Engineer

QGIS

the free desktop GIS

What it is

QGIS is a desktop application for looking at, editing and analysing map data. The project started in May 2002 under the name Quantum GIS, and its first goal was modest: a viewer for spatial data. It has since grown into a full geographic information system that opens layers from files, spatial databases and web map services, draws them together on one map, edits their geometry and attributes, runs analysis over them and lays the result out as a printed map, an atlas or a report.

It is an official project of the Open Source Geospatial Foundation, released under the GNU General Public License, written in C++ on the Qt toolkit, and it runs on most Unix platforms, Windows and macOS. A new release comes out roughly every four months. Every third one, a pattern that began with 2.8, becomes a long-term release, which the project keeps maintaining until the next one replaces it, while the latest release gets a bug-fix point release each month. At the time of writing those are 3.44.15 and 4.2.3.

QGIS reads a great deal through other projects. Vector files come in through the GDAL drivers, which is why the list of formats it opens is so long, and rasters arrive the same way. Tables in PostGIS, SpatiaLite, SQL Server and Oracle are read in place, and the project's own feature list makes the point that data is shown in its own format and projection, with no conversion to an internal one.

Four further pieces matter once QGIS is shared by a team.

The project is the unit of work. A `.qgs` file is XML listing every layer's data source with its style, labels, forms and the print layouts; a `.qgz` file is a zip archive that holds that XML together with a `.qgd` file of auxiliary data. A project

can also be stored inside a GeoPackage or a PostgreSQL database, beside the layers it describes.

Processing is the analysis framework: a toolbox of algorithms from QGIS's own provider and from GDAL, with GRASS available as a further provider, a model designer that chains algorithms into a graph, a batch interface, and a history of every run. The same algorithms run without the desktop through a command-line tool, `qgis_process` :

```
qgis_process run native:reprojectlayer -- \
  INPUT=roads.shp TARGET_CRS=EPSG:25832 OUTPUT=roads.gpkg
```

PyQGIS is the Python interface behind plugins, the built-in console and standalone scripts. And QGIS Server publishes a project file over the web as standard OGC services, WMS, WFS, WCS and OGC API Features, running behind a web server as a FastCGI or CGI application.

Use the concept in a decision

Put "What is QGIS?" to work in a real decision. We can help relate the concept to the system you are building or buying. Anything that happens in a place — vessels, vehicles, plots, sensors — put on a map you can search, measure and share.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Made a geographical data pipeline 50x faster — 50x faster geographical data pipelines

Made a geographical data pipeline 50x faster Get in touch →

Why it matters

Some spatial work stays manual however good the pipeline is: deciding whether a parcel boundary is wrong or the aerial photo under it is, tracing a footprint from a LiDAR surface, looking over a client's delivery before it is loaded. The decision QGIS bears on is which tool that judgement happens in, and what is left behind once it has been made.

Licence and platform settle the first half. There is no seat to buy, and the same release installs on Linux, macOS and Windows, so a contractor, the client's own staff and a build server can all run the version the project was made in. Esri's commercial desktop, ArcGIS Pro, runs on Windows only.

The second half is about what survives the session. QGIS reads PostGIS directly, so an edit lands in the table the application serves from and there is no export to reconcile later. The Processing history records each algorithm with its parameters, and any entry can be copied out as a Python command or as a `qgis_process` command. A correction that was worked out by hand in a dialog can become a line in a script that runs on every delivery after it. A project file can go further and be served as it stands by QGIS Server, so the styling a person set up on the desktop is the styling a web client receives over WMS.

The costs are mostly about discipline. A project points to data it does not contain, so it is only as portable as the paths it stores. Choosing between the long-term and the latest release is a real choice: one changes slowly, the other brings features every four months.

What it looks like in practice

On this site QGIS shows up as the place where spatial data was looked at and repaired by hand, beside the tools that did the bulk work.

From 2015 to 2022, the GIS Software Development Team Lead role ran more than thirty GIS projects end to end, building, deploying and then supporting each one once it was live. In that stack QGIS and JOSM were where the hands-on data work was done, while PostgreSQL with PostGIS held the data, GDAL/OGR moved it between shapefiles, GeoJSON, GeoTIFF, KML and vector tiles, and the Mapbox GL and Leaflet web maps displayed it. Between the 2D and 3D digital mapping at one end of those projects and the indoor mapping and navigation at the other came LiDAR processing, georectification and vectorisation.

From 2022 to 2024, as a Data Engineer on a large-scale geospatial analytics project, the job was finding and fixing over 1,000 issues in geographical and time-series data. Spatial errors were diagnosed and corrected with GDAL, QGIS and ArcGIS together, and the cleaning tasks that kept recurring were automated in Bash scripts. PostGIS carried the spatial queries and indexing, and the work contributed to a 35% reduction in spatial query run times.

Where it goes wrong

The desktop makes spatial data look right on screen, and that is where most of the trouble hides.

A layer lands in the wrong place. QGIS reprojects every layer on the fly to the project's coordinate reference system, so layers in different systems overlay cleanly, provided each layer declares its own system correctly. A shapefile that arrives without its `.prj` file declares none, and depending on a setting under Options, QGIS leaves it unknown, prompts for a system, or quietly applies the project's or a default one. The repair depends on which mistake was made. When the coordinates are right and only the label is wrong, Assign projection or Define projection change the label and leave every vertex where it is. When the coordinates really are in another system, Reproject layer converts them. Find projection, in the same toolbox group, lists the systems in which a layer's extent would fall over a chosen area, which turns a guess into a short list.

Areas come out in square degrees. The expression language has two area functions that look interchangeable. `$area` follows the project's ellipsoid and unit settings, so with an ellipsoid set it returns an ellipsoidal area in the unit chosen. `area($geometry)` is always planimetric in the layer's own system and returns that system's units, so on a layer stored in longitude and latitude it returns square degrees, a small decimal that looks plausible in a table. Use `$area` for any figure a person will read, or reproject the layer to a projected system suited to the area before measuring.

A project opens with half its layers missing. A project stores where each layer's data lives, not the data. Move the folder, rename a file or open the project on a colleague's machine and the stored paths stop resolving. QGIS then shows the Handle Unavailable Layers dialog; choosing to keep them opens the project with those layers listed but drawing nothing until each source is repaired. Saving paths relative to the project file, a project property, lets a folder of project and data move as one. For shared work, storing the project in the same PostgreSQL database or GeoPackage as its layers keeps the two together.

The same correction is clicked through again next month. A fix made in a dialog leaves the data right and the method in someone's memory. The next delivery with the same fault is fixed by hand again, slightly differently. Copy the run out of the Processing history as a `qgis_process` command the first time it works, and put it in the script that loads the delivery.

Further reading

- Foreword — the project's own account of what QGIS is, where it started and the licence it is released under.

- Road Map — the release cycle, how long-term releases are chosen and the current schedule.
- Working with Projections — on-the-fly reprojection and what QGIS does with a layer that has no coordinate system.
- Working with Project Files — the `.qgs` and `.qgz` formats, projects stored in databases and the repair of broken paths.
- Using processing from the command line — `qgis_process`, its commands and how to pass parameters.
- Introduction — the QGIS Server manual's overview of the services it publishes from a project.

Glossary

Choose for your actual constraints

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Make the terminology practical

Ask how "What is QGIS?" applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/qgis/