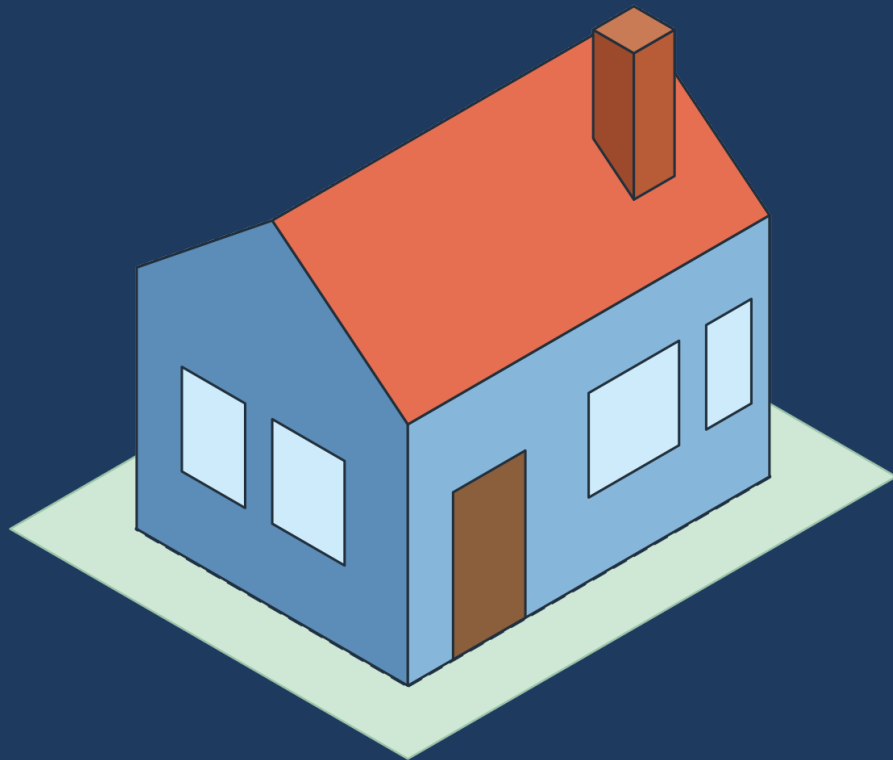


CityGML

Fundamentals

for the Curious Architect

A friendly, no-jargon guide to semantic 3D city models



Contents

1	Why Cities Need More Than Pictures	3
2	What CityGML Is (and Is Not)	4
3	The Ingredients of a City Model	5
4	Anatomy of a CityGML File	5
5	Levels of Detail	7
6	Anatomy of a Building	8
7	Beyond Buildings	9
8	Geometry Without Tears	10
9	Making It Yours: Attributes and Extensions	10
10	What Is New in CityGML 3.0	11
11	The Family of Formats	12
12	The Toolbox	13
13	Your First Hour: A Hands-On Workflow	14
14	Eight Classic Pitfalls	15
15	Quick Quiz	15
16	Cheat Sheet and Glossary	16
17	Where to Go Next	17

HOW TO USE THIS BOOK

This is a fast primer, about two hours from cover to cover. Skim each chapter, look at the pictures, and read the coloured boxes twice. Teal boxes are tips, orange boxes are traps, gold boxes are the one thing to keep, and blue boxes speak to people who think in buildings and drawings. No programming is needed, but a few short commands appear in Chapter 13 for the curious.

A WORD ABOUT THE FACTS

Statements about CityGML 3.0, CityJSON and the software tools were checked against the OGC, CityJSON.org and 3D City Database project pages. Everything else is general knowledge of the field. Standards and tools evolve, so confirm details in the current specification before you quote them in a report.

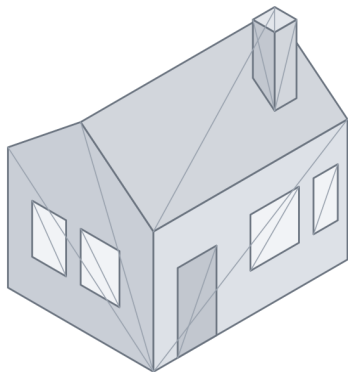
1 Why Cities Need More Than Pictures

A 3D city is only useful if the computer understands what it is looking at.

Open any web map in 3D and the buildings look convincing. Now ask it a real question: *which buildings taller than 30 metres were built before 1970?* The picture has no answer, because a typical 3D model is a **mesh**: a bag of triangles. The computer knows triangles. It does not know walls, roofs, or that a particular group of triangles is a *building at all*.

CityGML solves this by describing a city as a set of **things** with **types**, **properties** and **relationships**. A roof is a **RoofSurface** that belongs to a **Building**, which has a height, a use, and a year of construction. Once the model carries that knowledge, the computer can answer questions instead of only drawing pictures.

A plain 3D mesh

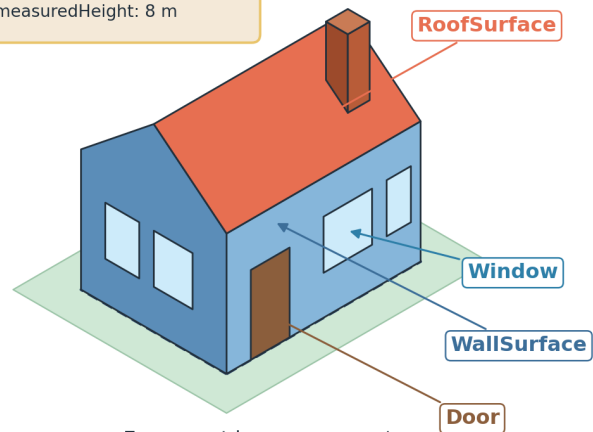


Triangles only. It looks like a house, but the computer sees no 'house'.

A semantic model (CityGML)

Building

function: residential
yearOfConstruction: 1998
measuredHeight: 8 m



Every part has a name, a type and attributes you can query.

Figure 1. Same house, two worlds: a mesh that only looks right, and a semantic model that knows what it is.

Questions a semantic model can answer

- Which roofs face south and are large enough for solar panels?
- How much facade area needs renovation in this district?
- Which buildings stand inside a flood zone, and what are they used for?
- How many residents are exposed to noise above a threshold?

FOR THE ARCHITECT Same mindset, bigger scale

Your BIM or CAD model already knows a wall from a slab. CityGML brings that same "knowing" to the scale of neighbourhoods and cities. The trade is simple: less construction detail, far more breadth.

REMEMBER Looks versus knows

A mesh **looks** like a city. A CityGML model **knows** what the city is made of.

2 What CityGML Is (and Is Not)

An open vocabulary for describing 3D city objects, not a drawing format.

CityGML is an open standard from the **Open Geospatial Consortium (OGC)** for storing and exchanging 3D city models. It defines a shared vocabulary (Building, Road, Tree, WaterBody and many more), how their shapes are described, and how attributes and appearance attach to them. "GML" stands for Geography Markup Language, the XML flavour in which CityGML has traditionally been written.

Version	Where it stands	In one line
1.0	OGC standard since August 2008	The founding idea: city objects with meaning, in several levels of detail.
2.0	Approved late 2011	Refined model with more themes, for example Bridge and Tunnel. Still found in many datasets today.
3.0	Conceptual model approved by OGC members as an official standard	Separates the model from the file format, adds time-dependent data and a rethought level-of-detail concept (Chapter 10).

What CityGML is not

- **Not only a file format.** GML is the classic encoding, but the same data model also lives in CityJSON and in databases (Chapter 11).
- **Not a rendering format.** It describes meaning and shape. Pretty visualisation usually means converting it to something lighter.
- **Not a BIM replacement.** It does not model wall layers, rebar or MEP. For construction detail, IFC remains the right tool.
- **Not a piece of software.** It is a standard that many programs read and write.

TIP Say it out loud

"City-G-M-L". When someone says "we have a CityGML model", always ask which **version** and which **level of detail**. Those two answers tell you almost everything about what you can do with it.

CHAPTER 3

3 The Ingredients of a City Model

Every city object is a recipe of five things: type, shape, properties, looks and links.

Ingredient	The question it answers	Example	Architect's analogy
Semantics	What is this?	Building, WallSurface, Road	The object type or layer in your CAD or BIM model
Geometry	What shape is it, and where?	A closed solid, or a set of polygons in 3D coordinates	The 3D model itself
Attributes	What do we know about it?	Height 8 m, built 1998, residential use	Schedule or property data
Appearance	What does it look like?	A colour or a facade photograph (texture)	Materials and textures
Relations	What is it connected to?	This window belongs to that wall, which belongs to that building	Hosting and grouping rules

Notice that **geometry is only one ingredient out of five**. This is the key mental shift. In a mesh, geometry is everything. In CityGML, geometry is the part that gets *labelled* by the others.

Every object also carries an **identifier**, written `gml:id`. Think of it as the object's licence plate. It lets one part of a file point to another, and lets other systems say "I mean exactly this building".

REMEMBER The recipe

Type + shape + properties + looks + links. If you can name the five for any object, you understand CityGML.

CHAPTER 4

4 Anatomy of a CityGML File

Container, object, parts, geometry, numbers. Once you see this ladder, every file looks familiar.

A CityGML file is a set of nested elements, and you can learn to read one in about a minute. The picture and the fragment on the next page show exactly what to look for.

How to read any file in 60 seconds

- Find `CityModel`: that is the whole file.
- Pick one `cityObjectMember`. The **element name** (Building, Road, ...) is the semantics.
- Scan the simple elements for **attributes**: names, heights, years, usage codes.
- Look for `lod1...`, `lod2...` element names. They reveal which levels of detail exist.
- Search for `srsName`. It names the coordinate reference system, the single most important line for placing the model correctly.

TIP Codes are not typos

Values like 1000 for a building function come from **code lists**, agreed vocabularies that map numbers to meanings. Organisations often publish their own lists, so check which one a dataset uses.

Container > Object > Parts > Geometry > Numbers

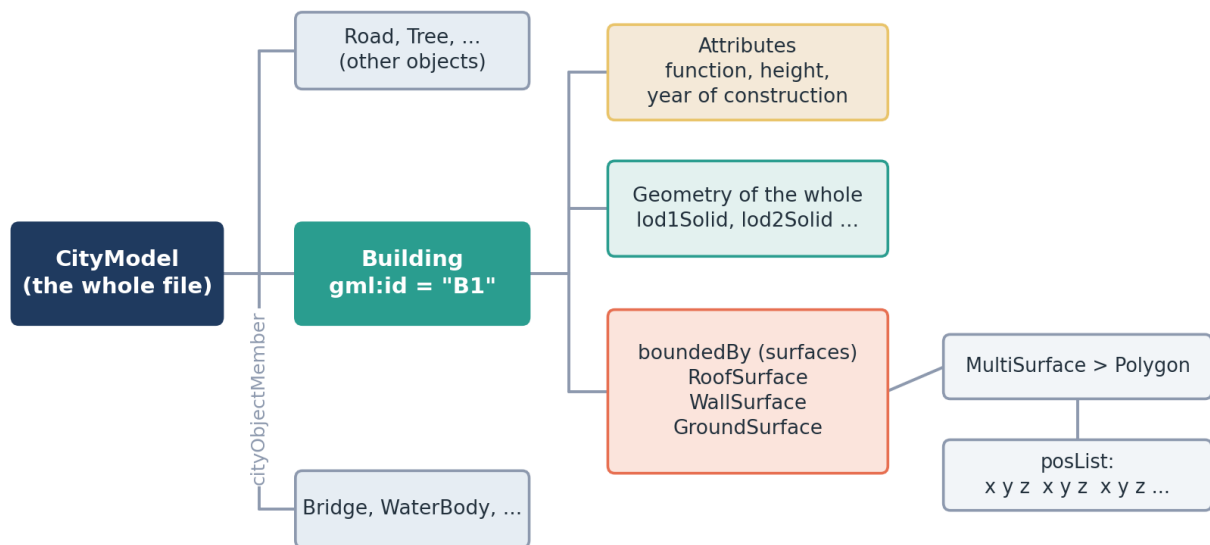


Figure 2. The nesting of a CityGML file, from the whole model down to the coordinates.

Below is a heavily trimmed fragment in the CityGML 2.0 style. Real files are far longer, but the skeleton is the same.

```

<core:CityModel>
  <core:cityObjectMember>
    <bldg:Building gml:id="B1">
      <gml:name>House 12</gml:name>
      <bldg:function>1000</bldg:function>
      <bldg:yearOfConstruction>1998</bldg:yearOfConstruction>
      <bldg:measuredHeight uom="m">8.0</bldg:measuredHeight>
      <bldg:lod2Solid> ... </bldg:lod2Solid>
      <bldg:boundedBy>
        <bldg:RoofSurface gml:id="B1_roof_1">
          <bldg:lod2MultiSurface>
            <gml:MultiSurface><gml:surfaceMember>
              <gml:Polygon><gml:exterior><gml:LinearRing>
                <gml:posList>10 20 5 14 20 5 14 24 8 10 24 8 10 20 5</gml:posList>
              </gml:LinearRing></gml:exterior></gml:Polygon>
            </gml:surfaceMember></gml:MultiSurface>
          </bldg:lod2MultiSurface>
        </bldg:RoofSurface>
      </bldg:boundedBy>
    </bldg:Building>
  </core:cityObjectMember>
</core:CityModel>
  
```

5 Levels of Detail

LoD tells you how much a model contains, not how correct it is.

A city model is rarely built at one single resolution. CityGML therefore defines **Levels of Detail (LoD)**, a ladder from a flat footprint to a finely detailed building. The same object can even carry several LoDs at once, so you can pick the one that suits your task.

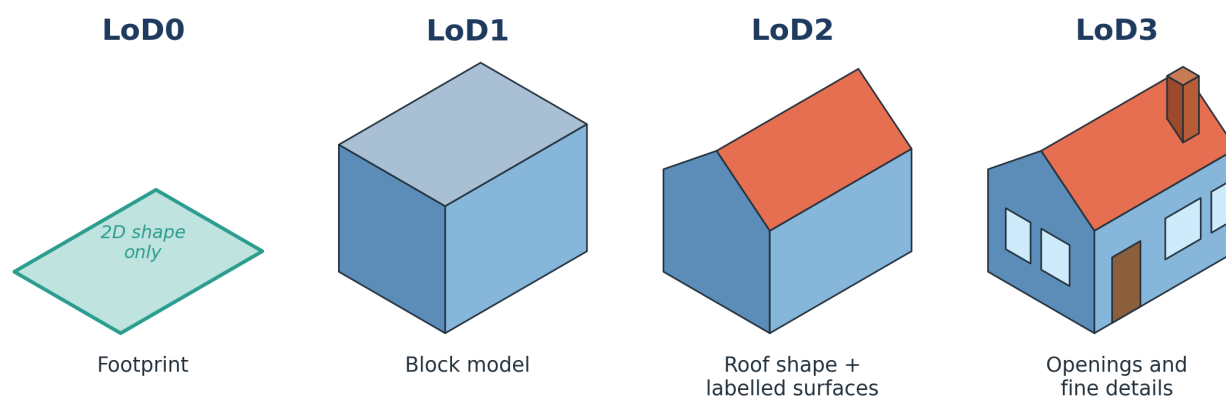


Figure 3. One house at four levels of detail.

Level	What you get	Typical uses
LoD0	Footprint or roof outline, essentially 2D	Regional planning, land-use and population mapping
LoD1	Extruded block with a flat top	Shadow, noise and density studies, volume and floor-area estimates
LoD2	Real roof shapes, and walls, roofs and ground labelled as separate surfaces	Solar potential, energy-demand estimates, flood-damage assessment
LoD3	Openings such as windows and doors, plus architectural detail	Facade analysis, detailed visualisation, visibility studies
LoD4 (2.0 only)	Interior rooms and furniture as a separate level	Indoor navigation, evacuation planning

What changed in CityGML 3.0? LoD4 was dropped. Instead, indoor features such as rooms can now appear at any of LoD0 to LoD3, where it makes sense (buildings, tunnels, bridges). A floor plan, for example, can serve as an LoD0 interior. So an exterior in LoD2 can sit beside an interior in LoD1 or LoD3.

WATCH OUT LoD is not accuracy

An LoD2 model can be wrong by several metres, and an LoD1 model built from precise survey data can be spot-on. Never assume "higher LoD means better data". Ask about the source, the date and the stated positional accuracy.

TIP Choose the lowest LoD that answers your question

Every step up costs file size, processing time and money. Noise maps rarely need windows.

6 Anatomy of a Building

CityGML names the parts of a building by the role each surface plays.

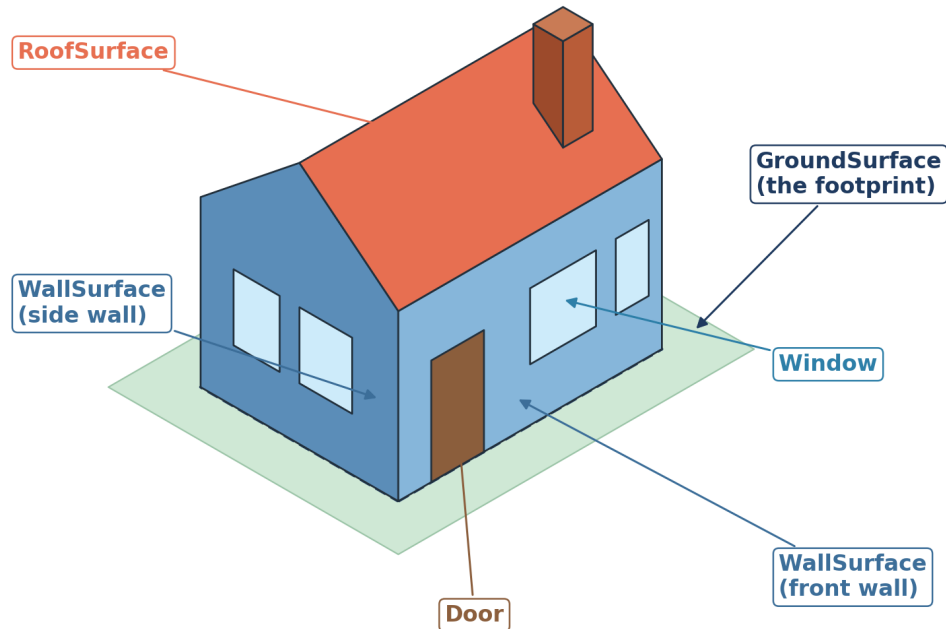


Figure 4. The labelled surfaces of a building. The dashed outline on the green plate is the footprint.

Part	What it is
RoofSurface	Any surface forming the roof, whatever its slope.
WallSurface	Exterior walls, one polygon or many.
GroundSurface	The building's footprint where it meets the ground.
ClosureSurface	A virtual lid that closes an open volume so it stays a valid solid, for example an open carport.
Window, Door	Openings that sit inside a wall surface. They appear from LoD3.
FloorSurface, CeilingSurface, InteriorWallSurface	The boundaries of interior rooms.
OuterFloorSurface, OuterCeilingSurface	Horizontal outer surfaces such as the underside of an overhang or a balcony floor.
BuildingPart	A section of a larger building, like a tower on a podium, that can carry its own geometry and attributes.
BuildingInstallation	Attached elements such as balconies, dormers, chimneys or stairs.
Room	An interior space bounded by interior surfaces (indoor models).

FOR THE ARCHITECT Roles, not layers

CityGML classifies surfaces by *role* (this one is the roof), not by construction build-up. You will not find wall thicknesses, material layers or reinforcement here. If you need those, you are in IFC territory (Chapter 11). CityGML 3.0 also gathers these surface types into a shared Construction module, so bridges and tunnels reuse the same ideas.

CHAPTER 7

7 Beyond Buildings

A city is more than its buildings. CityGML splits the world into themed modules.

Module	What lives there
Building	Buildings, building parts, installations, rooms and furniture.
Transportation	Roads, tracks, railways and squares. In 3.0 a redesigned traffic-space model supports multi-modal networks, including waterways and pedestrian areas, at several granularities.
Relief	Terrain, as triangulated surfaces, grids, breaklines or mass points.
Vegetation	Single trees and areas of plant cover.
WaterBody	Rivers, lakes and canals.
LandUse	Areas with a defined use, such as zoning or parks.
CityFurniture	Street lamps, benches, signs, bollards.
Bridge, Tunnel	Their structures, with exteriors and, where useful, interiors.
CityObjectGroup	Logical groups, for example all the buildings of one campus.
Generics	A safety valve for things the standard does not cover.
Appearance	Colours and textures that can be attached to surfaces.

New arrivals in CityGML 3.0

- **Point cloud:** an object can be represented by laser-scan or photogrammetry points.
- **Dynamizer:** a way to attach time-dependent values, such as sensor or IoT readings, to a property so a city model can have a fourth dimension.
- **Versioning:** support for tracking how objects change over time.
- **Construction:** a shared module for the building-like objects.

TIP Do not memorise the list

Learn Building well, then look up other modules when a project needs them. They all follow the same ladder from Chapter 4.

CHAPTER 8

8 Geometry Without Tears

A solid is just a watertight bag of polygons.

CityGML borrows its geometry rules from the international standard **ISO 19107**. You do not need the maths, only the ladder:

Building block	Plain-English meaning
Point, LineString	A location, or a chain of locations.
Polygon	A flat surface with an outer ring and optional holes. All its corners must lie on one plane.
MultiSurface	A loose collection of polygons. Fine for a roof, with no promise of being closed.
Solid	A closed shell of polygons enclosing a volume. No gaps, no overlaps, normals pointing outward.

A wall shared by two buildings does not need to be stored twice. CityGML lets one polygon be **referenced** from several places by its `gml:id`, which saves space and keeps neighbours consistent.

Typical geometry problems

- Polygon corners that are not quite on one plane (even by a few millimetres).
- Tiny gaps between walls and roof, so the "solid" leaks.
- Repeated or almost-repeated vertices; faces that touch themselves.
- Faces oriented the wrong way, so inside and outside swap.

WATCH OUT Valid XML is not valid geometry

A file can pass every XML check and still contain broken solids. Analyses such as volume, shadows or energy simulation quietly give wrong answers on invalid geometry. Run a geometry validator such as **val3dity** before you trust the numbers.

CHAPTER 9

9 Making It Yours: Attributes and Extensions

The standard cannot know every need, so it offers two ways to add your own data.

1. Standard and generic attributes

Buildings come with built-in attributes such as `measuredHeight`, `yearOfConstruction`, `roofType` and `storeysAboveGround`. For anything else you can attach **generic attributes**: simple name and value pairs (text, integer or decimal). They are quick, flexible and need no schema, but their meaning lives only in your own documentation.

You can also store a pointer to the same object in another system, for example a cadastre or BIM record, using an **external reference**.

2. Application Domain Extensions (ADEs)

An **ADE** is a formal, schema-based extension that adds new feature types and properties to CityGML for a whole domain. Well-known examples cover energy simulation and utility networks. An ADE keeps your extra data structured and shareable between organisations that adopt it.

You need...	Use
A one-off tag, like "survey batch 7"	A generic attribute
A link to another database	An external reference
A structured, repeatable domain model shared by many parties	An ADE

WATCH OUT ADEs do not always travel

A tool that does not know your ADE may ignore or drop that content during conversion. Test a round trip, and keep the ADE schema with the data.

CHAPTER 10

10 What Is New in CityGML 3.0

Same idea, a cleaner foundation, and a model that can handle time.

- **Model and encoding are separated.** The conceptual model can be implemented in many technologies beyond GML, including JSON and database schemas.
- **A tighter core.** The model revolves around two ideas: **space** (something that occupies room, such as a building or a room) and **space boundary** (the surfaces bounding it). This strengthens meaning and interoperability.
- **Time-dependent data.** Changes and values coming from IoT devices, sensors and simulations can be represented, enabling 4D city models and history.
- **Rethought levels of detail.** LoD0 to LoD3 remain, LoD4 is gone, and interiors can appear at any level, with floor plans as LoD0.
- **Better traffic modelling** for roads, railways, waterways and pedestrian areas.
- **A point-cloud module** for scan-based representations.
- **Nothing is thrown away.** Versions 1.0 and 2.0 are not deprecated, and existing data can be upgraded.

Which version should I use?

Your situation	Sensible choice
You must work with existing datasets and the widest range of tools	CityGML 2.0 is still the safest common ground
You are starting fresh and need time series, indoor spaces or richer traffic data	CityGML 3.0 data model
You mainly build web tools or scripts and want small, simple files	CityJSON (next chapter)

TIP Check the tools first

Support for 3.0 is growing but uneven. Before you commit, read the release notes of the tools in your pipeline to see which versions they actually read and write.

11 The Family of Formats

One data model, several ways to write it down.

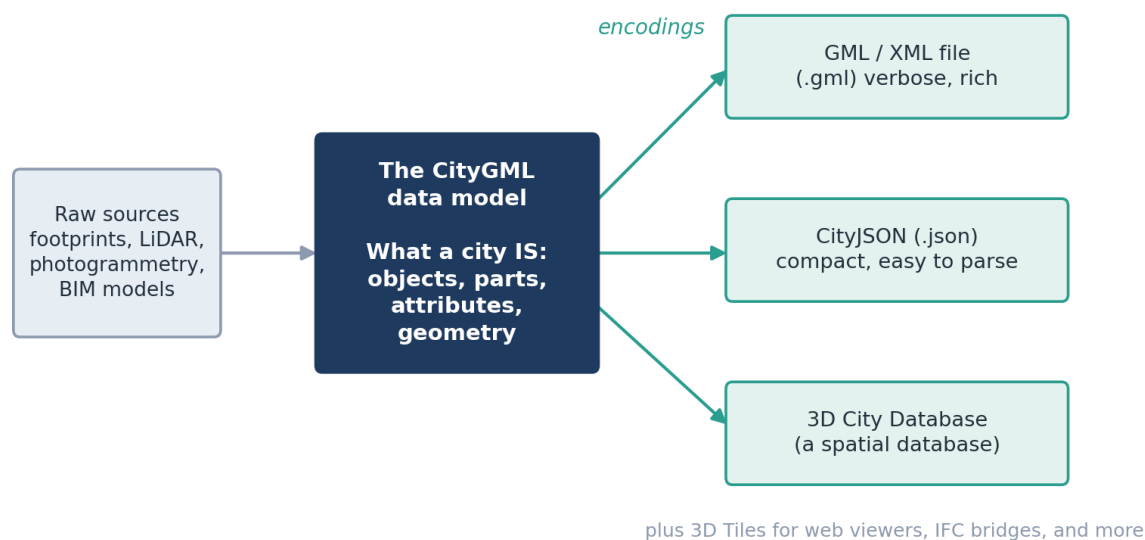


Figure 5. The data model sits at the centre. Files and databases are different containers for it.

CityJSON is a JSON-based encoding that implements a subset of the CityGML data model (version 3.0). It is an OGC Community Standard, currently at version 2.0.0. Its authors report that using JSON instead of GML compresses files by a factor of about six while greatly simplifying their structure, which is why developers like it.

Format	Strength	Typical use
CityGML (GML/XML)	Richest and most extensible, full semantics, ADEs	Official data exchange, archives, authoritative datasets
CityJSON	Same concepts, compact and easy to parse	Web apps, scripting, quick analysis
3D City Database	Stores many models, queryable with SQL, multi-user	City-wide platforms, digital twins
IFC	Deep construction detail, building scale	Design, construction, facility management
OBJ, glTF, 3D Tiles	Fast to display, light	Visualisation; little or no semantics

REMEMBER Neighbours, not rivals

IFC describes *one building* in detail. CityGML describes *a whole city* in moderate detail. Moving data from BIM to a city model means deciding what to keep and how to georeference it.

12 The Toolbox

You do not need to buy anything to start. Most of the essentials are free and open.

Job	Tools you will meet
Viewing	ninja (browser viewer), azul (macOS), QGIS plugin, ParaView reader, xeokit, FME
Converting	citygml-tools (CityGML to and from CityJSON), cjio (Python command line), tyler (builds 3D Tiles from CityJSON)
Validating	val3dity (geometry rules of ISO 19107), cjval (CityJSON validator)
Storing	3D City Database with citydb-tool , a command-line tool to import, export and manage data
Programming	citygml4j (Java library); readers and writers for other languages exist
Design tools	Blender, Rhino and Grasshopper, and Revit plugins for CityJSON

3D City Database v5 is a redesign that uses more generic mapping rules, which sharply reduces the number of database tables, and it maps geometry directly to native database types, so GIS software can connect to it more easily. The **citydb-tool** command line needs Java 17 or newer and offers commands such as **import**, **export**, **delete** and **info**. Docker images are available.

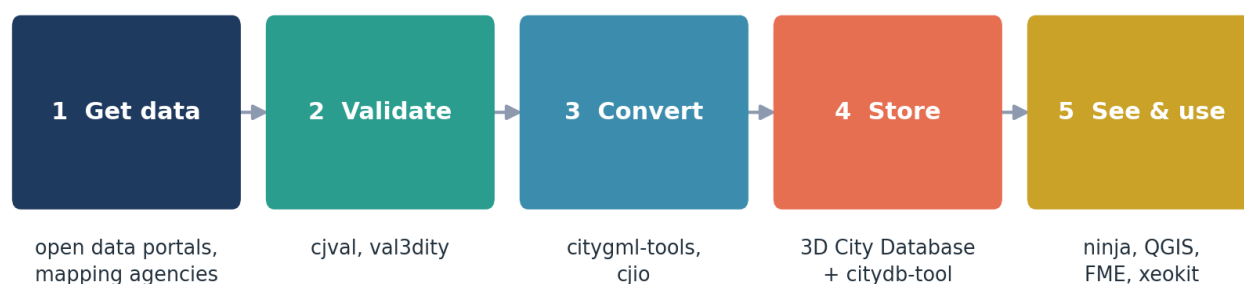


Figure 6. A typical workflow, and a tool to reach for at each step.

13 Your First Hour: A Hands-On Workflow

Five small steps from a downloaded file to a question answered.

Step 1. Get a sample

Many cities and national mapping agencies publish open 3D city models, and the CityJSON website lists open datasets. Start with a small tile of a few hundred buildings.

Step 2. Peek inside

Open the file in a text editor. Find the `srsName`, one building, and its attributes. You now know more than most people who have only seen the picture.

Step 3. Validate

```
# geometry rules (ISO 19107)
val3dity mymodel.city.json
# or paste the file into the online CityJSON validator (cjval)
```

Step 4. Convert and inspect

```
# CityGML to CityJSON
citygml-tools to-cityjson mymodel.gml
# summary of what is inside
cjio mymodel.city.json info
```

Step 5. View, then store

Drag the CityJSON file into the **ninja** viewer in your browser, click a building, and read its attributes. When you want a shared, queryable copy, load it into a 3D City Database:

```
citydb import citygml -H localhost -d citydb -u myuser -p mypassword mymodel.gml
# option names can differ between versions: run citydb import citygml --help
```

WATCH OUT Commands drift

Options and sub-commands change between releases. Treat the lines above as the shape of the workflow and confirm exact syntax with each tool's `--help` or documentation.

TRY THIS

Pick any three buildings in your sample. For each, write down: its **id**, the **LoDs** it carries, the **surface types** it contains, and **one attribute**. If you can do that, you can read CityGML.

CHAPTER 14

14 Eight Classic Pitfalls

Most beginner problems fall into one of these buckets.

Pitfall	Why it bites	What to do
Mixing up versions	1.0, 2.0 and 3.0 differ in element names and structure	Always record the version next to the file
Equating LoD with accuracy	LoD is detail, not correctness	Ask for source, date and accuracy
Ignoring the coordinate system	Wrong CRS puts the city in the wrong place; height reference and axis order trip people up	Read srsName first and keep units and vertical datum in mind
Trusting invalid geometry	Volumes and simulations come out wrong	Validate before analysing
Opening huge files directly	A big GML can freeze a generic viewer	Convert to CityJSON or tile it first
Exporting straight to a mesh	Semantics and attributes disappear	Keep the semantic master, export meshes only for display
Assuming ADE data travels	Unknown extensions may be dropped	Test a round trip; ship the schema
Treating CityGML as BIM	No construction layers or MEP	Use IFC for building detail and link the two

CHAPTER 15

15 Quick Quiz

Ten questions. Answers are at the bottom of the page. No peeking, or at least no sighing.

1. What does a CityGML model contain that a plain 3D mesh does not?
2. Which level of detail is an extruded block with a flat roof?
3. Which level of detail was removed in CityGML 3.0?
4. Which surface type is the footprint of a building?
5. What is `gml:id` used for?
6. In one sentence, what is CityJSON?
7. Which tool checks geometry against ISO 19107 rules?
8. What does ADE stand for?
9. True or false: an LoD3 model is always more accurate than an LoD2 model.
10. Name a database designed to store CityGML models.

Answers

1 Semantics: types, attributes and relationships for each object. | **2** LoD1. | **3** LoD4 (indoor features can now appear in LoD0 to LoD3 instead). | **4** GroundSurface. | **5** A unique identifier, so objects and shared geometry can be referenced. | **6** A compact JSON encoding of a subset of the CityGML data model, an OGC Community Standard. | **7** `val3dity`. | **8** Application Domain Extension. | **9** False. LoD is detail, not accuracy. | **10** The 3D City Database.

16 Cheat Sheet and Glossary

Keep this page open while you work.

If you want to...	Reach for...
Know what a file contains	A text editor: find CityModel, srsName, one object
Check shapes are sound	val3dity
Check a CityJSON file against its schema	cjval
Switch between CityGML and CityJSON	citygml-tools, cjio
Look at a model quickly	ninja, azul, QGIS plugin
Share and query a big model	3D City Database with citydb-tool
Make something web-friendly	CityJSON, or 3D Tiles via tyler

Glossary

Term	Meaning
ADE	Application Domain Extension: a schema-based extension of CityGML for one domain.
Attribute	A property of an object, such as height or year of construction.
BIM	Building Information Modelling: detailed, object-based models of single buildings.
BoundarySurface	A surface bounding a building or room, such as a roof, wall or floor.
cjio / cjval	A Python command-line tool for CityJSON, and a CityJSON validator.
Code list	An agreed table that maps codes to meanings, used for values like building function.
CRS	Coordinate Reference System: how coordinates relate to places on Earth.
CityJSON	A JSON encoding of a subset of the CityGML data model.
Digital twin	A live, data-connected model of a real asset or city.
Feature	An object in a geospatial model, such as a building or a road.
GML	Geography Markup Language, an XML grammar for geographic data.
gml:id	A unique identifier for an object in a GML file.
IFC	Industry Foundation Classes, the open standard for BIM data.
ISO 19107	The standard that defines the spatial geometry rules CityGML relies on.
LoD	Level of Detail: how much shape and meaning a model contains.
Mesh	A surface made of triangles with no built-in meaning.
OGC	Open Geospatial Consortium, the body that publishes the CityGML standard.
Semantics	The meaning of things: their type and role.
Solid	A closed volume bounded by polygons.
srsName	The element or attribute that names the coordinate system.
Texture	An image draped over a surface to show its appearance.
3DCityDB	The 3D City Database, a spatial database for CityGML models.

17 Where to Go Next

Read less, look at real data more.

- **CityGML 3.0 Conceptual Model** (the standard and its public workspace): github.com/opengeospatial/CityGML-3.0CM
- **CityJSON standard page** at the OGC: ogc.org/standards/cityjson
- **CityJSON software list**, with viewers, converters and validators: cityjson.org/software
- **CityJSON validation tutorial**: cityjson.org/tutorials/validation
- **3D City Database documentation**: 3dcitydb.github.io/3dcitydb-mkdocs
- **citydb-tool** source and releases: github.com/3dcitydb/citydb-tool
- **val3dity** documentation: val3dity.readthedocs.io

A good next step is to pick one dataset, take it through the five steps of Chapter 13, and write down three analysis questions that only a semantic model can answer. Then try to answer one.

FOR THE ARCHITECT A closing thought

A drawing is a conversation between people. A semantic model is a conversation between people *and* machines. Learning CityGML is learning to speak to both at once.