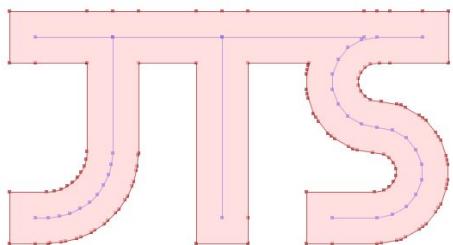




State of JTS

Presented by:
James, Jody, Rob, (Martin)



LocationTech

Welcome

Martin Davis	James Hughes	Jody Garnett	Rob Emanuele
Vivid Solutions	CCRI	Boundless	Azavea

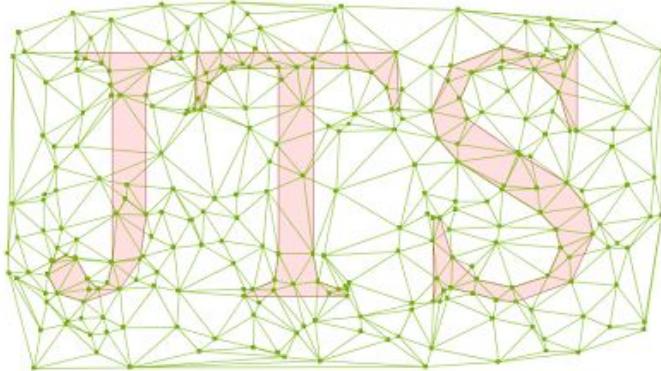
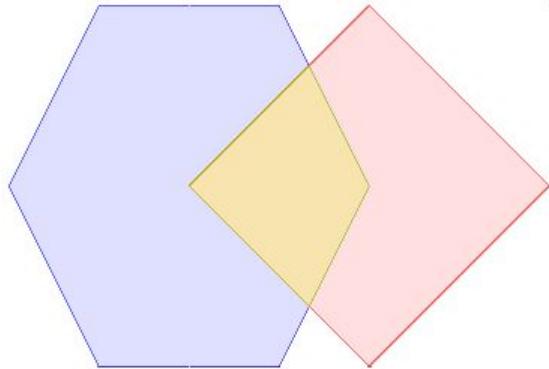




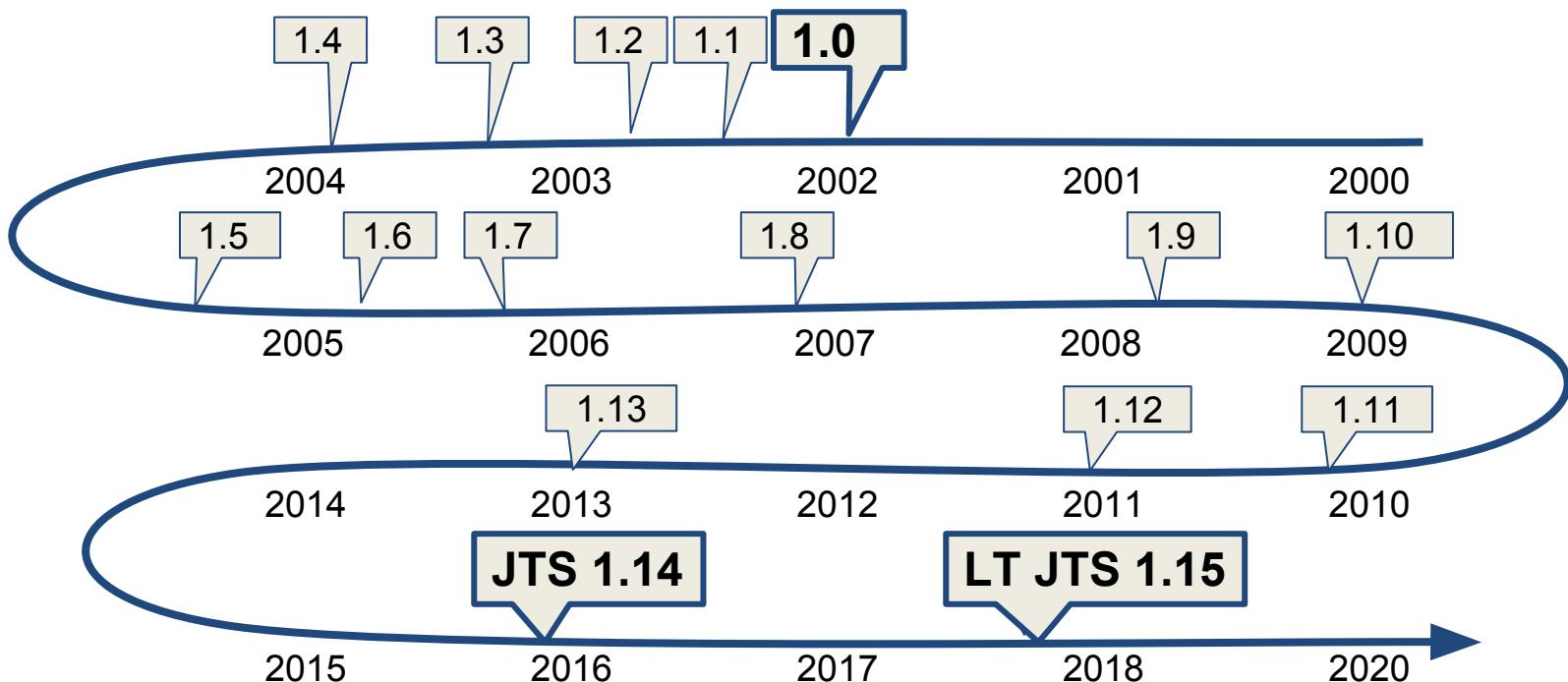
Introducing JTS Topology Suite

What is JTS Topology Suite?

Java API for working with **2D Geometries**



JTS Project History



JTS is EVERYWHERE

Net Topology Suite

JTS

OGR
geoDjango
SourceGIS
R-GEOSS
MapServer
WebProcessingServer
FME
MapWindow
MapGuide
SpatiaLite
PostGIS
Open
Quantum
Shapely
MonetDB
RGeo

GEOS

JSTS



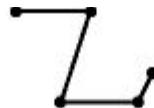
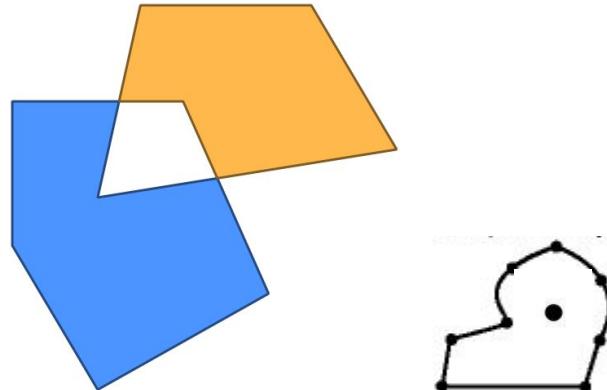
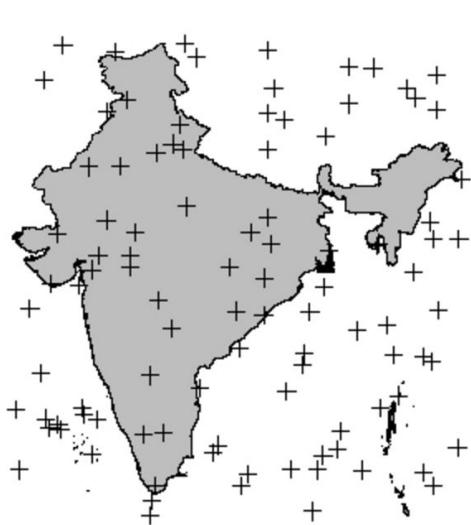
LocationTech

JTS Topology Suite

Representations:

OGC Simple Features

- Point
- LineString
- LinearRing
- Polygon
- MultiPoint
- MultiLineString
- MultiPolygon
- GeometryCollection



JTS Topology Suite

Predicates (DE-9IM)

- Equals
- Disjoin
- Intersects
- Touches
- Crosses
- Within
- Contains
- Overlaps
- Covers
- CoveredBy

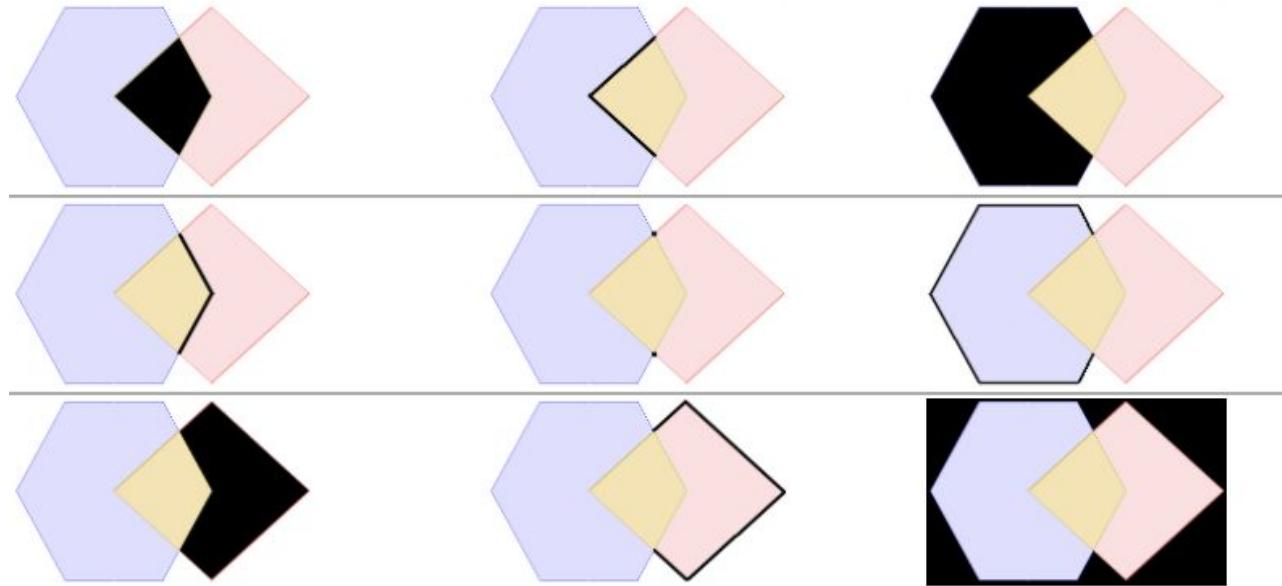


	Interior	Boundary	Exterior
Interior	The intersection of the interior regions of shapes a and b is shaded red. $\dim[I(a) \cap I(b)] = 2$	The intersection of the boundary regions of shapes a and b is shaded red. $\dim[I(a) \cap B(b)] = 1$	The intersection of the exterior regions of shapes a and b is shaded red. $\dim[I(a) \cap E(b)] = 2$
Boundary	The intersection of the boundary regions of shapes a and b is shaded red. $\dim[B(a) \cap I(b)] = 1$	The intersection of the boundary regions of shapes a and b is shaded red. $\dim[B(a) \cap B(b)] = 0$	The intersection of the exterior regions of shapes a and b is shaded red. $\dim[B(a) \cap E(b)] = 1$
Exterior	The intersection of the exterior regions of shapes a and b is shaded red. $\dim[E(a) \cap I(b)] = 2$	The intersection of the boundary regions of shapes a and b is shaded red. $\dim[E(a) \cap B(b)] = 1$	The intersection of the exterior regions of shapes a and b is shaded red. $\dim[E(a) \cap E(b)] = 2$

JTS Topology Suite

Overlays

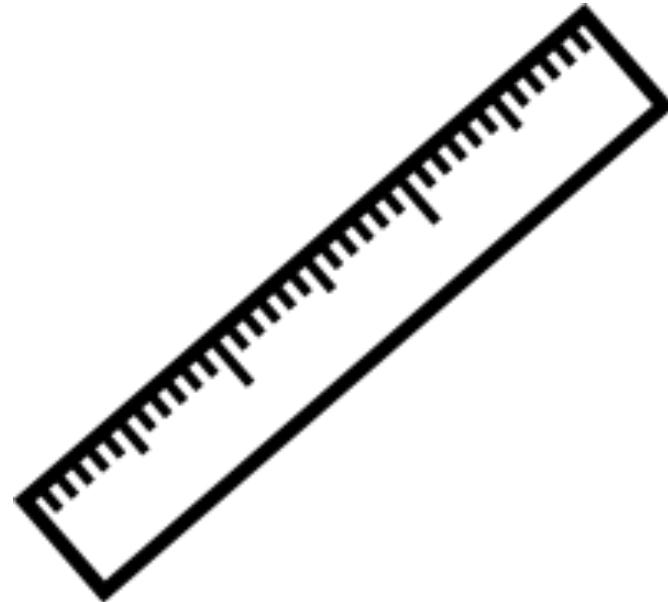
- Intersection
- Union
- Difference
- SymDifference



JTS Topology Suite

Measurements

- Length
- Area
- Distance



JTS Topology Suite

IO:

- WKT
- WKB
- GeoJSON
- KML
- GML2

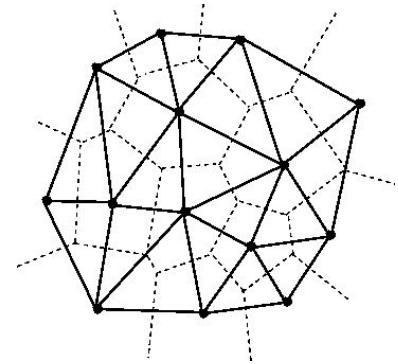
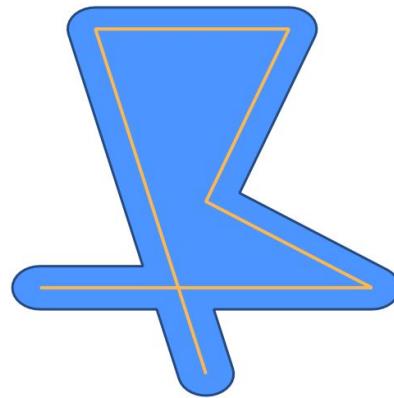
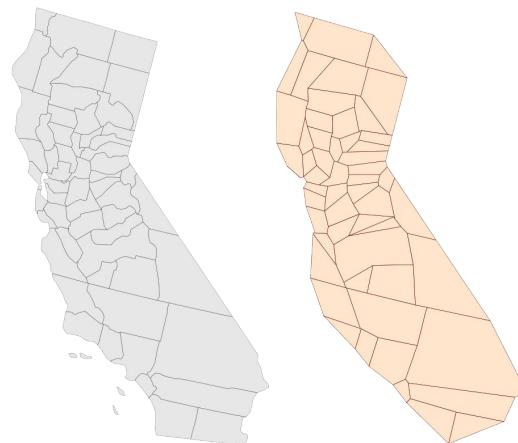
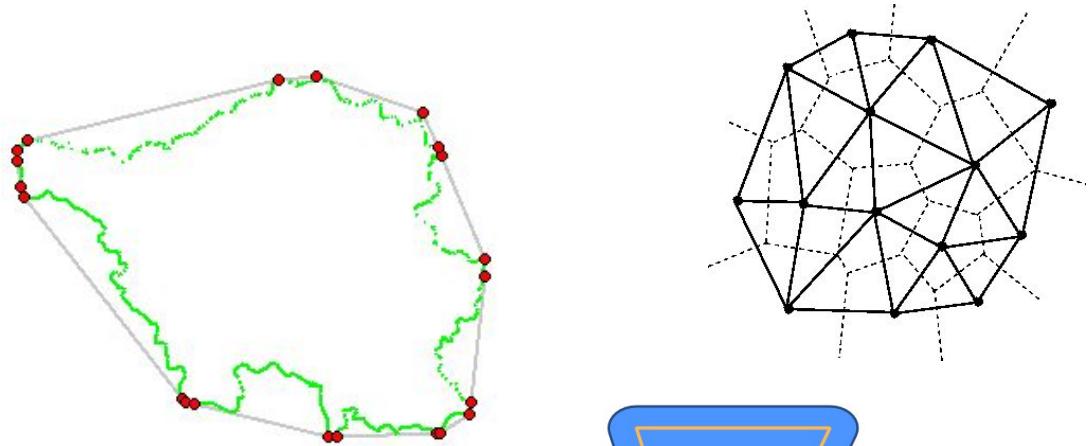
```
wkt_geom
Polygon ((-105.03792611059080286
39.78014782225491786, -105.04818400099962616
39.75856265597848704, -105.02284438556741009
39.75418720873850731, -105.01231287864754904
39.76789982851657612, -105.01364722199988933
39.78389171288461768, -105.03792611059080286
39.78014782225491786))
```

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [
      -122.65335738658904,
      45.512083676585156
    ]
  },
  "properties": {
    "name": "Hungry Heart Cupcakes",
    "address": "1212 SE Hawthorne Boulevard",
    "website": "http://www.hungryheartcupcakes.com",
    "gluten free": "no"
  }
}
```

JTS Topology Suite

Algorithms

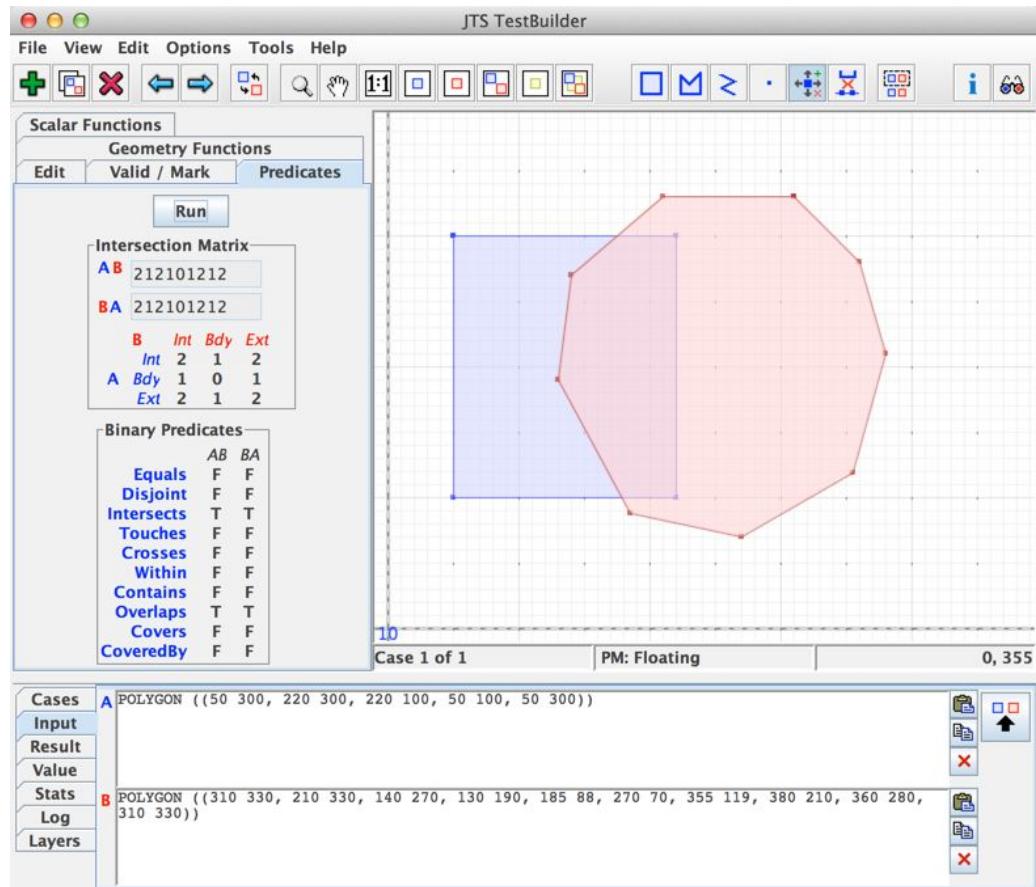
- Convex Hull
- Buffer
- Validation
- Dissolve
- Polygonization
- Simplification
- Triangulation
- Voronoi
- Linear Referencing
- and more...



JTS Topology Suite

Applications

- TestBuilder
- TestRunner





JTS 1.14

JTS 1.14 Release

January 2016

- LineDissolver
- edgegraph package
- Visvalingam-Whyatt simplification



Visvalingam-Whyatt
vs
Douglas-Peucker

Improvements:

- Improved thread-safety
- Fixed Java 7 compatibility
- Added Spatialite WKB
- CoordinateSequence
- many bug fixes and performance improvements

JTS I/O

- KML Writer
- GeoJsonReader/Writer
- Oracle SDO Performance

JTS 1.14 with Maven

JTS 1.14

```
<dependency>
  <groupId>com.vividsolutions</groupId>
  <artifactId>jts-core</artifactId>
  <version>1.14.0</version>
</dependency>
```

Published

Official release on SF

- Install into local repo

On Maven Central

- We do not know
who did this!



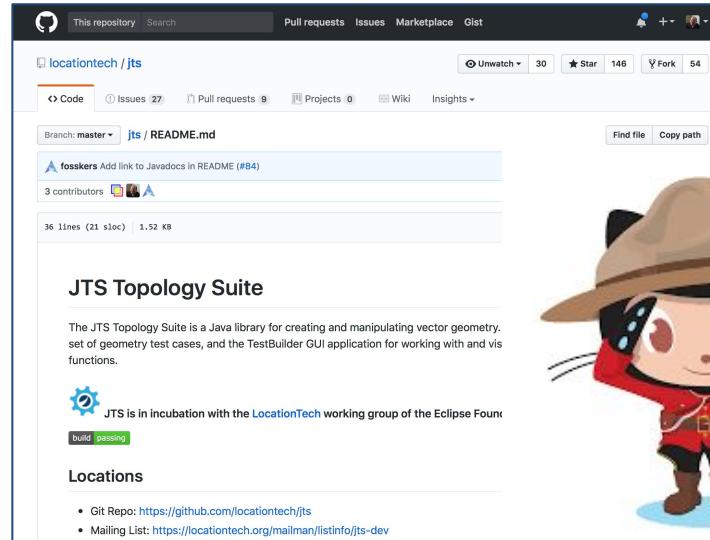
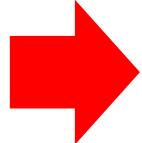
JTS 1.15

JTS 1.15 Release

- Focus on codebase
 - organization and packaging
- Some functionality improvements
 - K Nearest Neighbor search for STR-Tree
 - Improve handling of Quadtree queries with null Envelope
 - Intersects now supports GeometryCollection
 - JTSTestRunnerCmd command-line app

Sourceforge → GitHub

- Moving from SVN to GIT
- <https://github.com/locationtech/jts>



This repository

locationtech / jts

Code Issues Pull requests Marketplace Gist

Branch: master | jts / README.md

fosskers Add link to Javadocs in README (#84)

3 contributors

36 lines (21 sloc) | 1.52 KB

JTS Topology Suite

The JTS Topology Suite is a Java library for creating and manipulating vector geometry. It includes a set of geometry test cases, and the TestBuilder GUI application for working with and visualizing geometry functions.

JTS is incubation with the LocationTech working group of the Eclipse Foundation.

build passing

Locations

- Git Repo: <https://github.com/locationtech/jts>
- Mailing List: <https://locationtech.org/mailman/listinfo/jts-dev>



Why choose GitHub?

- High Visibility
- Great tools
 - Git tools
 - Issue tracking
 - Pull Requests
 - Continuous Integration
 - Website
- Easier for contributions
- Where the action is!



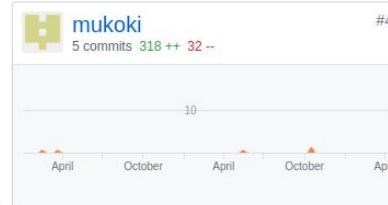
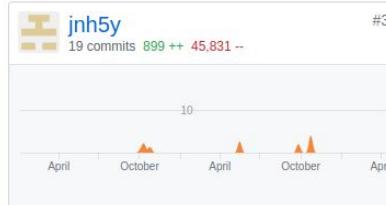
GitHub: JTS Project Activity

- Pull Requests
 - 76 accepted, 15 open
- Issues
 - 39 closed, 49 open

Jan 31, 2016 – May 11, 2018

Contributions: **Commits** ▾

Contributions to master, excluding merge commits



Mavenization

- Build chain now uses Maven instead of Ant
 - Easier to build and use
 - Easy Eclipse IDE configuration
- Unit tests run by Maven build
 - including XML tests
- Better release story
 - Code artifacts will be hosted on Maven Central
 - Apps built as fat-jars (TestBuilder, TestRunner)
- To Do
 - Work on packaging a distro with source, scripts, etc...



Modular Codebase

- Codebase organized into modules
 - **jts-core** - geometry implementation for use
 - **jts-tests** - extensive testing for correctness and stability
 - **jts-io** - read and write geometry
 - **jts-example** - examples of using the jts api
 - **jts-lab** - experimental playground use at your own risk
 - **jts-app** - test builder application for defining tests
- better clarity of internal dependencies

JTS Joins LocationTech

- LocationTech offers
 - project infrastructure
 - project visibility
 - stability, governance
- Immediate benefits
 - More team members
 - Synergy with other LocationTech projects
 - In-depth legal review for IP (Intellectual Property) cleanliness
- Initial Work
 - Project Application
 - License Change
 - LocationTech Incubation
 - Build Infrastructure
 - Official Maven Deployment
- Long term hopes
 - Additional Contributors
 - Funding for JTS 2.0

LocationTech Incubation

A new License

- Eclipse Public License
- Eclipse Distribution License
(BSD-3 Clause License)

Challenges:

- Contact assorted contributors
(because we did not have a CLA)
- changing package names
- Opportunity to work together
- Maintaining codebase history

A new home:

- Project Website
- Mailing List
- Build Server
- GitHub repo

LocationTech Project Site

- www.locationtech.org/projects/technology.jts

The screenshot shows the JTS Topology Suite project page on the LocationTech website. The header includes the LocationTech logo, a search bar, and a 'DONATE' button. The navigation menu has links for TECHNOLOGY, MEMBERS, EVENTS, STEERING COMMITTEE, and ABOUT US. The breadcrumb navigation shows: HOME / ECLIPSE WORKING GROUPS / LOCATIONTECH / TECHNOLOGY / JTS TOPOLOGY SUITE / JTS TOPOLOGY SUITE. The main content area is titled 'JTS Topology Suite' and features tabs for Overview, Downloads, Who's involved, Developer Resources, Governance, and Contact Us. The 'Overview' tab is selected. Below it, there is a brief description of the JTS Topology Suite, mentioning it is an open source Java software library for planar geometry and conforms to the Simple Features Specification for SQL. It also lists the 'Licenses:' (Eclipse Distribution License 1.0 (BSD) and Eclipse Public License 1.0), 'Contribution Activity:' (a chart showing commits over the last 12 months), and 'PROJECT LINKS' (links to Javadoc, Mailing List, Website, Code Repository, Hudson (HIPP), Travis CI, and Documentation). A small 'eclipse incubation' logo is visible on the right.

JTS 1.15

- Packaging
 - `org.locationtech.jts`
- GitHub repo
 - <https://github.com/locationtech/jts>
- Releases available on Maven Central
(and LT Nexus)
- Snapshots Available via LT Nexus
 - <https://repo.locationtech.org/>

Using JTS 1.15 with Maven

JTS 1.14

```
<dependency>
  <groupId>com.vividsolutions</groupId>
  <artifactId>jts-core</artifactId>
  <version>1.14.0</version>
</dependency>
```

JTS 1.15.0

```
<dependency>
  <groupId>org.locationtech.jts</groupId>
  <artifactId>jts-core</artifactId>
  <version>1.15.0</version>
</dependency>
```

Using JTS 1.15.1-SNAPSHOT

JTS 1.14

```
<dependency>
  <groupId>com.vividsolutions</groupId>
  <artifactId>jts-core</artifactId>
  <version>1.14.0</version>
</dependency>
```

JTS 1.15.1-SNAPSHOT

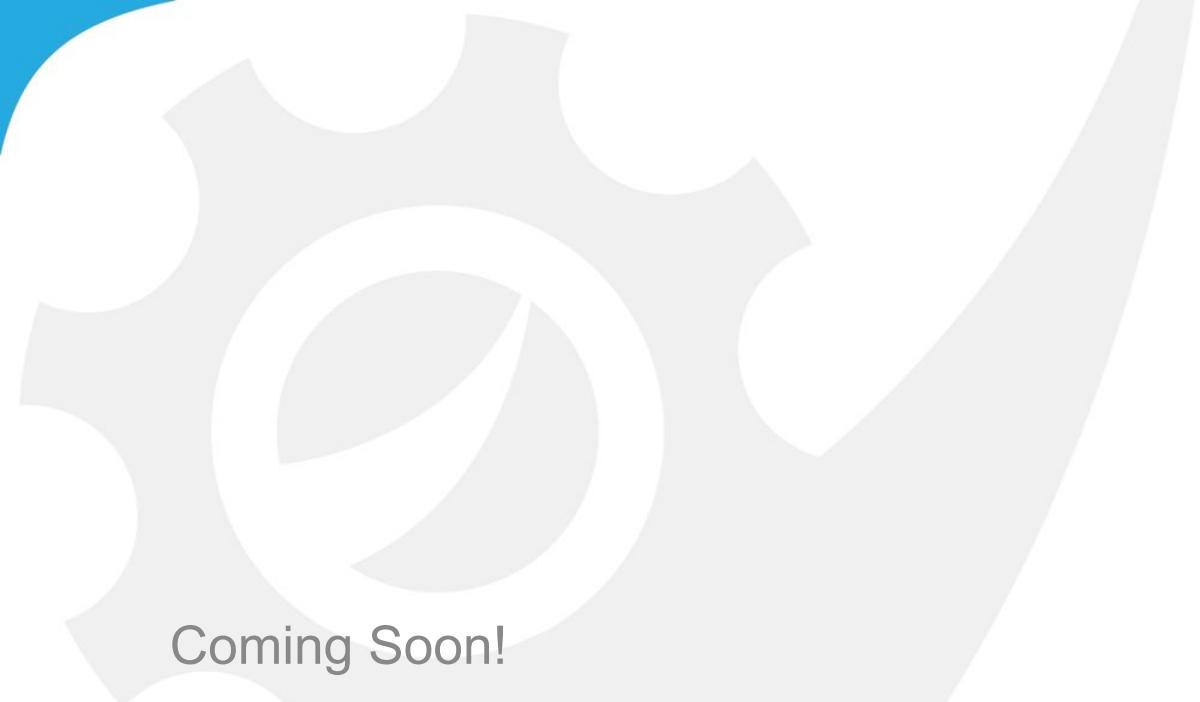
```
<dependency>
  <groupId>org.locationtech.jts</groupId>
  <artifactId>jts-core</artifactId>
  <version>1.15.1-SNAPSHOT</version>
</dependency>
.....
<repositories>
  <repository>
    <id>locationtech-snapshots</id>
    <url>https://repo.locationtech.org/content/groups/snapshots</url>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
</repositories>
```

Migration to JTS 1.15

- New module structure
 - `jts-core`
 - `jts-io-common` - GeoJSON
 - `jts-io-ora` - Oracle support
 - `jts-io-sde` - SDE support
 - `jts-tests` - XML Tests & TestRunner
- Change package names
 - `org.locationtech.jts.*`
- Change Maven reference
 - GroupId change: `com.vividsolutions` to `org.locationtech.jts`

JTS 1.15 Coming Soon!

- Coming soon to a repo near you!
 - Incubation is nearly complete
- LocationTech Release process
 - Final IP issues being resolved
(checking in new icons for the test builder application)
 - Two week release review
- Deploy to Maven Central (and LocationTech repo)



Coming Soon!

JTS 1.15.1 Roadmap

JTS 1.15.1 Planning

- Support projects migrating
- Java Roadmap Compatibility
 - module names for “jigsaw” packaging
- Some functionality improvements
 - TWKB - Reader for PostGIS TWKB format

"Tiny Well-known Binary" or "TWKB"

JTS Topology Suite

IO:

- WKT
- WKB
- GeoJSON
- KML
- GML2
- TWKB (In progress!)

Version	Release
0.23	May 1, 2015

Abstract

TWKB is a multi-purpose format for serializing vector geometry data into a byte buffer, with an emphasis on minimizing size of the buffer.

Why not WKB?

The original OGC "well-known binary" format is a simple format, and is capable of easily representing complex OGC geometries like nested collections, but it has two important drawbacks for use as a production serialization:

- it is not aligned, so it doesn't support efficient direct memory access; and,
- it uses IEEE doubles as the coordinate storage format, so for data with lots of spatially adjacent coordinates (basically, all GIS data) it wastes a lot of space on redundant specification of coordinate information.

A new serialization format can address the problem of alignment, or the problem of size, but not both. Given that most current client/server performance issues are bottlenecked on network transport times, TWKB concentrates on solving the problem of serialization size.

Basic Principles

TWKB applies the following principles:

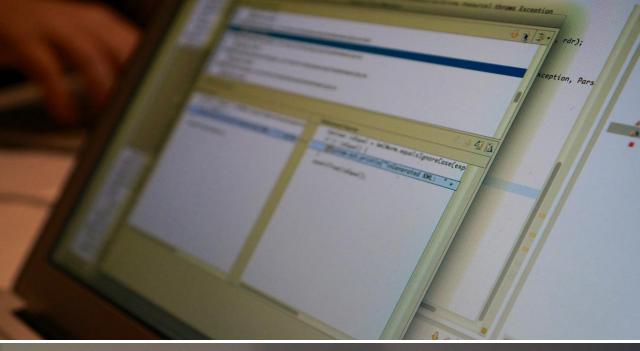
- Only store the absolute position once, and store all other positions as delta values relative to the preceding position.
- Only use as much address space as is necessary for any given value. Practically this means that "variable length integers" or "varints" are used throughout the specification for storing values in any situation where numbers greater than 128 might be encountered.



JTS Community Building

JTS Team Code Sprints

- 2016 January and November
 - Sourceforge → GitHub
 - Build change to maven
 - Addressed “Intellectual Property” review questions



JTS Team Code Sprints

- 2018 Bon Code Sprint
 - Java 10 compatibility with “jigsaw” module names
- 2017 FOSS4GNA Code Sprint
 - Helping projects upgrade
 - GeoTools PostGIS DataStore TWKB
 - Join us Thursday for Community and workshop day!





JTS in the ‘Cloud’

JTS is EVERYWHERE

Net Topology Suite

JTS

OGR geoDjango
SourceGIS
R-GEOSS
MapServer
FME
WebProcessingServer
SpatiaLite
MapWindow
MapGuide
PostGISOpen
Quantum
Shapely
MonetDB
RGeo

GeoTrellis HatBox SWCDF MapJAI Tools
degree geoKettle Mapyrus JUMP SWECDF
RoadMatcher GeoScript DeeJUMP Straightedge
Puzzle-GIS OpenJUMP MoxieMedia SkyJUMP
JCSSuite GISGeoTools GEQL Kosmo Sextante
GeoDB DB2 uDig GeOxygene Hibernate
SpatiaLite gvSIG Geomesa JASPA RasterFrames
IMF Conflation OGC Geomajas GeoServer
GeoWave Kosmo

GEOS

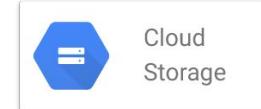
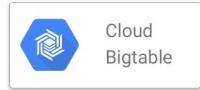
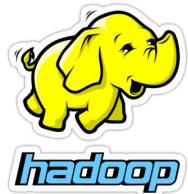
JSTS



Cloud Projects using JTS



Big Data Ecosystem



Distributed Spatial Goals

- Distribute the **storage** of vector and raster data
 - Database integration (HBase, Accumulo, C*)
 - File format integration (Arrow, Avro, Parquet)
- Distribute the **processing** of geospatial data
 - MapReduce integration
 - Spark integration
 - **SparkSQL**

JTS + Spark

- In 2015-2016
 - GeoMesa had RDD level support for JTS Geometry types (as well as GeoTools SimpleFeatures)
- In 2017
 - GeoMesa integrated with Spark's SQL query planner
 - Added Spatial UDTs
 - Added Spatial UDFs
 - Adds PostGIS syntax to Spark
 - (Limitation) Tied to GeoMesa
- In 2018
 - GeoMesa project refactored JTS+Spark module
 - Being used by the RasterFrames project

JTS + Spark going forward

- The JTS + Spark integration is pretty straightforward; the goal is to have more projects integrate with it.
 - Performance enhancements can be shared by all the projects
- Currently, each Spark release introduces changes to the UDT/UDF protected interfaces.
 - This risk is best shared by a community (rather than having each project reimplement and update their individual projects)



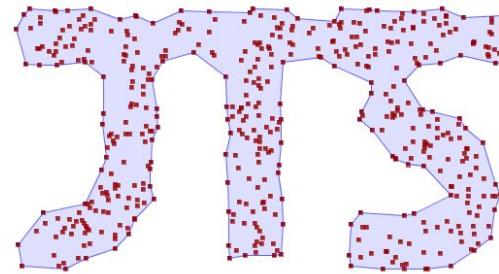
JTS 2.0 Roadmap / Wishlist

Algorithm Improvements

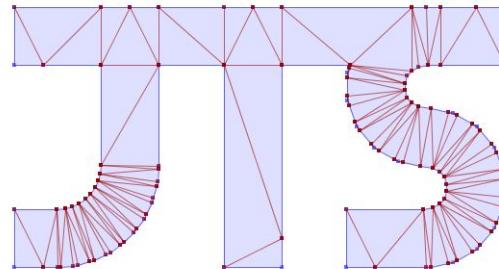
- Goal: improve some key JTS algorithms
 - Overlay
 - Snap-rounding (no more TopologyExceptions!)
 - Support PreparedGeometry for caching
 - Fast & robust Clip to Rectangle
 - Spatial Predicate improvements
 - Streaming / Lazy evaluation with short-circuiting
 - User-defined precision model
 - Less sensitive to valid geometry (e.g. Intersects)
 - Distance
 - Support cached PreparedGeometry

New Algorithms

- Concave Hull
- Polygon Triangulation
- Polygon Cleaning (“MakeValid”)
- Split Geometry by Line
- Polygon Coverage Simplification



Concave Hull



Polygon Triangulation

New API - JTS 2.0

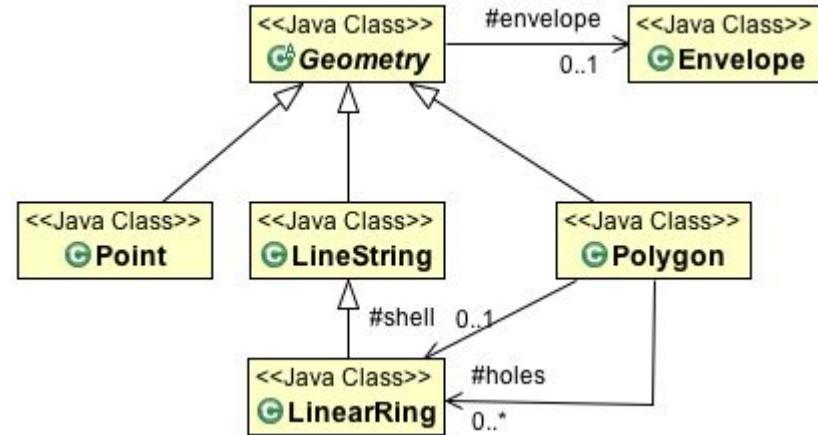
- Concept for a redesign of JTS
- Key Goals
 - Interface-based Geometry access
 - Immutable Geometry objects
 - Geodetic (WGS84) support, with some basic algorithms
 - Pluggable/discoverable Geometry operation framework
 - Coordinate extensions (XY, XY+M)
- Non-goals
 - Backwards compatibility
 - Improving geometry algorithms

JTS 1.0 Baseline

SFQL, GML2

Primitives:

- Geometry
- Point
- LineString
- Polygon

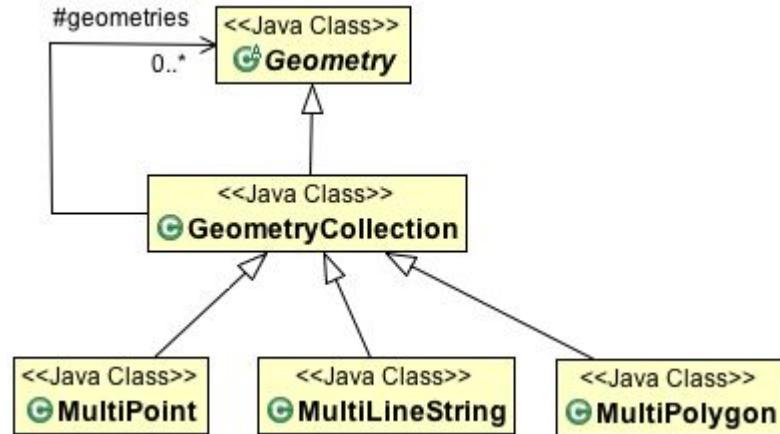


JTS 1.0 Baseline

SFQL, GML2

Collections

- GeometryCollection
- MultiPoint
- MultiLineString
- MultiPolygon

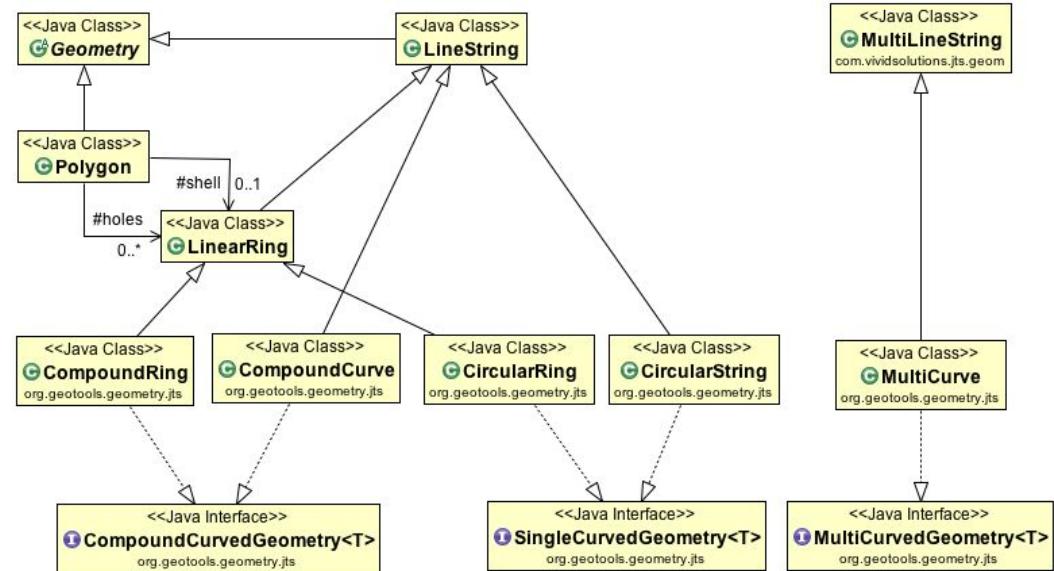


JTS 2.0 Challenge

SQL/MM, GML3, ISO19107

Primitives:

- Point
- Curve
- Surface

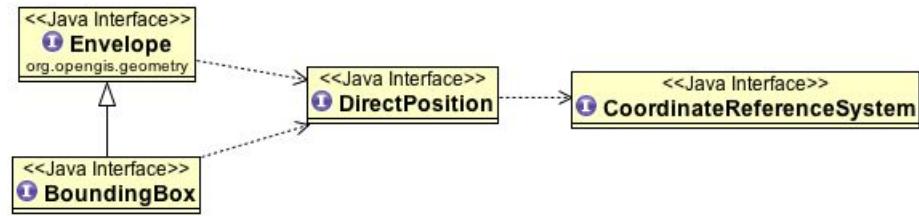


JTS 2.0 Challenge

SQL/MM, GML3, ISO19107

Geometry defined using:

- Positions
- Reference System



JTS 2.0 Challenge

JTS Topology Suite

Linear Geometry

Euclidean operations

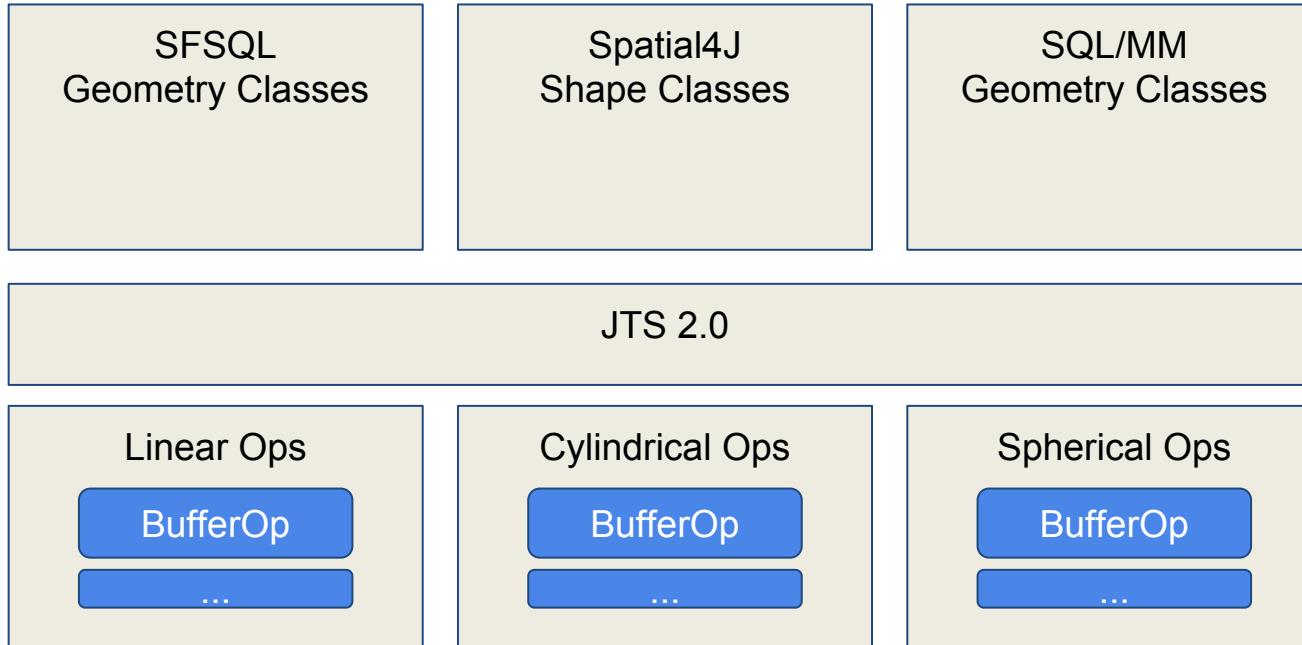
Spatial4J

Curved Geometry

Cylindrical operations

Spherical operations

JTS 2.0 Approach



A large, semi-transparent graphic of a mechanical gear is positioned in the upper left corner of the slide. It has a light gray-to-white gradient and is partially cut off by the slide's edge.

Join JTS Topology Suite

Shape the Future

Contributing to JTS

- Register as a Contributor
 - Sign the Eclipse Contributor Agreement
 - <https://www.eclipse.org/legal/ECA.php>
- Develop a patch, making sure to include
 - Javadoc
 - Unit Tests - JUnit and/or JTS XML tests
- Make a Pull Request on GitHub
 - Acknowledge code is IP clean by signing-off each Git commit
 - Make sure the Travis CI validation tests pass

See also <https://github.com/locationtech/jts/blob/master/CONTRIBUTING.md>

Join Us at the Code Sprint!

Thursday, at the code sprint, we will work on two projects

1. Polishing a new TWKB Reader/Writer
2. Upgrading the GeoServer ecosystem to LocationTech JTS



Questions?

Project Resources

- Source Code repo
 - <https://github.com/locationtech/jts>
- Issue Tracker
 - <https://github.com/locationtech/jts/issues>
- Mailing List
 - <https://dev.locationtech.org/mailman/listinfo/jts-dev>
- Project website
 - <https://locationtech.github.io/jts>
- Javadoc
 - <https://locationtech.github.io/jts/javadoc>



Thank you from the JTS Team