State of JTS

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

<table>
<thead>
<tr>
<th>Martin Davis</th>
<th>James Hughes</th>
<th>Jody Garnett</th>
<th>Rob Emanuele</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vivid Solutions</td>
<td>CCRi</td>
<td>Boundless</td>
<td>Azavea</td>
</tr>
</tbody>
</table>

![Vivid Solutions logo](image1)
![CCRi logo](image2)
![Boundless logo](image3)
![Azavea logo](image4)

![Martin Davis](image5)
![James Hughes](image6)
![Jody Garnett](image7)
![Rob Emanuele](image8)
Introducing
JTS Topology Suite
What is JTS Topology Suite?

Java API for working with 2D Geometries
JTS Project History

- 2000: Version 1.10
- 2001: Version 1.9
- 2002: Version 1.8
- 2003: Version 1.7
- 2004: Version 1.6
- 2005: Version 1.5
- 2006: Version 1.4
- 2007: Version 1.3
- 2008: Version 1.2
- 2009: Version 1.1
- 2010: Version 1.11
- 2011: Version 1.12
- 2012: Version 1.13
- 2013: Version 1.14
- 2014: Version 1.15
- 2015: Version 1.16
- 2016: Version 1.17
- 2017: Version 1.18
- 2018: Version 1.19
- 2019: Version LT JTS 1.16
JTS is EVERYWHERE
Vector Data (Lines)
Vector Data (Polygons)

German Regions

### Vector Data

**Public Building**

<table>
<thead>
<tr>
<th>Name</th>
<th>Julius Nyerere International Convention Centre</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multilingual name</td>
<td></td>
</tr>
<tr>
<td>Abbreviation</td>
<td>JNICC</td>
</tr>
<tr>
<td>Address</td>
<td>123 Shaaban Robert Street, Dar es Salaam</td>
</tr>
<tr>
<td>Levels</td>
<td>1</td>
</tr>
<tr>
<td>Height (Meters)</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

![Map of Julius Nyerere International Convention Centre](image)

**LocationTech**
Contains

True

False
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

<table>
<thead>
<tr>
<th></th>
<th>Interior</th>
<th>Boundary</th>
<th>Exterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interior</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \dim[I(a) \cap I(b)] = 2 )</td>
<td>( \dim[I(a) \cap B(b)] = 1 )</td>
<td>( \dim[I(a) \cap E(b)] = 2 )</td>
</tr>
<tr>
<td>Boundary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \dim[B(a) \cap I(b)] = 1 )</td>
<td>( \dim[B(a) \cap B(b)] = 0 )</td>
<td>( \dim[B(a) \cap E(b)] = 1 )</td>
</tr>
<tr>
<td>Exterior</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \dim[E(a) \cap I(b)] = 2 )</td>
<td>( \dim[E(a) \cap B(b)] = 1 )</td>
<td>( \dim[E(a) \cap E(b)] = 2 )</td>
</tr>
</tbody>
</table>
JTS Topology Suite

Overlays

- Intersection
- Union
- Difference
- SymDifference
JTS Topology Suite

Measurements

- Length
- Area
- Distance
JTS Topology Suite

IO:

- WKT
- WKB
- GeoJSON
- KML
- GML2

```
```

```
{  
  "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

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

```xml
<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 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
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
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](http://www.locationtech.org/projects/technology.jts)
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 1.15.1

<dependency>
    <groupId>org.locationtech.jts</groupId>
    <artifactId>jts-core</artifactId>
    <version>1.15.1</version>
</dependency>
Using JTS 1.15.2-SNAPSHOT

JTS 1.14

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

JTS 1.15.2-SNAPSHOT

<dependency>
  <groupId>org.locationtech.jts</groupId>
  <artifactId>jts-core</artifactId>
  <version>1.15.2-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 1.15.1

• Support projects migrating
• Java Roadmap Compatibility
  • module names for “jigsaw” packaging
• Fixes
  • Geometry.clone() → Geometry.copy()
Coming Soon!

JTS Roadmap
JTS 1.16 Coming Soon!

• How soon?
  • Release Candidate 1.16.0-RC1 is already available
  • Scheduled 1.16.0 for September 5th Eclipse release review
    • Final IP issues being resolved
      (checking in new icons for the test builder application)
    • Two week release review
**CoordinateSequence**

**XYZM**

**Dimension:** number of ordinates in each coordinate, this total includes any measures.

**Measure:** number of measures included in dimension for each coordinate

<table>
<thead>
<tr>
<th>Coordinate</th>
<th>Dimension</th>
<th>Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>XY</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>XYM</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>XYZ</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>XYZM</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>
Coordinate XYZM

Subclasses for XY, XYM, XYZM representations.

The field “z” is deprecated! Please use accessors to access fields.
WKT XYZM Support

Felix has extended WKT reader/writer support to support XYZM.

The reader has a flag to support “legacy” JTS representation.

Supports Well-Known Text representations for Z, M, ZM forms:

LINESTRING (10 10, 20 20, 30 40)
LINESTRING Z(10 10 10, 20 20 10, 30 40 10)
LINESTRING M(10 10 11, 20 20 11, 30 40 11)
LINESTRING ZM(10 10 10 11, 20 20 10 11, 30 40 10 11)
JTS Topology Suite

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

"Tiny Well-known Binary" or "TWKB"

<table>
<thead>
<tr>
<th>Version</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.23</td>
<td>May 1, 2015</td>
</tr>
</tbody>
</table>

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

[Cloud of words including JTS, GeoTrellis, GeoMesa, GeoWave, Net Topology Suite, etc.]
Cloud Projects using JTS

- Geomesa
- GeoTrellis
- GeoWave
- RasterFrames
- GeoSpark
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 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
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

- SFSQL Geometry Classes
- Spatial4J Shape Classes
- SQL/MM Geometry Classes

JTS 2.0

- Linear Ops: BufferOp, ...
- Cylindrical Ops: BufferOp, ...
- Spherical Ops: BufferOp, ...

LocationTech
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](https://github.com/locationtech/jts)
- Issue Tracker
  - [https://github.com/locationtech/jts/issues](https://github.com/locationtech/jts/issues)
- Mailing List
  - [https://dev.locationtech.org/mailman/listinfo/jts-dev](https://dev.locationtech.org/mailman/listinfo/jts-dev)
- Project website
  - [https://locationtech.github.io/jts](https://locationtech.github.io/jts)
- Javadoc
  - [https://locationtech.github.io/jts/javadoc](https://locationtech.github.io/jts/javadoc)
Thank you from the JTS Team