

Feature List

This document outlines several key features of the Geotools toolkit. Geotools development is active and there are [many ideas](#) for future development. The features on the following page refer to current development on the SVN trunk. For features included in specific releases, see the following pages:

Specifications and Interfaces

Geotools implements [Open Geospatial Consortium \(OGC\) specifications](#) as they are developed.

Geotools is associated with the [GeoAPI](#) project that is creating interfaces for Java GIS projects. Many of these interfaces are implemented in geotools with more implementations planned.

Data Formats

Data is the life blood of any GIS project, so geotools tries to support as many data formats as possible.

The DataStore interface is used to access vector data.

Grid coverage (raster) data formats are accessed with the GridCoverageExchange interface.

DataStore

Data formats supported are:

- Shapefile - an ESRI shapefile (R/W)
- [GML](#) - Geography Markup Language (R)
- [WFS](#) - Features from an OGC Web Feature Server (RW)
- [PostGIS](#) - geometric objects for [PostgreSQL](#) (R)
- Oracle Spatial - Oracle's extension for spatial data (R)
- ArcSDE - ESRI's middleware for spatial databases (R)
- [MySQL](#) - support for the new [geometry types](#)
- GeoMedia - an Intergraph format (R)
- Tiger - Topologically Integrated Geographic Encoding and Referencing developed at the US Census Bureau (R)
- [VPF](#) - Vector Product Format, a data interchange format (R) (**work in progress**)
- [MapInfo](#) - MIF (Mapinfo Interchange Format) (RW) (**work in progress**)

GridCoverageExchange

See [Grid Coverage Exchange Design](#) for design notes about this new interface.

The following data formats are supported:

- ArcGrid - [ArcInfo ASCII Grid format](#) and [GRASS ASCII Grid format](#) (optionally compressed) (R/W)
- Image - can load [images](#) georeferenced with a [world file](#) (R/W)
- [GeoTIFF](#) - a georeferenced tiff image (R) (**work in progress**)
- WMS - OGC Web Mapping Server client (R) (**work in progress**)

Legacy DataSource (grid coverage)

Deprecated DataSource raster implementations are:

- GTopo30 - [GTOPO30](#) file format, a publicly available world wide DEM (R)

Features and Geometries

Features are used to represent basic geographic elements in a vector system. These are composed of both geometries and attributes.

[Java Topology Suite \(JTS\)](#) is used as the current geometry model. This library is an implementation of the Simple Features Specification for SQL, <http://www.opengis.org/docs/99-049.pdf>. It also provides in memory spatial indexing support and **robust** topologic operations, though they are not used much at present in the codebase.

If you're willing to add some vector computation operations (for example, overlay) please step up: we need someone willing to work on an operation API and the associated implementations. See the [Process API](#) page for more information.

Grid coverages

A grid coverage provides support for the raster data format, that is, a georeferenced grid of numbers that can be used to represent data ranging from satellite images, to digital elevation models, noise distribution and so on. See also the OpenGIS specification about grid coverages, <http://www.opengis.org/docs/01-004.pdf>.

Grid coverages are based on the [JAI](#) library which provides support for data management, presentation, image data format access, tiling support, a framework for raster data processing with lots of predefined operators.

Coordinate Transformation

The `cts-coordtrans` (CTS) module implements a subset of the OGC's Coordinate Transformation Services specification, <http://www.opengis.org/docs/01-009.pdf>. It provides an implementation for general positioning, coordinate reference systems, and coordinate transformations.

Types of transformations provided include datum shifts, map projections (Mercator, Transverse Mercator, Lambert Conformal Conic, Albers Equal Area Conic, Stereographic, Orthographic) and others.

Renderers

Geotools has two renderer implementations:

- `LiteRenderer` - a 'stateless' renderer, particularly useful in server-side environments. Main features: code simplicity and low memory footprint, but it's faster than `J2D` only for the first map rendering.
- `J2D` - a 'stateful' renderer with optimizations for update rate, good for interactive client-side displays. It's way more complex but supports on the fly coordinate reprojection, fast update by intelligent data decimation and memory caching.

Styling

Allows for symbolizing (colour, line thickness, opacity, etc.) of datasets. This is closely modeled on, and fully supporting, the OpenGIS Styled Layer Descriptor specification, <http://www.opengis.org/docs/02-070.pdf>. These can be stored as XML documents.

Filter Encodings

Filters provide a way to specify a subset of features to operate on. This supports both attribute and spatial constraints and a method of storing these as XML documents.

This is an implementation of the Filter Encoding specification, <http://www.opengis.org/docs/02-059.pdf>

GUI

This is currently an active area of development. See the [GUI Architecture](#) page for information about where development is heading.

The Legend module was recently added. Also maps and some tools (zoom, pan, etc.) are present.

Graph

Allows the creation of graphs and networks based on `MultiLineStrings` or arbitrary Geometries and Spatial Relationships. Graphs can be walked using the visitor pattern. Common walks, such as shortest route, are already available. Other walks may also be defined by the user.

Validation

This provides a method to validate features based on different constraints. Allows the definition of test suites based on validation plug-ins. Test suites (and indeed validation plug-ins) are configured using XML files.

A working validation Processor is provided as a demo.