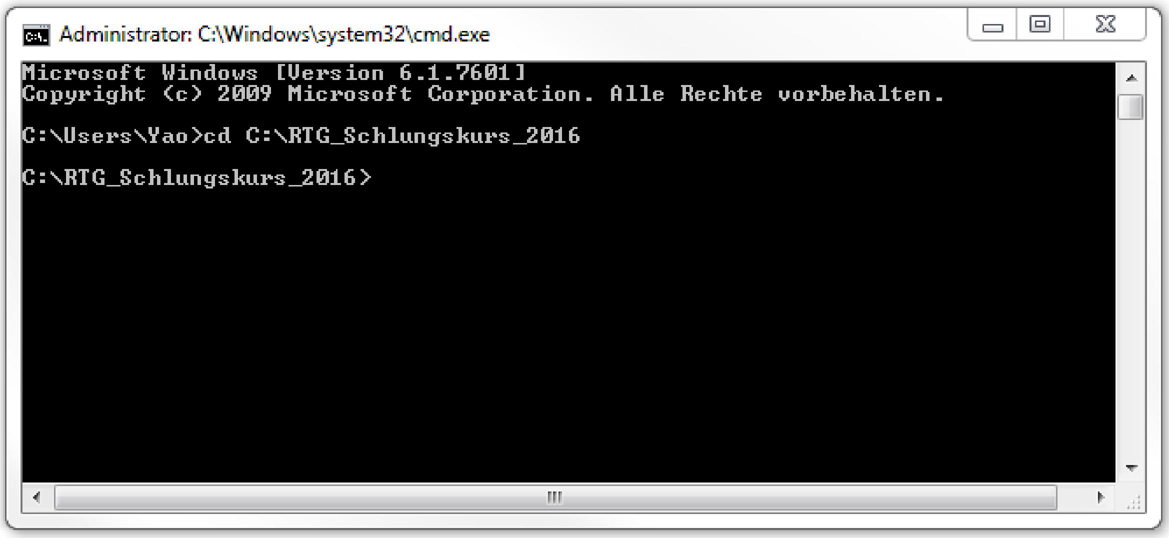
Let's run the local web server by completing the following steps:

□ Click the ***Windows Start*** button and enter "***cmd***" into the search text field

□ Click the ***cmd.exe*** in the search results (see the following figure)

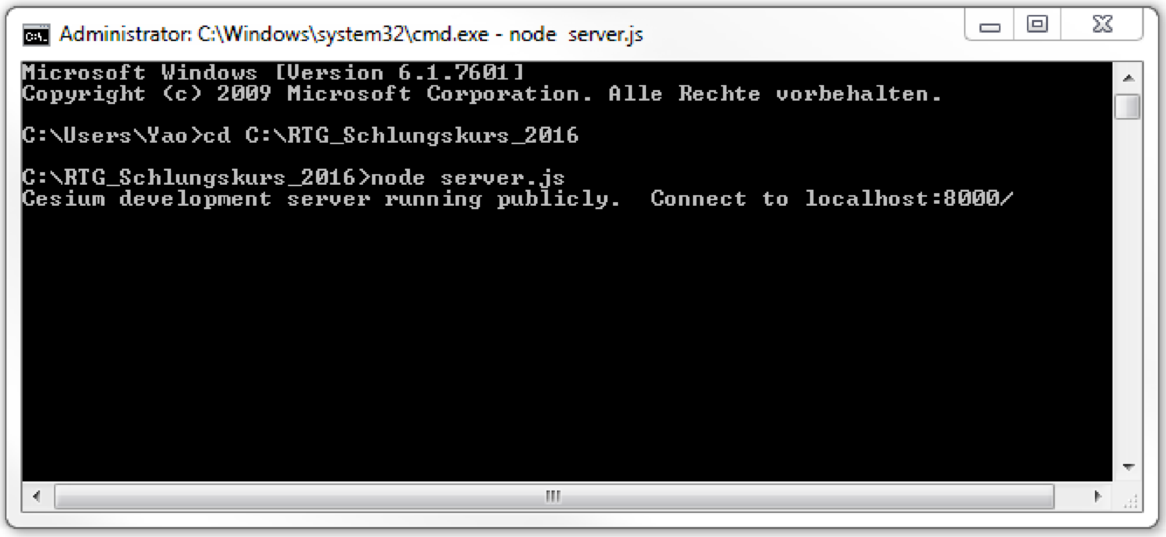


□ Type “**cd C:\RTG_Schulungskurs_2016**” into the command shell and press ***ENTER*** key to go to our working directory (see the screenshot below)



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Yao>cd C:\RTG_Schlungskurs_2016

C:\RTG_Schlungskurs_2016>
```

□ Type “**node server.js**” into the command shell and press **ENTER** key to run the local server (see the screenshot below)



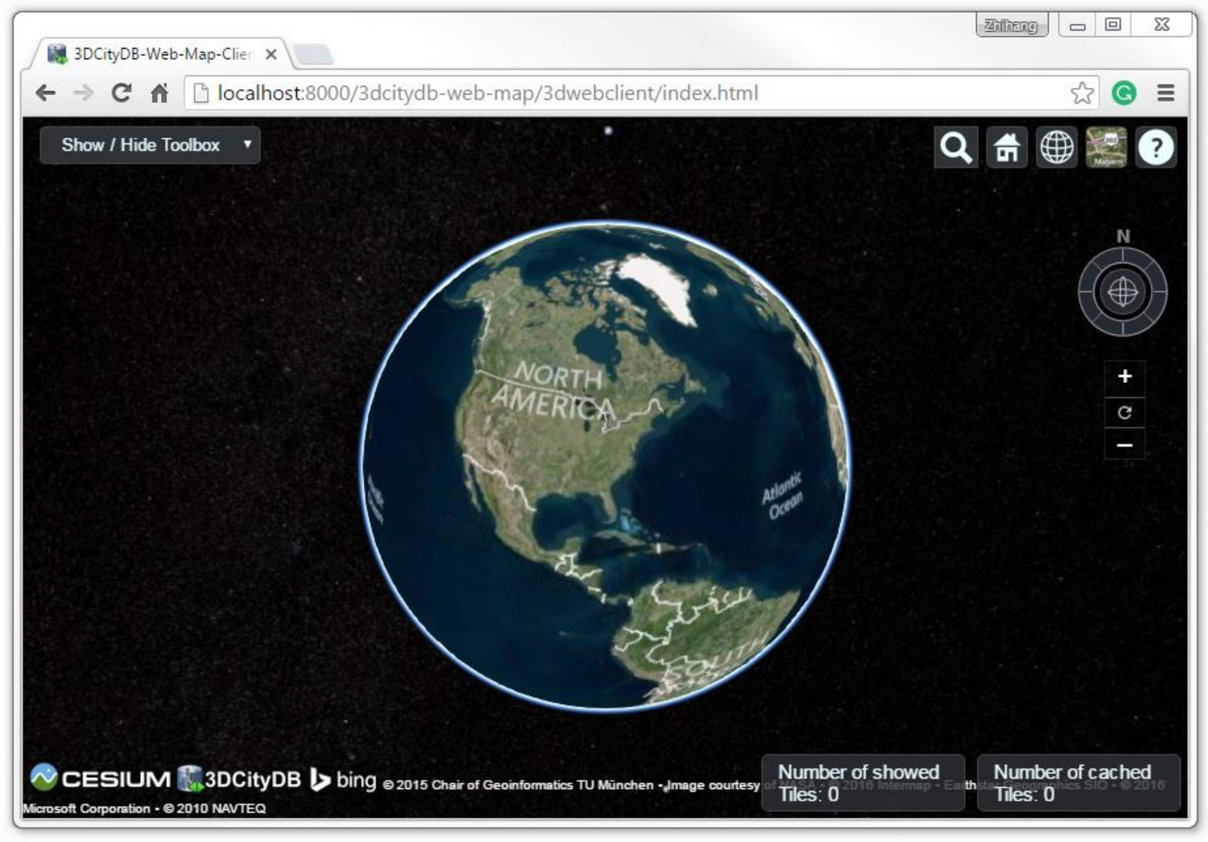
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Yao>cd C:\RTG_Schlungskurs_2016

C:\RTG_Schlungskurs_2016>node server.js
Cesium development server running publicly.  Connect to localhost:8000/
```

Now, we have a local web server up and running. **Please don’t close this command window.** Let’s launch a web browser and open our 3D web client!

□ Open a web browser (for example Google Chrome) and navigate to the 3D web client by entering the following link address into the address bar

http://localhost:8000/3dcitydb-web-map/3dwebclient/index.html

After opening the link you can see the user interface of the web client (see the following screenshot)

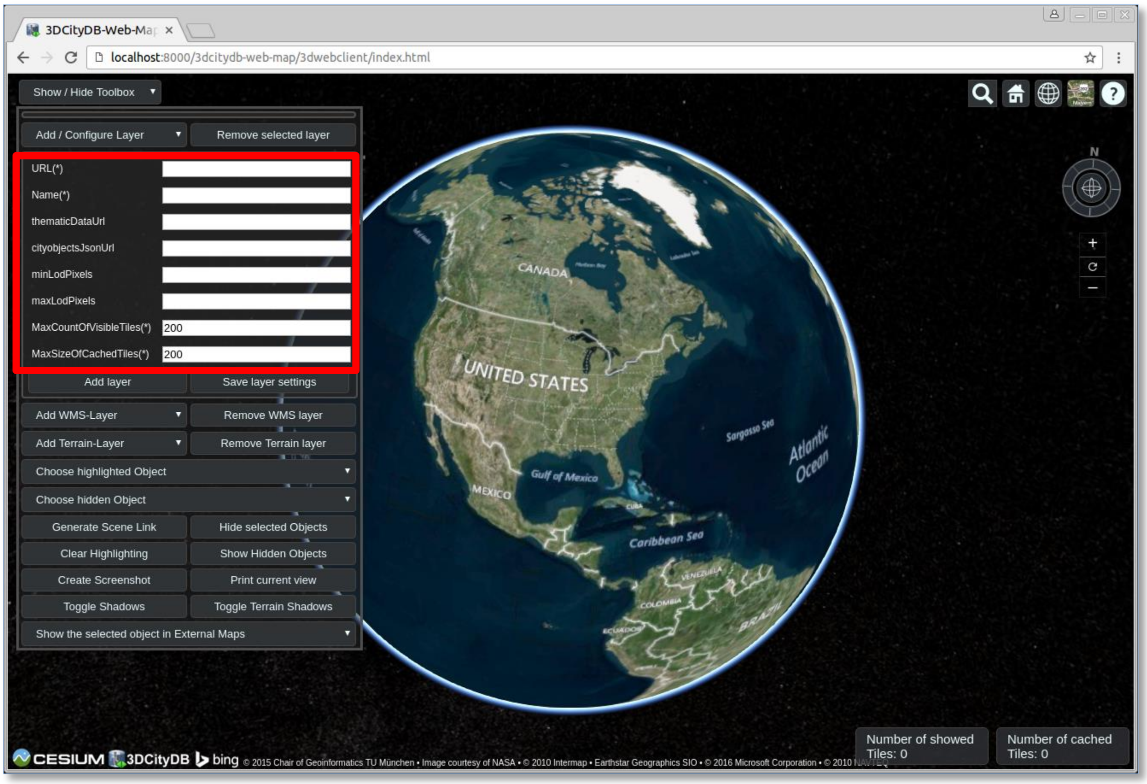

3DCityDB-Web-Map-Clie
Zhihang
localhost:8000/3dcitydb-web-map/3dwebclient/index.html
Show / Hide Toolbox
N
NORTH AMERICA
Atlantic Ocean
CESIUM
3DCityDB
bing
© 2015 Chair of Geoinformatics TU München • Image courtesy
Microsoft Corporation • © 2010 NAVTEQ
Number of showed Tiles: 0
Number of cached Tiles: 0

Exercise 7.2 – Visualization of multiple KML/glTF layers

Now, let’s load the Berlin KML and spreadsheet data (created in the previous exercises) into our 3D web client.

□ Click the ***Show / Hide Toolbox*** button on the top-left corner to open the toolbox panel

□ Click the ***Add / Configure Layer*** button to open the layer settings panel (marked in the following figure)



MaxCountOfVisibleTiles(*) 200

MaxSizeOfCachedTiles(*) 200

Number of showed Tiles: 0

Number of cached Tiles: 0

□ Fill the text field ***URL(*)*** with the following data link which references the exported extruded KML data of Berlin

http://localhost:8000/Exports/KML/Layer1/Berlin_Buildings_Layer1_extruded_MasterJSON.json

□ Fill the text field ***Name(*)*** with the text value "***Berlin_Buildings_Layer1***"

□ Fill the text field ***thematicDataUrl*** with the web link of the created Google Fusion Table (please refer to Exercise 5)

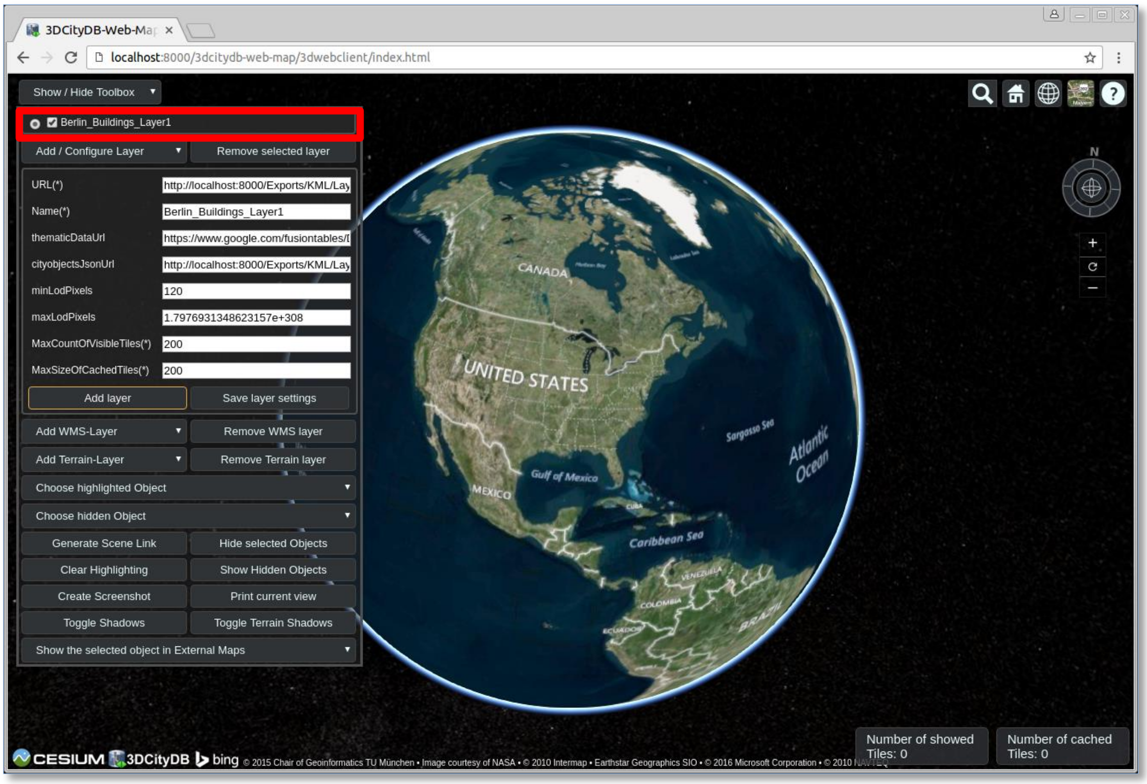
□ Fill the text field ***cityobjectsJsonUrl*** with the data link of the created json data

http://localhost:8000/Exports/KML/Layer1/Berlin_Buildings_Layer1.json

□ Keep the other text fields as the default settings

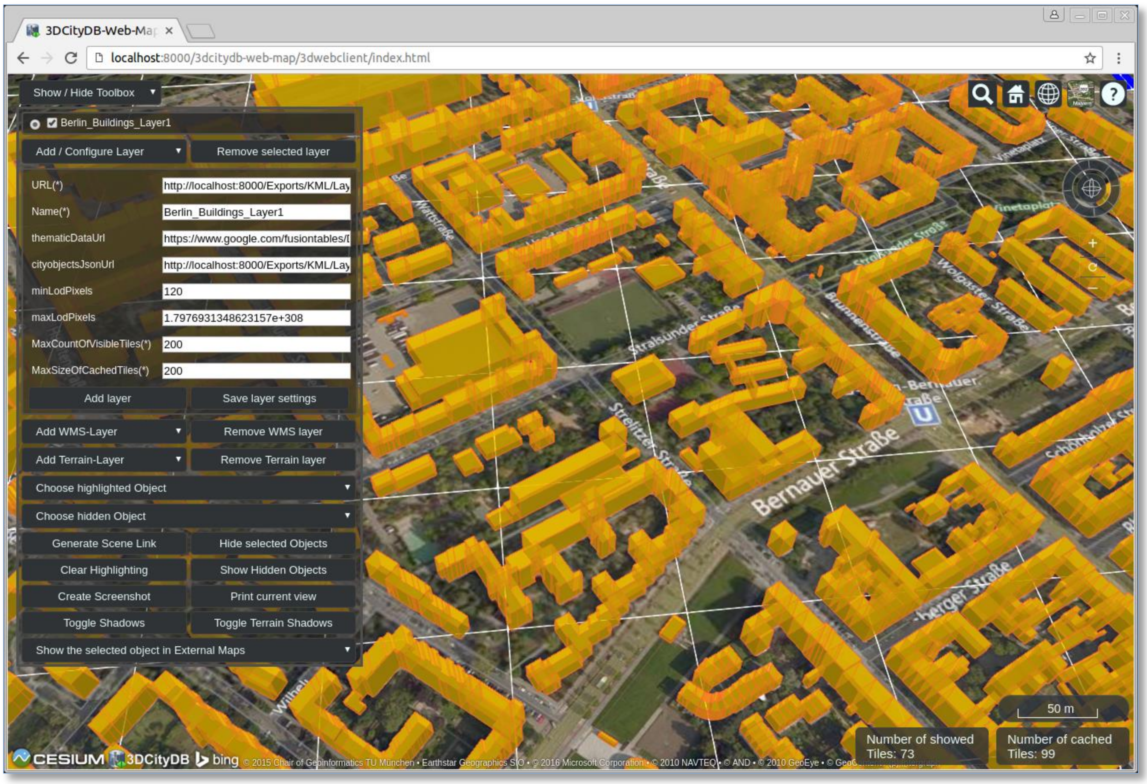
□ Click the ***Add Layer*** button.

A layer will be loaded into the web client and the corresponding layer name is also listed in the layer settings panel.


URL(*) http://localhost:8000/Exports/KML/Lay
Name(*) Berlin_Buildings_Layer1
thematicDataUrl https://www.google.com/fusiontables/
cityobjectsJsonUrl http://localhost:8000/Exports/KML/Lay
minLodPixels 120
maxLodPixels 1.7976931348623157e+308
MaxCountOfVisibleTiles(*) 200
MaxSizeOfCachedTiles(*) 200
Number of showed Tiles: 0
Number of cached Tiles: 0

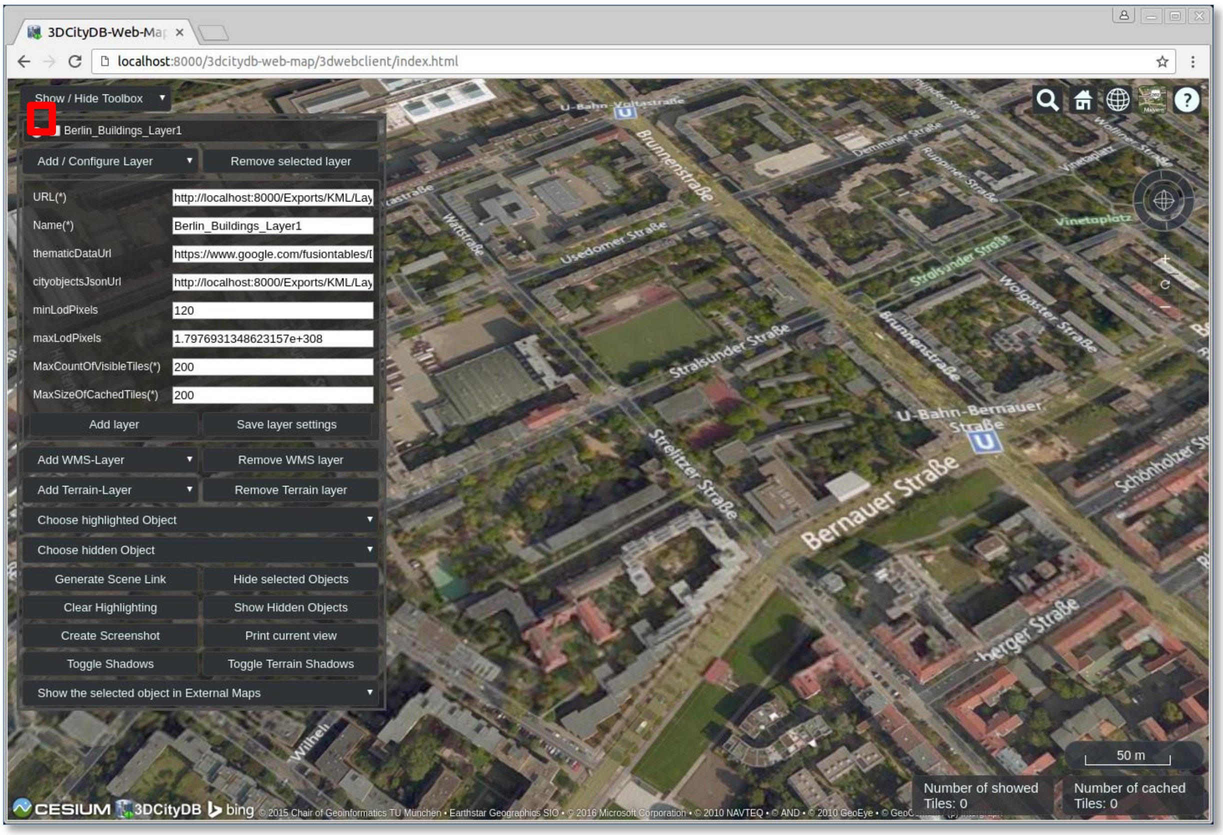
- Double-click the on the layer name (marked in the screenshot above)

The 3D map will zoom to the loaded layer (see the following screenshot)


URL(*) http://localhost:8000/Exports/KML/Lay
Name(*) Berlin_Buildings_Layer1
thematicDataUrl https://www.google.com/fusiontables/
cityobjectsJsonUrl http://localhost:8000/Exports/KML/Lay
minLodPixels 120
maxLodPixels 1.7976931348623157e+308
MaxCountOfVisibleTiles(*) 200
MaxSizeOfCachedTiles(*) 200
Number of showed Tiles: 73
Number of cached Tiles: 99

We can add more KML data layers within the web client.

□ Deactivate the checkbox (marked in the following screenshot) on the left side of the layer name label to disable and hide the loaded layer (***Berlin_Buildings_Layer1***)



URL(*) http://localhost:8000/Exports/KML/Lay

Name(*) Berlin_Buildings_Layer1

thematicDataUrl https://www.google.com/fusiontables/I

cityobjectsJsonUrl http://localhost:8000/Exports/KML/Lay

minLodPixels 120

maxLodPixels 1.7976931348623157e+308

MaxCountOfVisibleTiles(*) 200

MaxSizeOfCachedTiles(*) 200

Let's add a second data layer within the web client:

□ Fill the text field ***URL(*)*** with the following data link which references the exported KML data of Berlin (***Berlin_Buildings_Layer2***)

http://localhost:8000/Exports/KML/Layer2/Berlin_Buildings_Layer2_collada_MasterJSON.json

□ Fill the text field ***Name(*)*** with the text value "***Berlin_Buildings_Layer2***"

□ Fill the text field ***thematicDataUrl*** with the web link of the created Google Fusion Table (please refer to Exercise 5)

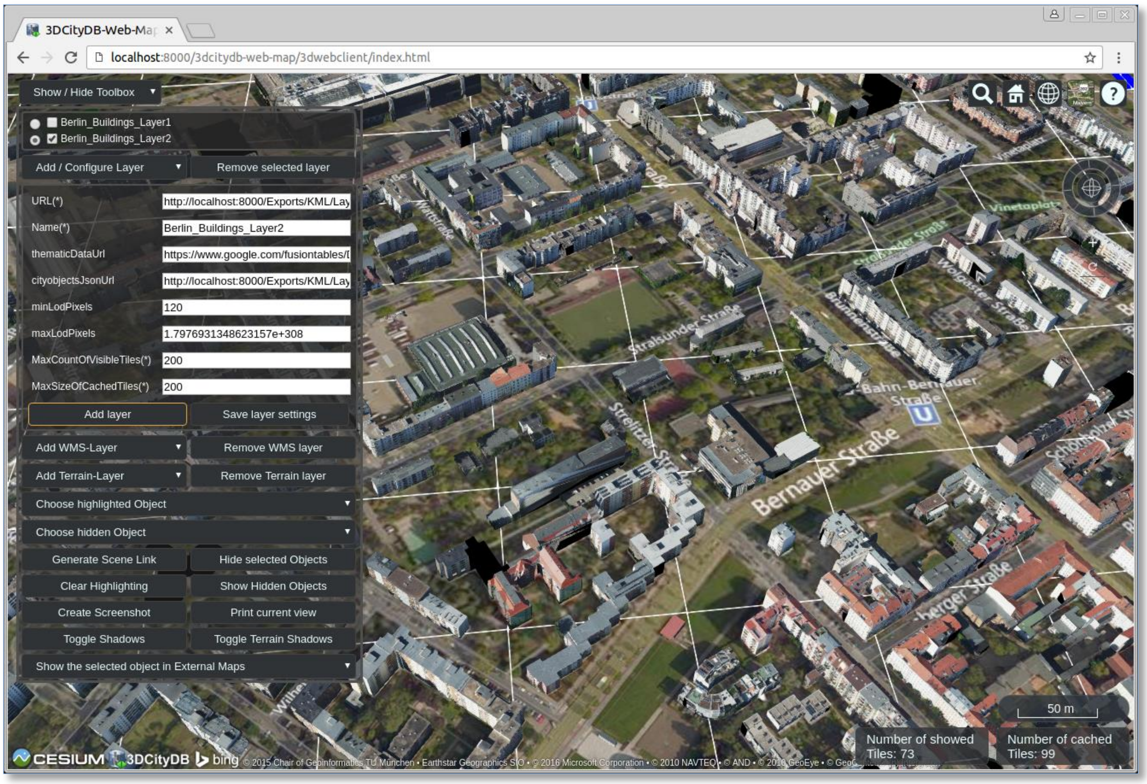
□ Fill the text field ***cityobjectsJsonUrl*** with the data link of the created json data

http://localhost:8000/Exports/KML/Layer2/Berlin_Buildings_Layer2.json

□ Keep the other text fields unchanged

□ Click the ***Add Layer*** button

A new layer ***Berlin_Buildings_Layer2*** will be loaded within the web client and the corresponding layer name is added and listed in the layer settings panel (see the screenshot below).



○ □ Berlin_Buildings_Layer1
● ☑ Berlin_Buildings_Layer2

URL(*) http://localhost:8000/Exports/KML/Lay
Name(*) Berlin_Buildings_Layer2
thematicDataUrl https://www.google.com/fusiontables/
cityobjectsJsonUrl http://localhost:8000/Exports/KML/Lay
minLodPixels 120
maxLodPixels 1.7976931348623157e+308
MaxCountOfVisibleTiles(*) 200
MaxSizeOfCachedTiles(*) 200

Number of showed Tiles: 73
Number of cached Tiles: 99

□ Deactivate the checkbox on the left side of the layer name label to disable and hide the loaded layer (***Berlin_Buildings_Layer2***)

Let's add a third data layer to the web client:

□ Fill the text field ***URL(*)*** with the following data link which references the exported KML data of Berlin (***Berlin_Buildings_Layer3***)

http://localhost:8000/Exports/KML/Layer3/Berlin_Buildings_Layer3_collada_MasterJSON.json

□ Fill the text field ***Name(*)*** with the text value "***Berlin_Buildings_Layer3***"

□ Fill the text field ***thematicDataUrl*** with the web link of the created Google Fusion Table (please refer to Exercise 5)

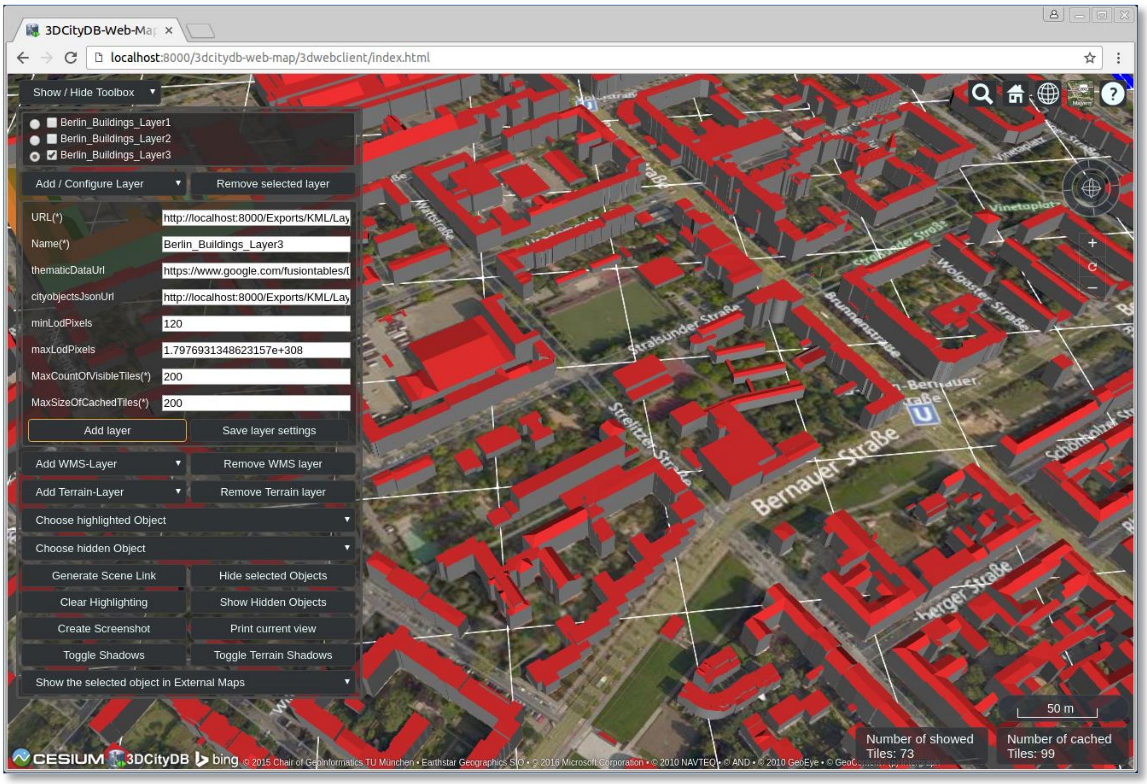
□ Fill the text field ***cityobjectsJsonUrl*** with the data link of the created json data

http://localhost:8000/Exports/KML/Layer3/Berlin_Buildings_Layer3.json

□ Keep the other text fields unchanged

□ Click the ***Add Layer*** button

A new layer ***Berlin_Buildings_Layer3*** will be loaded within the web client and the corresponding layer name is added and listed in the layer settings panel (see the screenshot below).



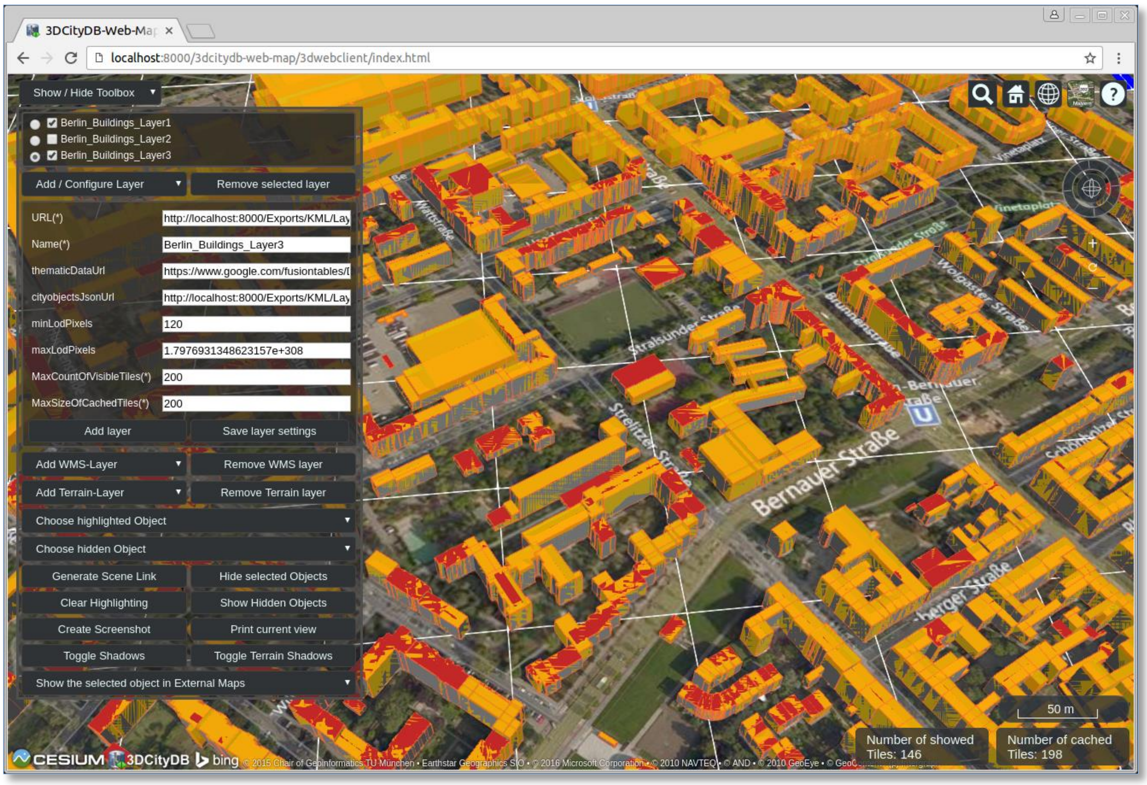
○ □ Berlin_Buildings_Layer1
○ □ Berlin_Buildings_Layer2
● ☑ Berlin_Buildings_Layer3

URL(*) http://localhost:8000/Exports/KML/Lay
Name(*) Berlin_Buildings_Layer3
thematicDataUrl https://www.google.com/fusiontables/t
cityobjectsJsonUrl http://localhost:8000/Exports/KML/Lay
minLodPixels 120
maxLodPixels 1.7976931348623157e+308
MaxCountOfVisibleTiles(*) 200
MaxSizeOfCachedTiles(*) 200

Number of showed Tiles: 73
Number of cached Tiles: 99

By now, three data layers have been loaded into the 3D web client. Let’s take a look at what will happen when two layers are displayed simultaneously.

□ Activate the checkbox ***Berlin_Buildings_Layer1*** on the left side the corresponding layer name label to make this layer visible in the web client (see the screenshot below)



○ ☑ Berlin_Buildings_Layer1
○ □ Berlin_Buildings_Layer2
● ☑ Berlin_Buildings_Layer3

URL(*) http://localhost:8000/Exports/KML/Lay
Name(*) Berlin_Buildings_Layer3
thematicDataUrl https://www.google.com/fusiontables/t
cityobjectsJsonUrl http://localhost:8000/Exports/KML/Lay
minLodPixels 120
maxLodPixels 1.7976931348623157e+308
MaxCountOfVisibleTiles(*) 200
MaxSizeOfCachedTiles(*) 200

Number of showed Tiles: 146
Number of cached Tiles: 198

In the screenshot above, you can see that two data layers (***Berlin_Buildings_Layer1*** and ***Berlin_Buildings_Layer3)*** are overlapping each other representing different level of details (LoD1 and LoD2) respectively.

In order to avoid this negative effect, the 3D web client has been designed to support the so-called *Level of Details* (LOD) concept, which is a common solution being used in 3D computer graphics and GIS for efficient streaming and rendering of heterogeneous data. According to the LOD concept, the data tiles with higher resolution will be loaded and visualized when the observer is viewing them from a short distance. When data tiles are far away from the observer, the data tiles with higher resolution will be substituted by the data tiles with lower resolution.

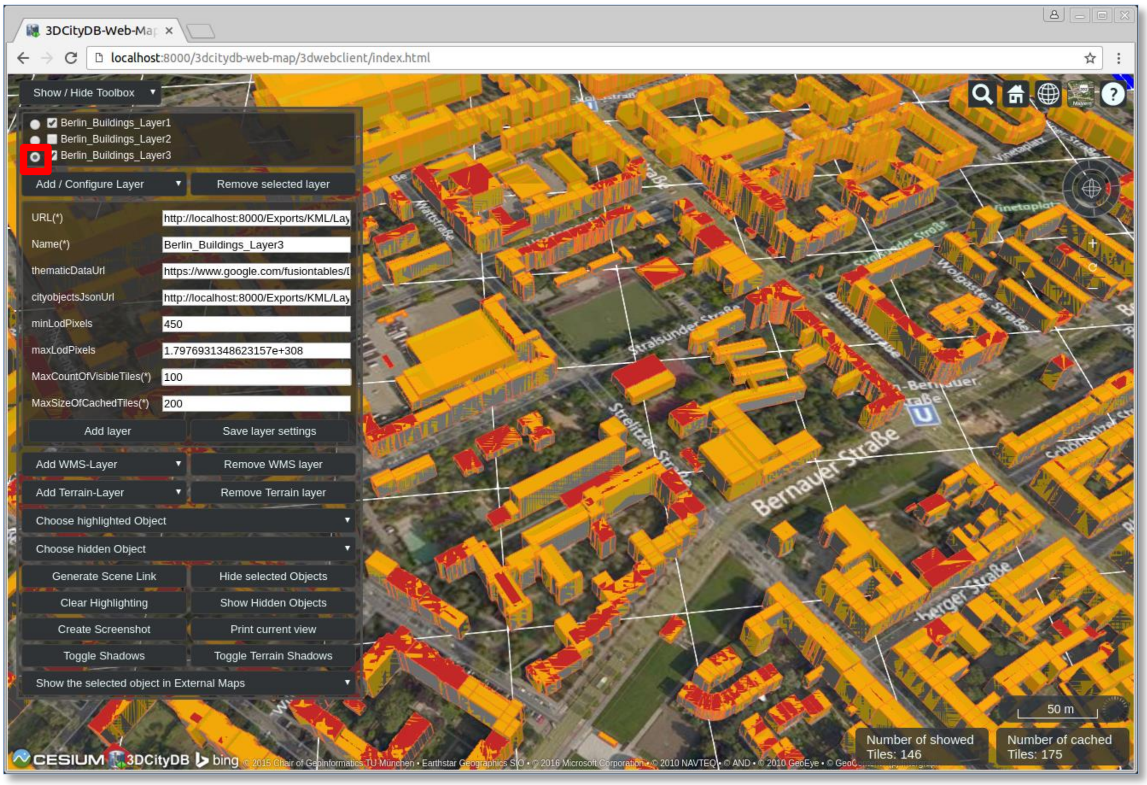
For realization of the LoD concept in the 3D webclient, each data layer has two parameters *minLodPixels* and *maxLodPixels* which are the measurement in screen pixels that represents the minimum and maximum limit of the visibility range for a data tile. When the data tile is projected onto the screen, the diagonal of its projected area on the screen must be greater than *minLodPixels* and less than *maxLodPixels* in order to be visible. Once the diagonal size of the data tile lies outside of these limits, it will be inactive and invisible.

Now, let’s try to configure these two parameters for both data layers:

□ Check the radio button of the data layer ***Berlin_Buildings_Layer3*** (marked in the screenshot below)

□ Fill the text field ***minLodPixels*** with the number value **450**

□ Click the ***Save layer settings*** button to apply the settings



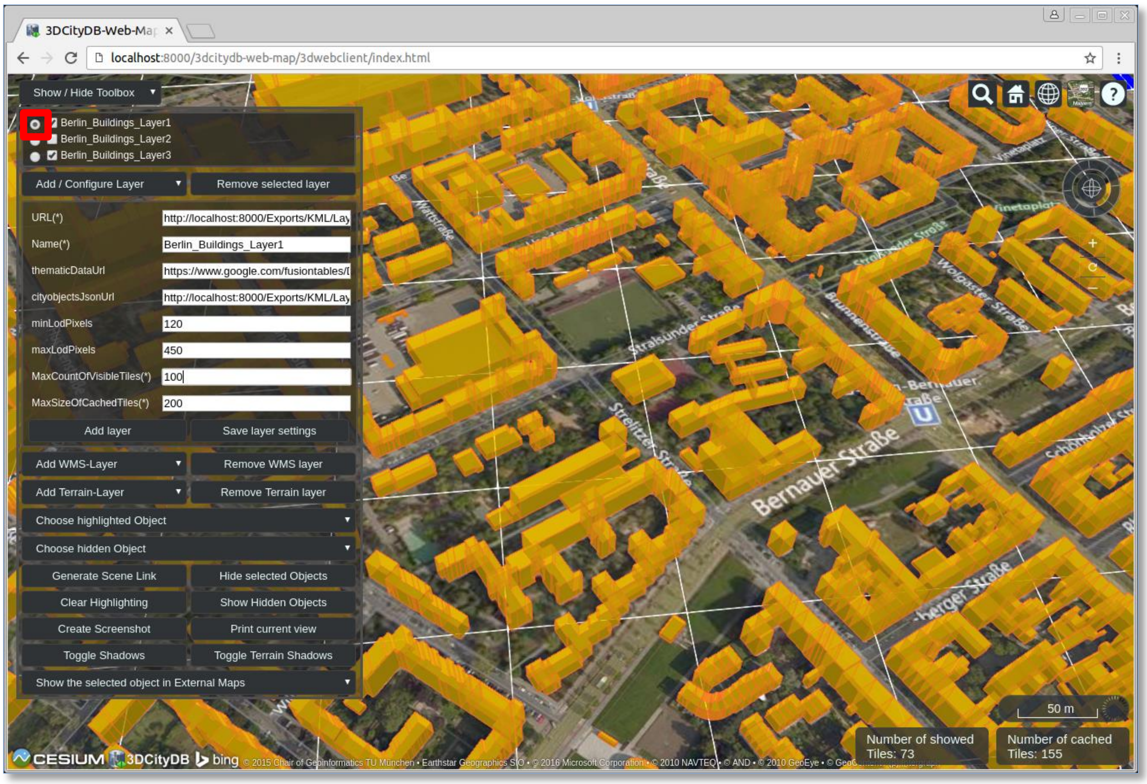
URL(*): http://localhost:8000/Exports/KML/Lay
Name(*): Berlin_Buildings_Layer3
thematicDataUrl: https://www.google.com/fusiontables/t
cityobjectsJsonUrl: http://localhost:8000/Exports/KML/Lay
minLodPixels: 450
maxLodPixels: 1.7976931348623157e+308
MaxCountOfVisibleTiles(*): 100
MaxSizeOfCachedTiles(*): 200

Number of showed Tiles: 146
Number of cached Tiles: 175

□ Check the radio button of the data layer ***Berlin_Buildings_Layer1*** (marked in the screenshot below)

□ Fill the text field ***maxLodPixels*** with the number value **450**

□ Click the ***Save layer settings*** button to apply the settings



localhost:8000/3dcitydb-web-map/3dwebclient/index.html

● ☑ Berlin_Buildings_Layer1
○ ☑ Berlin_Buildings_Layer2
○ ☑ Berlin_Buildings_Layer3

URL(*) http://localhost:8000/Exports/KML/Lay
Name(*) Berlin_Buildings_Layer1
thematicDataUrl https://www.google.com/fusiontables/I
cityobjectsJsonUrl http://localhost:8000/Exports/KML/Lay
minLodPixels 120
maxLodPixels 450
MaxCountOfVisibleTiles(*) 100
MaxSizeOfCachedTiles(*) 200

Number of showed Tiles: 73
Number of cached Tiles: 155

The *maxLodPixels* value of the data layer ***Berlin_Buildings_Layer1*** is now identical to the *minLodPixels* of the of the next data layer ***Berlin_Buildings_Layer3***. If you zoom the 3D map in and out, you may see that the two layers are seamlessly switched.

Exercise 7.3 – Integration of Web Map Service

The 3D web client also supports the ability of adding an OGC compliant Web Map Service (WMS) layer to the virtual globe.

In this exercise we’ll add some historic aerial imagery of Berlin using the web map service provided by the ***Senatsverwaltung für Stadtentwicklung und Umwelt Berlin***. URL:

http://www.stadtentwicklung.berlin.de/index.shtml

The following image shows one part of the aerial imagery.

□ Click the ***Add / Configure Layer*** button to collapse the layer settings panel

□ Click the ***Add WMS-Layer*** button to expand the WMS layer settings panel (marked in the screenshot below)

□ Fill the text field ***Name(*)*** with the text value “***Berlin_Aerialphotos_1928***”

□ Fill the text field ***iconUrl(*)*** with the following Icon-URL

https://www.youtube.com/yt/space/media/images/yt-space-icon-berlin.png

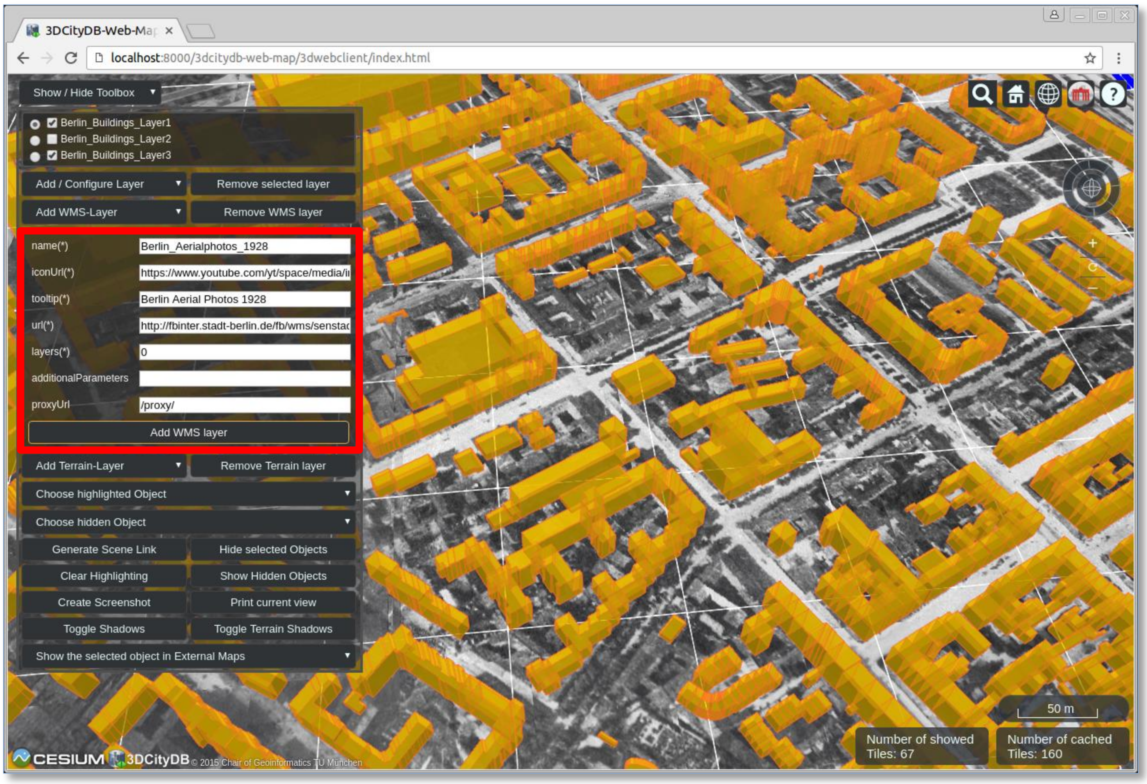
□ Fill the text field ***tooltip(*)*** with the text value “***Berlin Aerial Photos 1928***”

□ Fill the text field ***url(*)*** with the following WMS-URL

http://fbinter.stadt-berlin.de/fb/wms/senstadt/k_luftbild1928

□ Fill the text field ***layers(*)*** with the value “**0**”

□ Click the ***Add WMS Layer*** button. Then a new WMS imagery layer will be loaded as the base layer of the web client and looks like in the screenshot below



name(*) Berlin_Aerialphotos_1928

iconUrl(*) https://www.youtube.com/yt/space/media/i

tooltip(*) Berlin Aerial Photos 1928

url(*) http://fbinter.stadt-berlin.de/fb/wms/senstad

layers(*) 0

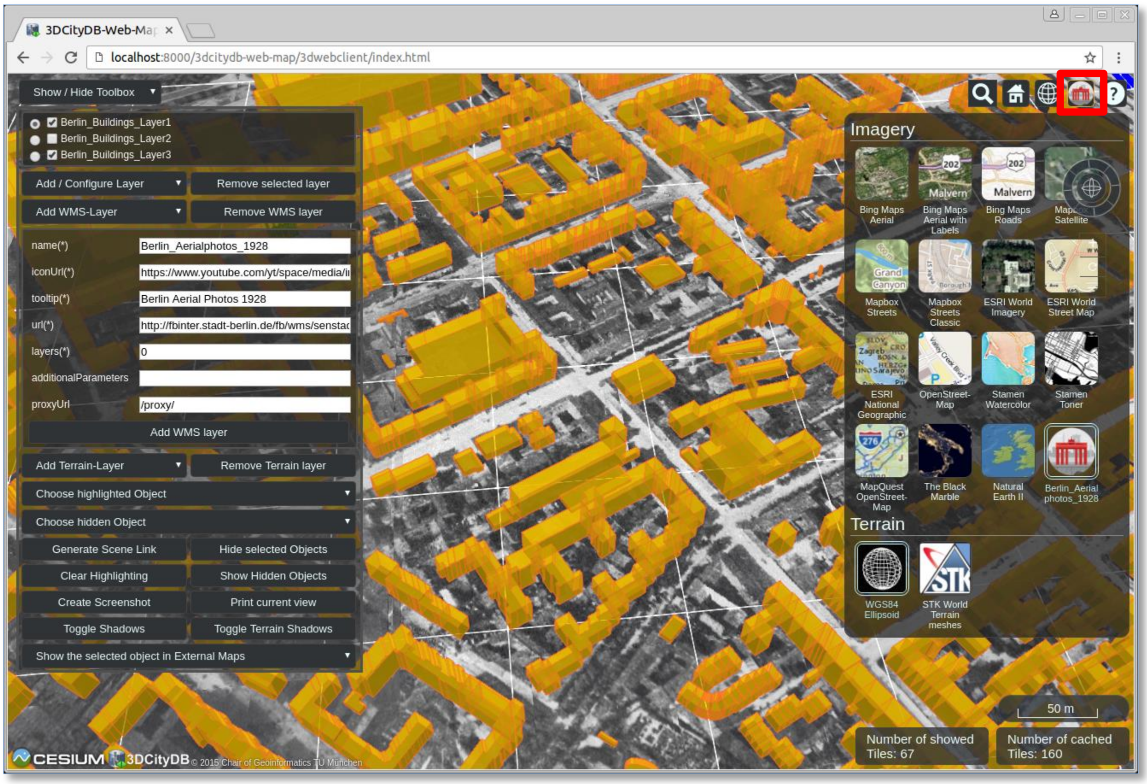
additionalParameters

proxyUrl /proxy/

Number of showed Tiles: 67

Number of cached Tiles: 160

For switching between base layers, Cesium provides a ***BaseLayerPicker*** button (see the following screenshot) that displays a panel of available imagery providers. When an imagery item is selected, the corresponding imagery layer will be loaded to replace the current base layer.



3DCityDB-Web-Map

localhost:8000/3dcitydb-web-map/3dwebclient/index.html

○ ☑ Berlin_Buildings_Layer1
○ □ Berlin_Buildings_Layer2
○ ☑ Berlin_Buildings_Layer3

name(*): Berlin_Aerialphotos_1928

iconUrl(*): https://www.youtube.com/yt/space/media/i

tooltip(*): Berlin Aerial Photos 1928

url(*): http://fbinter.stadt-berlin.de/fb/wms/senstac

layers(*): 0

additionalParameters:

proxyUrl: /proxy/

Number of showed Tiles: 67

Number of cached Tiles: 160

Exercise 7.4 – Interaction with 3D objects

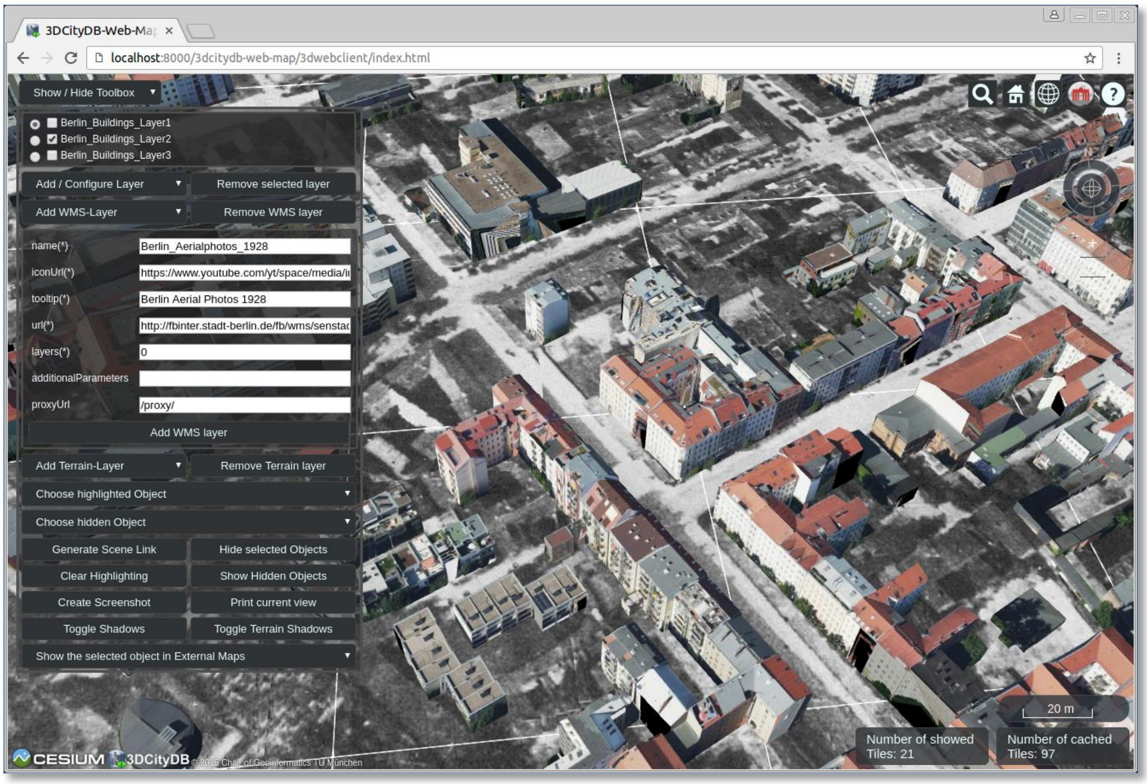
The 3DCityDB web client supports rich model interaction such as highlighting of the 3D objects on mouse over and mouse click. In addition, more than one 3D objects can be selected by Ctrl-clicking on them and can also be hidden and shown on the 3D web client interactively. Moreover, if the user clicks on a single 3D object, a table view will pop up showing the attributes and their values of the respective 3D object. The attribute information is dynamically fetched from the Google Fusion Table using the gml:id of the selected 3D object.

In this exercise we'll use the data layer ***Berlin_Buildings_Layer2*** as an example.

□ Deactivate the checkboxes of the data layers ***Berlin_Buildings_Layer1*** and ***Berlin_Buildings_Layer3***

□ Activate the checkbox of the data layer ***Berlin_Buildings_Layer2***

Now, the Berlin buildings with textures are displayed in the 3D web client (see the screenshot below)



localhost:8000/3dcitydb-web-map/3dwebclient/index.html

○ □ Berlin_Buildings_Layer1
● ☑ Berlin_Buildings_Layer2
● □ Berlin_Buildings_Layer3

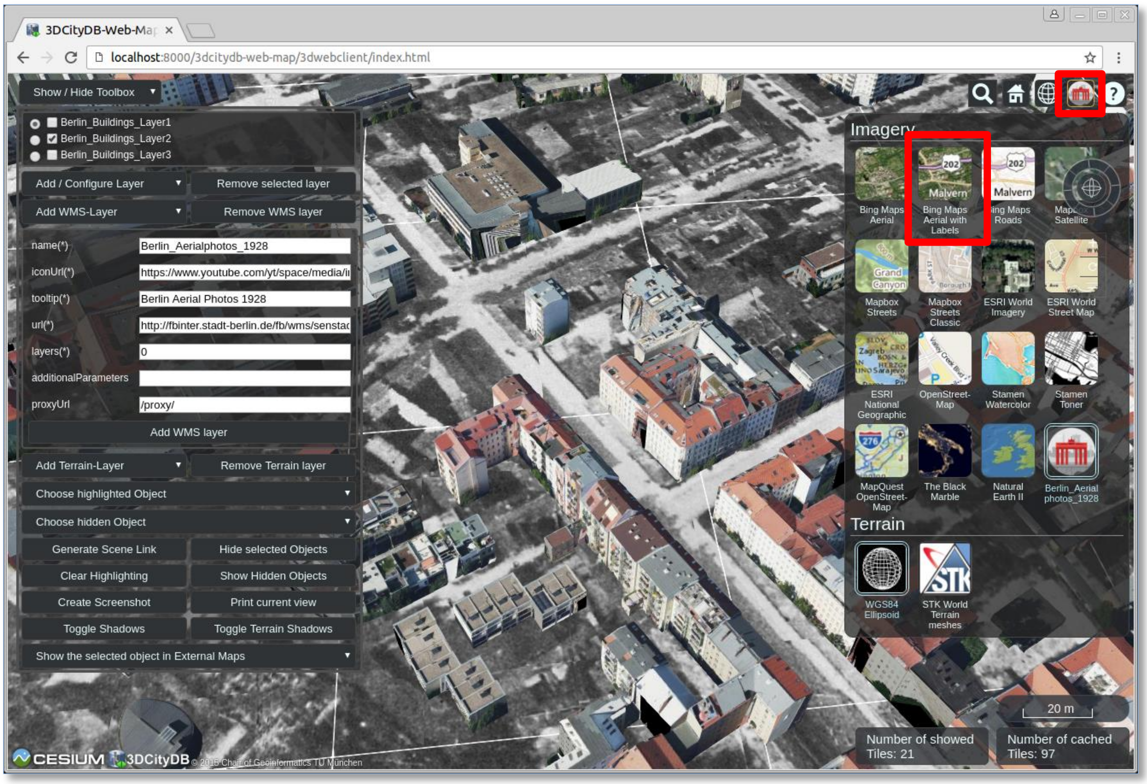
name(*) Berlin_Aerialphotos_1928
iconUrl(*) https://www.youtube.com/yt/space/media/i
tooltip(*) Berlin Aerial Photos 1928
url(*) http://fbinter.stadt-berlin.de/fb/wms/senstad
layers(*) 0
additionalParameters
proxyUrl /proxy/

Number of showed Tiles: 21
Number of cached Tiles: 97

In this exercise, we will use the default imagery layer ***Bing Maps Aeral with Labels*** as the base layer.

□ Click on the ***BaseLayerPicker*** button to expand the drop-panel (see the following screenshot)

□ Select the second item which represents the imagery layer ***Bing Maps Aeral with Labels***

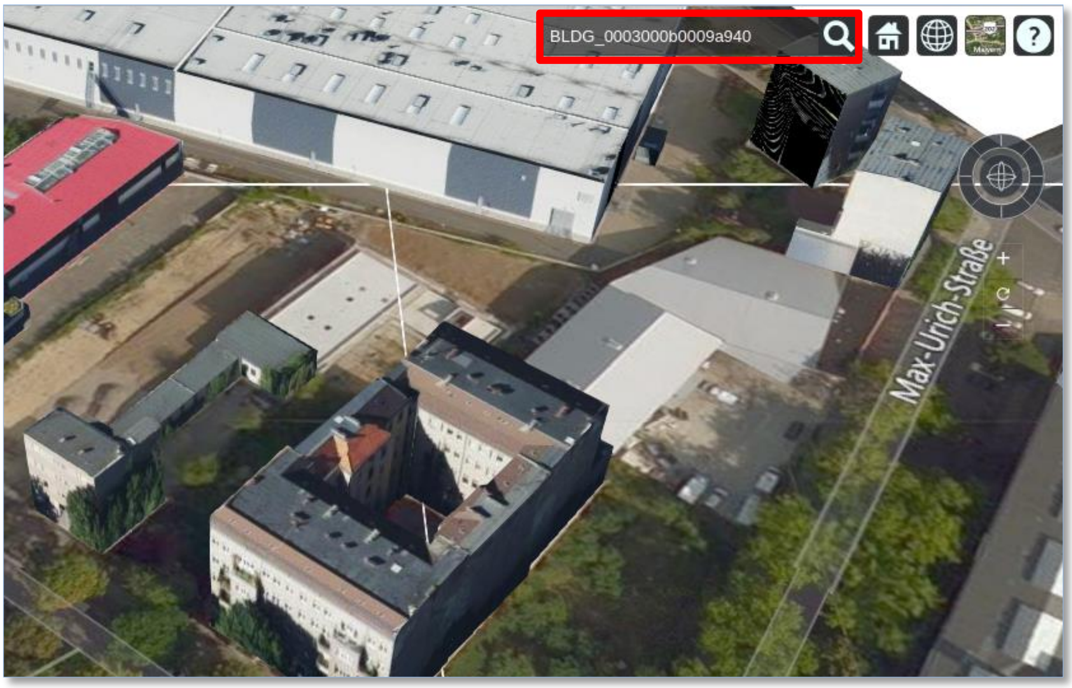


localhost:8000/3dcitydb-web-map/3dwebclient/index.html

- name(*): Berlin_Aerialphotos_1928
- iconUrl(*): https://www.youtube.com/yt/space/media/i
- tooltip(*): Berlin Aerial Photos 1928
- url(*): http://fbinter.stadt-berlin.de/fb/wms/senstad
- layers(*): 0
- additionalParameters:
- proxyUrl: /proxy/

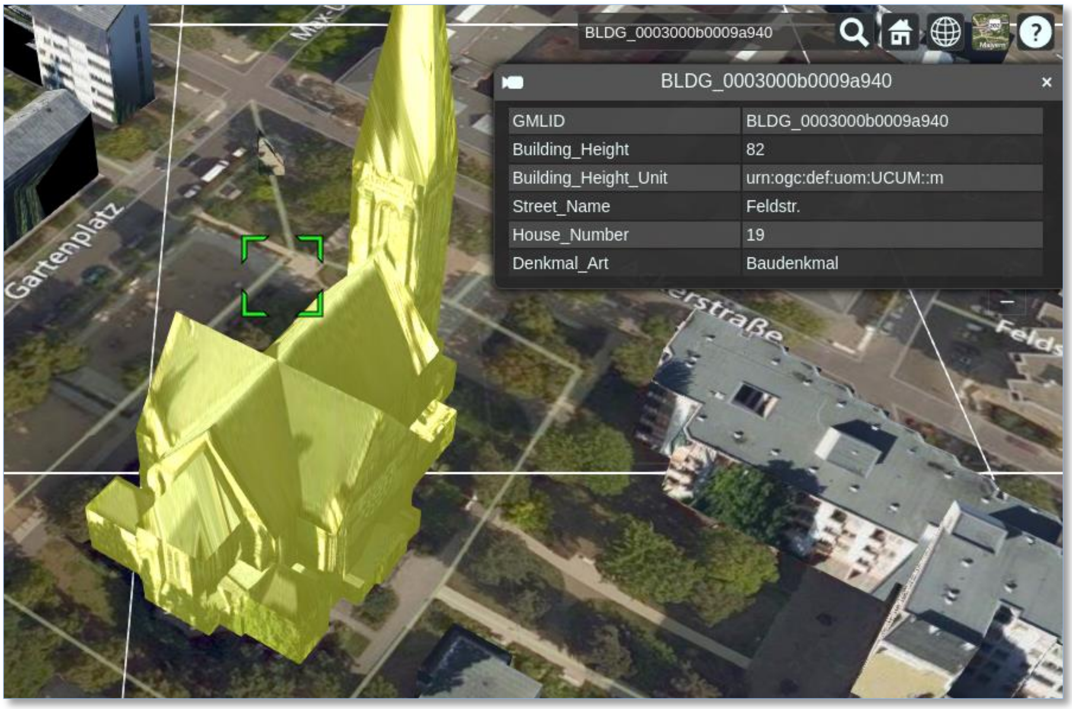
The default imagery layer ***Bing Maps Aeral with Labels*** is displayed in the 3D viewer. In the next step, let's zoom the 3D map to a building object using its gml:id value.

□ Click on the Cesium Geocoder button and fill the input field with the text value "***BLDG_0003000b0009a940***" (marked in the following screenshot)



BLDG_0003000b0009a940

□ Click on the target building object, the attribute information will be queried and displayed in a tabular form on the web client



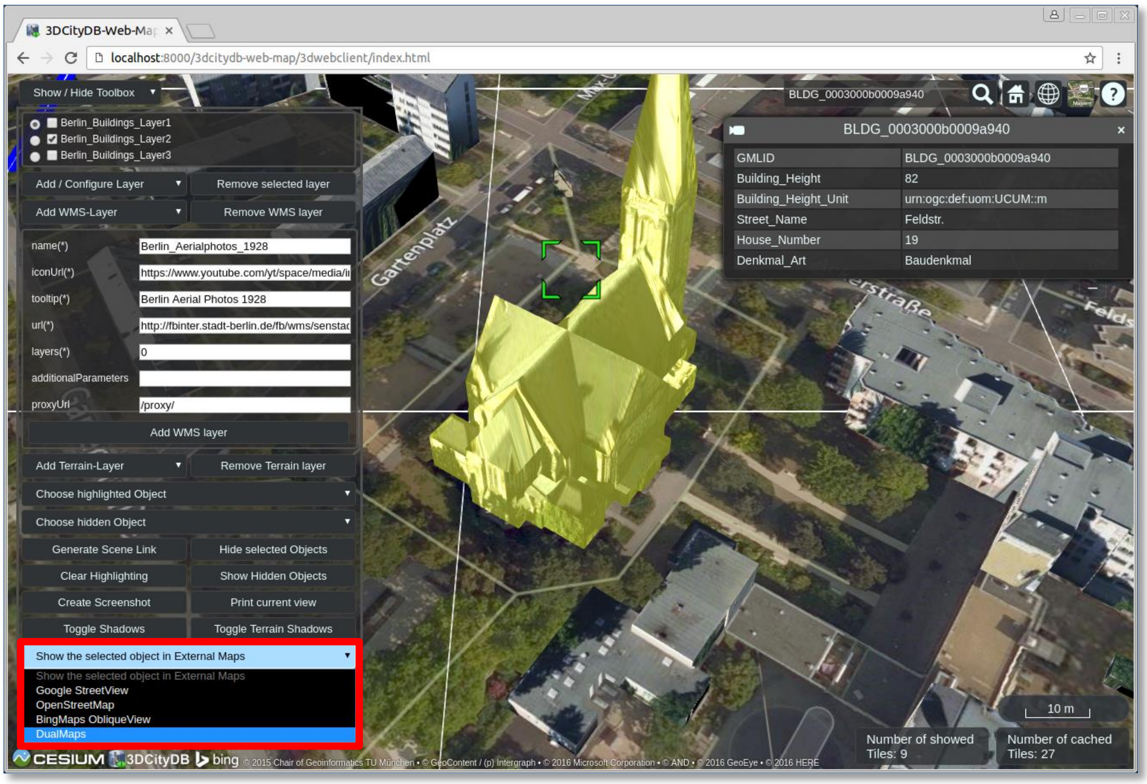
BLDG_0003000b0009a940

BLDG_0003000b0009a940

GMLID	BLDG_0003000b0009a940
Building_Height	82
Building_Height_Unit	urn:ogc:def:uom:UCUM::m
Street_Name	Feldstr.
House_Number	19
Denkmal_Art	Baudenkmal

The 3D web client also allows the user to visually inspect a selected building object using its geospatial location in third-party mapping applications such as *BingMaps*, *Google Streetview*, *OpenStreetMap* and a combined version *DualMaps*.

□ Click the dropdown list ***Show the selected object in External Maps***, and choose the option ***DualMaps*** (marked in the screenshot below)



3DCityDB-Web-Ma

localhost:8000/3dcitydb-web-map/3dwebclient/index.html

Show / Hide Toolbox

□ Berlin_Buildings_Layer1
☑ Berlin_Buildings_Layer2
□ Berlin_Buildings_Layer3

name(*)	Berlin_Aerialphotos_1928
iconUrl(*)	https://www.youtube.com/yt/space/media/i
tooltip(*)	Berlin Aerial Photos 1928
url(*)	http://fbinter.stadt-berlin.de/fb/wms/senstad
layers(*)	0
additionalParameters	
proxyUrl	/proxy/

Show the selected object in External Maps
Show the selected object in External Maps
Google StreetView
OpenStreetMap
BingMaps ObliqueView
DualMaps

BLDG_0003000b0009a940

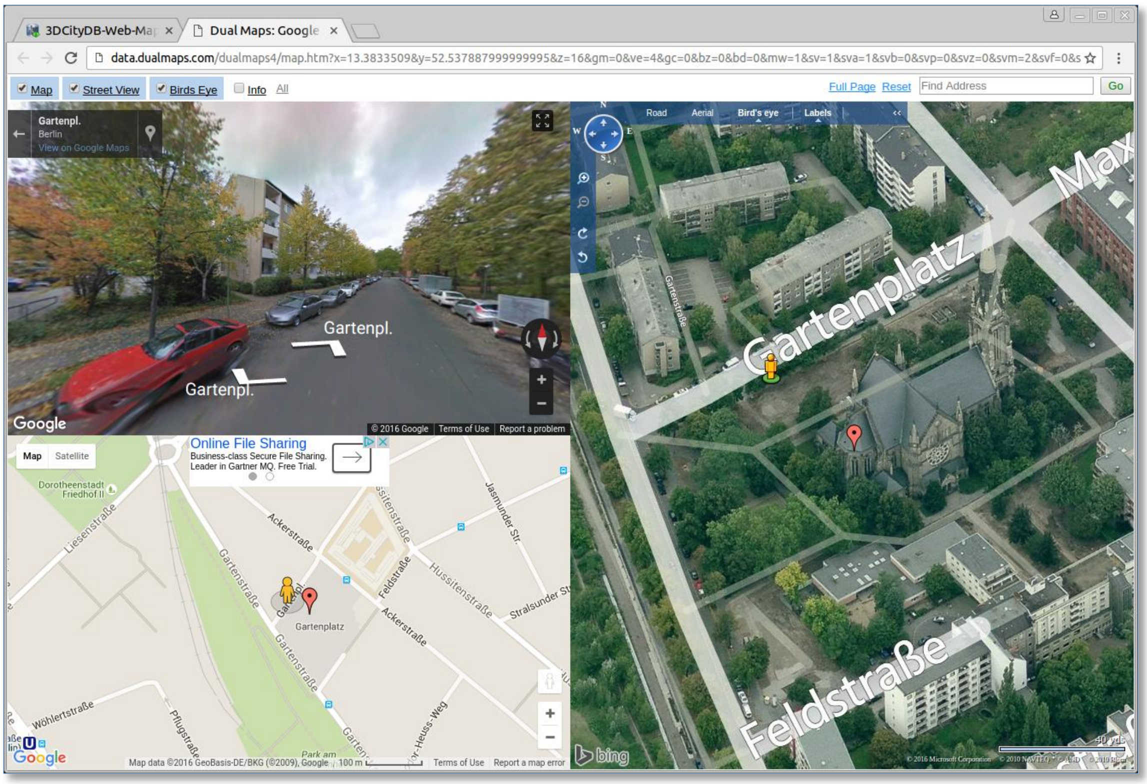
BLDG_0003000b0009a940

GMLID	BLDG_0003000b0009a940
Building_Height	82
Building_Height_Unit	urn:ogc:def:uom:UCUM::m
Street_Name	Feldstr.
House_Number	19
Denkmal_Art	Baudenkmal

Number of showed Tiles: 9

Number of cached Tiles: 27

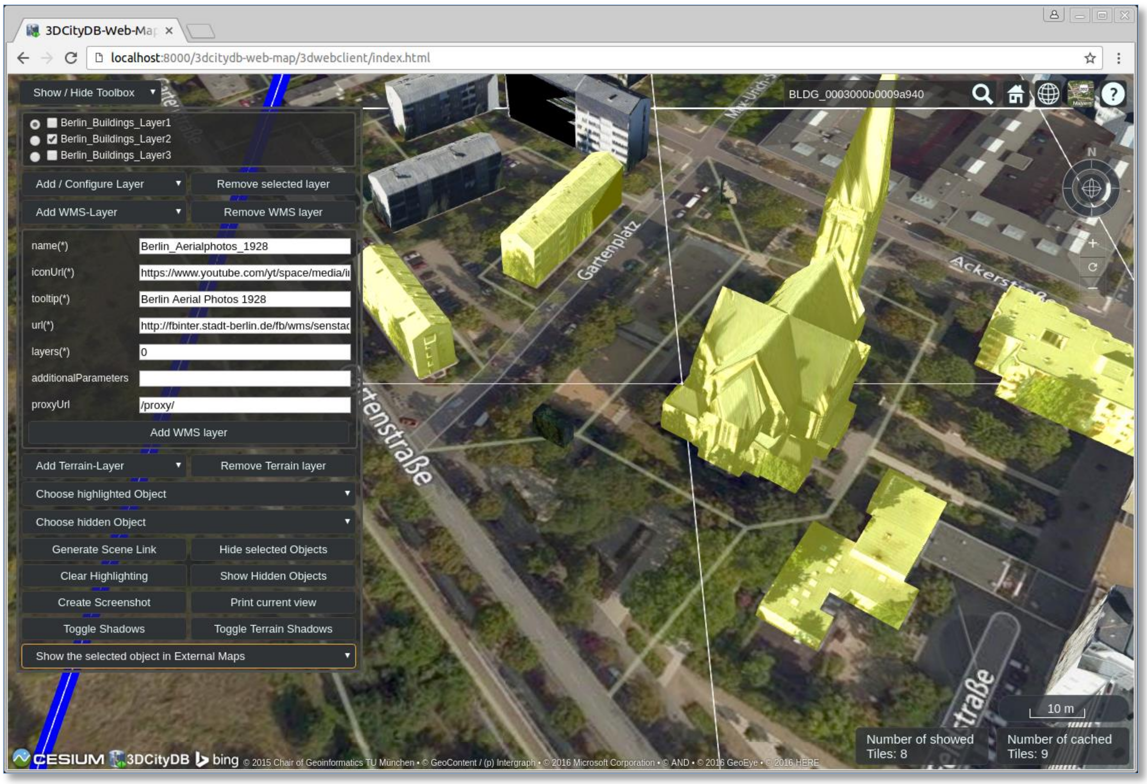
A new browser window or tab showing the clicked building object will be opened. You can explore the same buildings from different view perspectives.



data.dualmaps.com/dualmaps4/map.htm?x=13.3833509&y=52.537887999999995&z=16&gm=0&ve=4&gc=0&bz=0&bd=0&mw=1&sv=1&sva=1&svb=0&svp=0&svz=0&svm=2&svf=0&s

□ Close this ***DualMaps*** browser window or tab to go back to the web client window

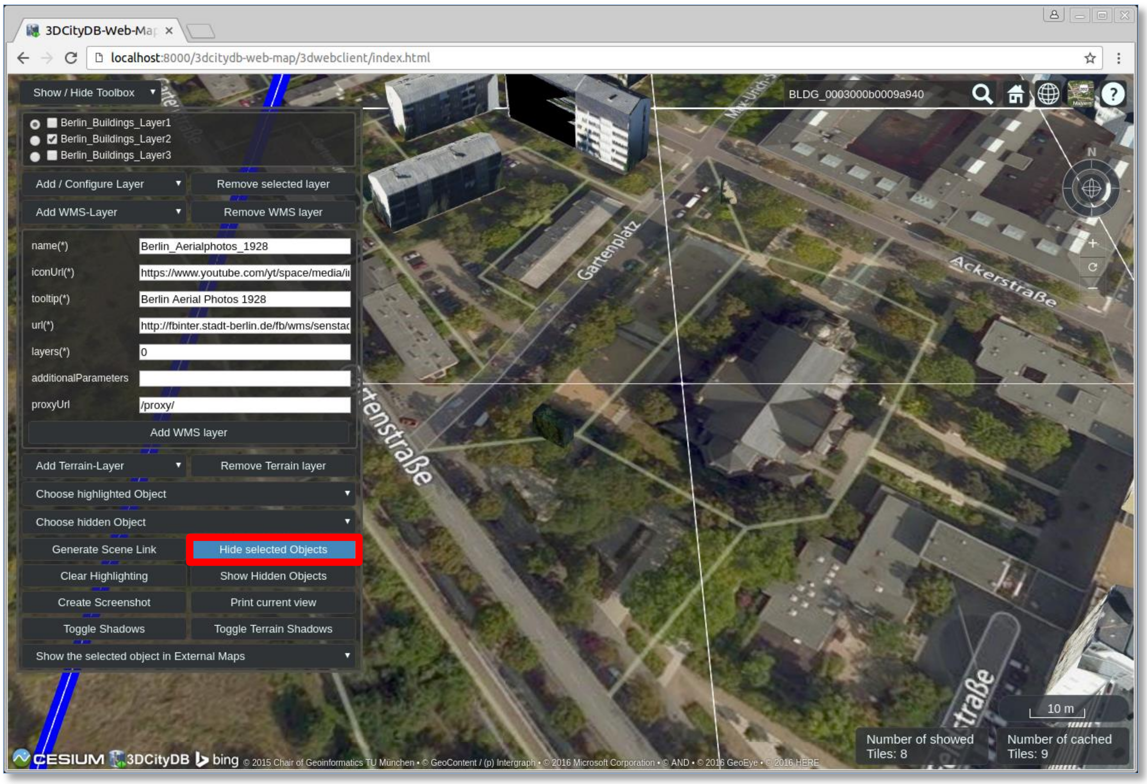
□ Select some more buildings by Ctrl-clicking on them (see the following screenshot)



localhost:8000/3dcitydb-web-map/3dwebclient/index.html

- name(*): Berlin_Aerialphotos_1928
- iconUrl(*): https://www.youtube.com/yt/space/media/i
- tooltip(*): Berlin Aerial Photos 1928
- url(*): http://fbinter.stadt-berlin.de/fb/wms/senstad
- layers(*): 0
- additionalParameters:
- proxyUrl: /proxy/

□ Click on the button ***Hide selected Objects*** (marked in the following screenshot), the selected building object will be hidden



localhost:8000/3dcitydb-web-map/3dwebclient/index.html

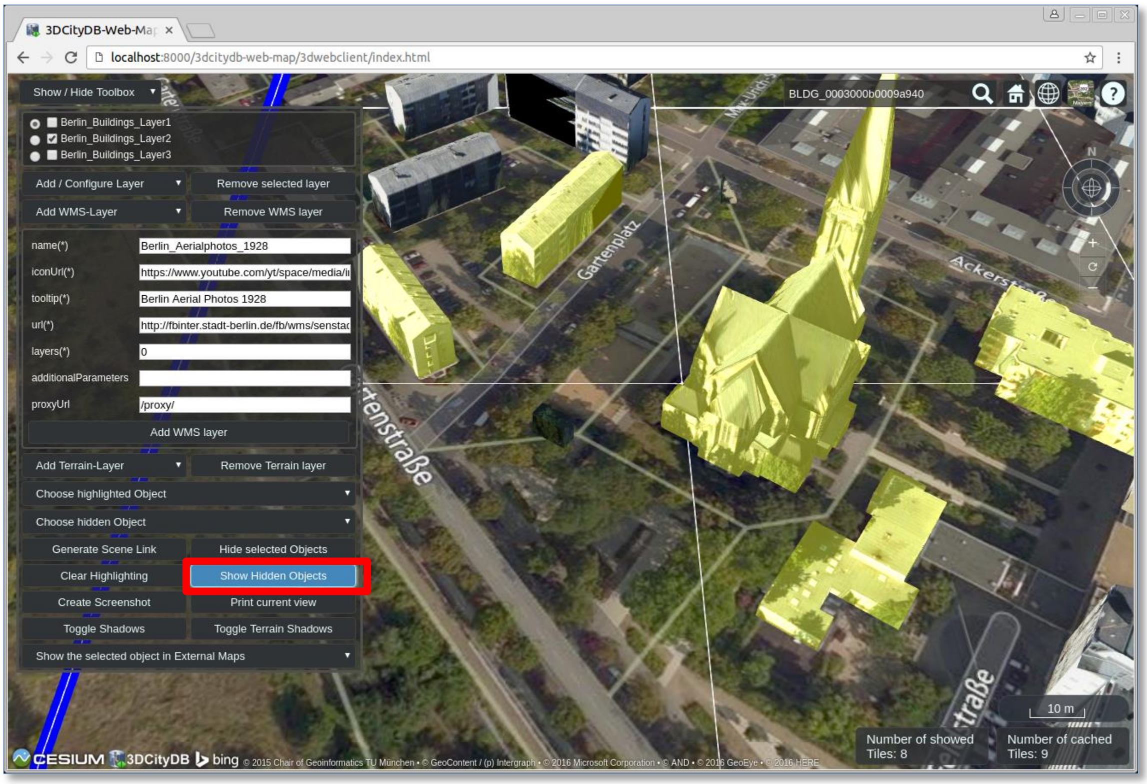
BLDG_0003000b0009a940

○ □ Berlin_Buildings_Layer1
● ☑ Berlin_Buildings_Layer2
● □ Berlin_Buildings_Layer3

name(*) Berlin_Aerialphotos_1928
iconUrl(*) https://www.youtube.com/yt/space/media/i
tooltip(*) Berlin Aerial Photos 1928
url(*) http://fbinter.stadt-berlin.de/fb/wms/senstad
layers(*) 0
additionalParameters
proxyUrl /proxy/

Number of showed Tiles: 8
Number of cached Tiles: 9

□ Click the button ***Show Hidden Object*** (marked in the following screenshot), the hidden building object will be shown again



localhost:8000/3dcitydb-web-map/3dwebclient/index.html

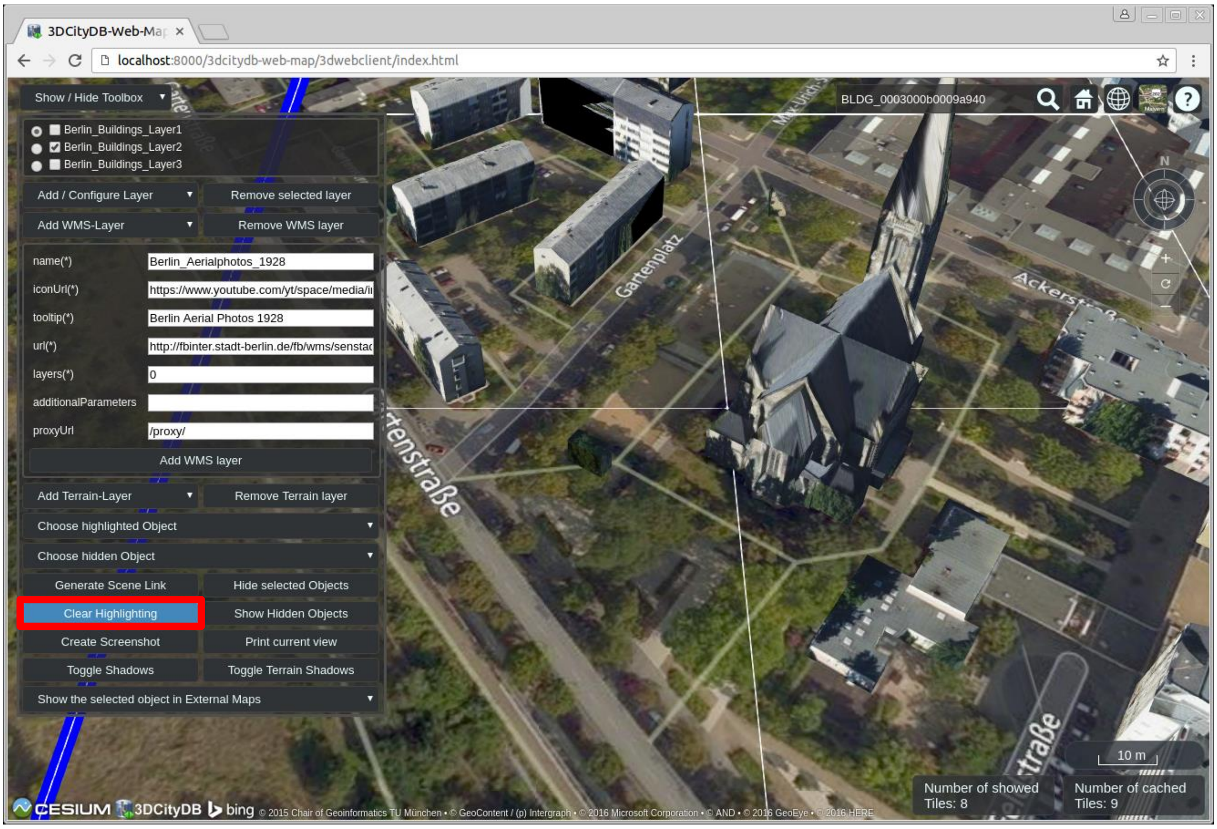
BLDG_0003000b0009a940

○ □ Berlin_Buildings_Layer1
● ☑ Berlin_Buildings_Layer2
● □ Berlin_Buildings_Layer3

name(*) Berlin_Aerialphotos_1928
iconUrl(*) https://www.youtube.com/yt/space/media/i
tooltip(*) Berlin Aerial Photos 1928
url(*) http://fbinter.stadt-berlin.de/fb/wms/senstad
layers(*) 0
additionalParameters
proxyUrl /proxy/

Number of showed Tiles: 8
Number of cached Tiles: 9

□ Click the ***Clear Highlighting*** button (marked in the following screenshot), the highlighting effect of the selected building objects will be removed



3DCityDB-Web-Ma

localhost:8000/3dcitydb-web-map/3dwebclient/index.html

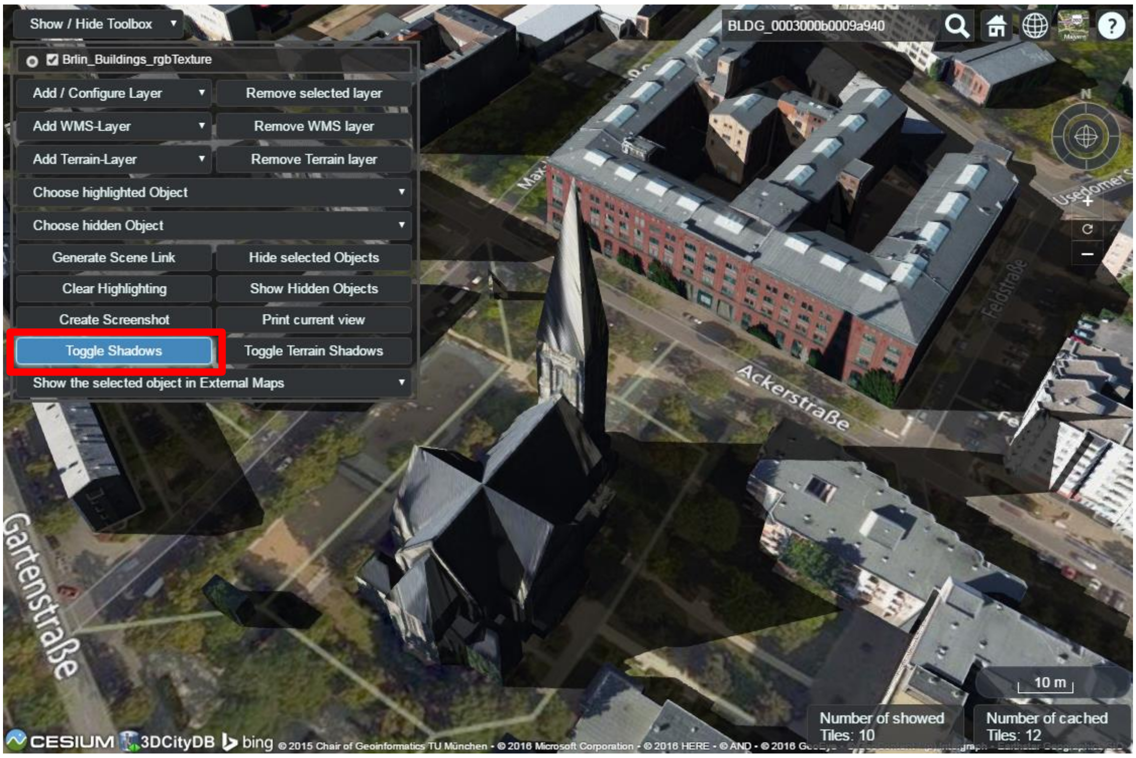
○ □ Berlin_Buildings_Layer1
● ☑ Berlin_Buildings_Layer2
● □ Berlin_Buildings_Layer3

name(*) Berlin_Aerialphotos_1928
iconUrl(*) https://www.youtube.com/yt/space/media/i
tooltip(*) Berlin Aerial Photos 1928
url(*) http://fbinter.stadt-berlin.de/fb/wms/senstad
layers(*) 0
additionalParameters
proxyUrl /proxy/

BLDG_0003000b0009a940

Number of showed Tiles: 8
Number of cached Tiles: 9

□ Shadow visualization of the 3D city models can also be activated and deactivated by clicking the *Toggle Shadows* button

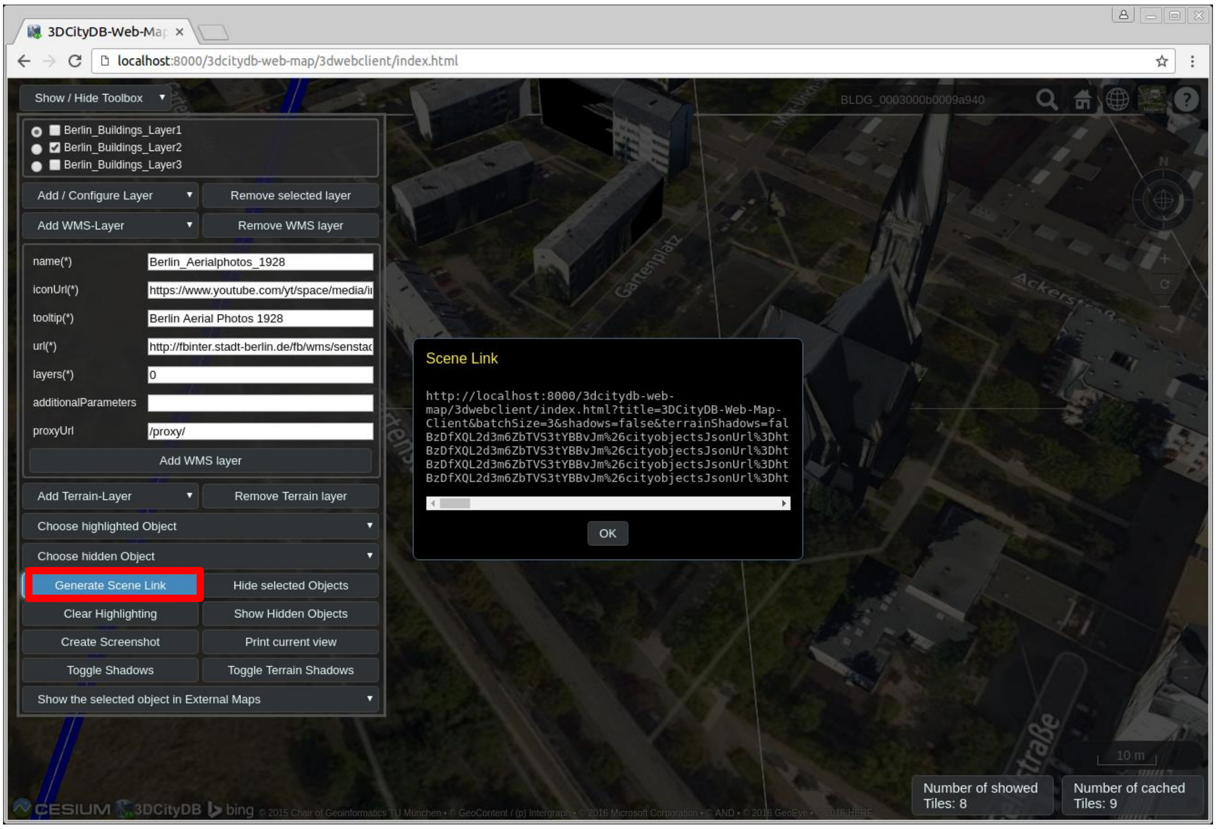


● ☑ Brlin_Buildings_rgbTexture

BLDG_0003000b0009a940

Number of showed Tiles: 10
Number of cached Tiles: 12

□ It is possible to save the current status of the 3D web client as a bookmark in your browser by clicking on the ***Generate Scene Link*** button (see the following screenshot) to create a scene link that contains the configuration information such as the parameters of the loaded layers and the current camera perspective. You may save the scene link as a bookmark in your web browser or send it to your friends via email.



name(*)	Berlin_Aerialphotos_1928
iconUrl(*)	https://www.youtube.com/yt/space/media/i
tooltip(*)	Berlin Aerial Photos 1928
url(*)	http://fbinter.stadt-berlin.de/fb/wms/senstad
layers(*)	0
additionalParameters	
proxyUrl	/proxy/

Scene Link

```
http://localhost:8000/3dcitydb-web-
map/3dwebclient/index.html?title=3DCityDB-Web-Map-
Client&batchSize=3&shadows=false&terrainShadows=fal
BzDfXQL2d3m6ZbTVS3tYBBvJm%26cityobjectsJsonUrl%3Dht
BzDfXQL2d3m6ZbTVS3tYBBvJm%26cityobjectsJsonUrl%3Dht
BzDfXQL2d3m6ZbTVS3tYBBvJm%26cityobjectsJsonUrl%3Dht
BzDfXQL2d3m6ZbTVS3tYBBvJm%26cityobjectsJsonUrl%3Dht
```

OK

Number of showed Tiles: 8

Number of cached Tiles: 9