

JTS TOPOLOGY SUITE

Version History

This document lists the Change History of release versions of the JTS Topology Suite

Version 1.4

Release Date: November 4, 2003

Semantics Changes

- none

Functionality Improvements

- Added "LINEARRING" tag to WKT syntax
- Added GeometryEditor class to allow easy copy/modify of Geometries
- Added GeometricShapeFactory class to easily create standard geometric shapes
- Geometries can now carry arbitrary user-defined data objects (via Geometry#getUserData(Object) method)
- Added CoordinateSequence and CoordinateSequenceFactory interfaces, and default implementations (BasicCoordinateSequence, BasicCoordinateSequenceFactory)
- Added Geometry#getFactory
- Added new PrecisionModel type of FLOATING_SINGLE, for rounding to single precision floating point
- Added DistanceOp#getClosestPoints method, which returns the closest points between two Geometries
- Added com.vividsolutions.jts.noding package containing classes to perform fast indexed noding of linestrings
- Added SimpleGeometryPrecisionReducer to allow reducing precision of coordinates of a Geometry
- Added LineSegment#getClosestPoints method to compute the closest points between two line segments
- Added MinimumDiameter class to compute minimum diameters of Geometries
- Added geom.Triangle class to contain algorithms for Triangles
- BufferOp now allows end cap styles to be specified. Three types are supported: round, butt and square.

Performance Improvements

- EdgeList now provides a findEqualEdge method which is substantially faster than findEdgeIndex, for large lists

- Buffering is now faster and much more robust
- Overlap operations are now more robust

Bug Fixes

- `Envelope#init(Envelope)` now handles null Envelopes correctly
- `CoordinateList#add()` now correctly ignores the z-value of Coordinates in determining equality
- `Geometry#isValid` now correctly handles checking validity of LinearRings
- Fixed infinite loop bug causing Out Of Memory errors during polygon intersection
- `Geometry#clone` now correctly clones the Geometry's Envelope
- `LineIntersector#computeEdgeDistance` now correctly computes a non-zero edge distance in certain situations when a fixed precision model was being used and the line segment was a single unit in length
- Fixed incorrect calculation of depths in `DirectedEdgeStar#computeDepths`
- Fixed `BufferSubgraph#addReachable` to use explicit stack to avoid stack overflow problems
- Fixed various bugs causing some kinds of buffers to be computed incorrectly

API Changes

- WKTRewriter/Writer: changed protected members to private
- PrecisionModel type is now an object rather than an int
- ConvexHull API changed to remove requirement to pass in CGAlgorithms object

Code Architecture Changes

- `geom.util` package added for utility classes which parse and modify geometries

Documentation

- More examples provided in `com.vividsolutions.jtsexamples` package
- JTS Developers Guide started

Version 1.3

Release Date: April 4, 2003

Semantics Changes

- all Geometry methods are now reentrant (thread-safe)
- Fixed-precision coordinates are now stored in a rounded but non-scaled form. This makes them compatible with non-precise (Floating) coordinates, and simplifies working with precise coordinates directly. Mixed precision models are now

supported in Geometry methods; method results are in the more precise of the input precision models.

- Offsets are no longer supported in the Fixed precision model. This is necessary to allow storing fixed precision coordinates in a non-scaled form. This does not reduce the total precision available, since coordinates are stored in a floating-point format.
- SRID and Precision Model are no longer checked for equality during Geometry operations. This removes a limitation which provided little semantic benefit.

Functionality Improvements

- added `Geometry.isWithinDistance(Geometry g, double distance)` method, to provide optimized proximity queries
- added `Geometry.buffer(double distance, int quadrantSegments)` method, allowing control over accuracy of buffer approximation
- added `Geometry.getCentroid()` method
- added `Geometry.getInteriorPoint()` method, which uses heuristic methods to return a point in the interior of a Geometry
- `GeometryFactory.toGeometryArray` now returns null if the argument is null

Performance Improvements

- Removed unnecessary string construction in `EdgeEndStar.propagateSideLabels()`
- Eliminated unnecessary computation of self-intersections in rings during relate and spatial functions. This provides a large increase in speed when working with large rings and polygons. (Note that `IsValid` still checks for these self-intersections, which are illegal in `LinearRings`)
- Add short-circuit code to `RobustLineIntersector` to detect non-intersections more efficiently

Bug Fixes

- Fixed `ClassCastException` occurring in `GeometryCollection.getLength()`
- Fixed bug in Edge Intersection insertion (replaced `Coordinate#equals` with `equals2D` to ensure that intersection creation is not sensitive to Z-value).
- Fixed handling `LineStrings` with too few points in `GeometryGraph.addLineString`
- Fixed: was not checking that `MultiPolygons` don't contain components with too few points.
- Fixed `Envelope.distance()` to return correct distance for all envelopes.
- Fixed a few Geometry methods to make them re-entrant.
- Fixed `CoordinateList.closeRing()` to ensure endpoints are not duplicated

- Fixed `CGAlgorithms.signedArea()` to use a simpler algorithm which is more robust and faster.
- Fixed bug preventing validating Rings containing an initial repeated point.

API Changes

- Added default constructor to `WKTRReader`. It uses the default `GeometryFactory`
- Add two static `intersects()` methods to `Envelope`, to allow computing intersections with envelopes defined by points only.
- Dropped `BinaryPower`; its functionality is provided by `DoubleBits` in a more robust fashion.
- Removed a couple of redundant private static methods from `Geometry`; they have been replaced by methods in `CoordinateArrays`
- The `Geometry` class is now marked as `Serializable`

Version 1.2

Release Date: 7 October 2002

Semantics Changes

- JTS now allows Geometries to have repeated points. All operations will continue to perform as before. This removes a significant incompatibility with the OGC spatial data model.
- `TopologyExceptions` may now be thrown by spatial overlay methods. This helps to distinguish between code bugs and known robustness problems. It also provides a machine-readable coordinate for the error location.

Functionality Improvements

- `RobustLineIntersector` now uses “normalized” coordinates to maximize the accuracy of intersection computation.
- Upgraded `Quadtree` with more robust implementation
- Replaced `IntervalTree` with a more robust implementation of `BinTree`
- Added `STRTree` 2-D spatial index, which exhibits better performance than `QuadTrees` in many situations.
- Added `EnhancePrecisionOp`, which uses precision enhancing techniques to reduce the number of failure cases due to robustness problems.

Bug Fixes

- fixed `ConvexHull` to use `TreeSet` instead of `HashSet` for coordinates
- Fixed `isValid` for `GeometryCollections` containing `Polygons`, which were sometimes erroneously returning a validity failure for correct Geometries.

- Fixed bug in `LineSegment.distancePointLine()` which would return the incorrect distance for a `LineSegment` with two identical points
- Improved error handling in `CGAlgorithms.isCCW()`
- `IsValid` now checks for too few points in a geometry component (e.g. due to repeated points in a ring)

API Changes

- added `Stopwatch` class
- added `Geometry.getArea()` and `Geometry.getLength()` methods
- added `CGAlgorithms.signedArea()` method
- added methods to `LineSegment` – `closestPoint()`, `getLength()`
- added `CoordinateArrays` and `CoordinateLists` utility classes
- Added `TopologyValidationError.getErrorType()` method
- Added `Envelope#intersects`; deprecated `Envelope#overlaps`.
- Added `Geometry#geometryChanged()` method to allow signaling when `Geometry` coordinates have been mutated by a client class
- Added `STRTree` class implementing a Sort-Tile-Recursive spatial index (a variant of a packed R-tree)
- Deleted `IntervalTree` 1-D spatial index (replaced by `BinTree`)
- Add `BinTree` 1-D spatial index

Version 1.1.1

Release Date: 9 April 2002

Bug Fixes

- fixed decimal-point symbol localization bug in `WKTWriter`
- fixed bug in `Envelope.int(Envelope env)`
- fixed filename case of `SFSMultiLineString.java` and `IntervalTree.java`

API Changes

- deleted `TopologyException` class
- renamed `CGAlgorithms.isPointInPolygon` to `isPointInRing` (a more accurate description of what the method computes)

API Additions

- added `Geometry.getCoordinate()` method
- added `Geometry.distance()` method

- added GeometryComponentFilter interface and Geometry.apply(GeometryComponentFilter) method

Version 1.1

Release Date: 28 March 2002

New Features

- added Geometry.isSimple() and Geometry.isValid() methods
- improved design of topological data structures
- added Geometry.setSRID() method
- improved functionality of the Envelope class
- added ability to write to an arbitrary java.io.Writer object to WKTWriter
- added Validate and Mark Location functionality to TestBuilder

Version 1.0

Release Date: 1 February 2002

- Removed some non-compatibilities with Java 1.1
- Fixed bug in constructing buffer outline around inside of angles
- In TestBuilder vertices are now displayed with fixed size in view units
- Improved code for WKTWriter.writeFormatted()
- Fixed bug in constructor for LinearRing
- Improved implementation of sweepline intersection algorithm to avoid use of dynamic set.
- Fixed bug in ConvexHull.cleanRing()
- Refactored RobustLineIntersector and NonRobustLineIntersector

Version 0.9

Release Date: 17 December 2001

- Removed PrecisionReducer class. Changed operation of overlay so that it computes precisely for Fixed precision inputs, and non-precisely for Floating precision inputs
- Cleaned up buffer package to remove unused classes
- Improved Technical Specs description of PrecisionModel, isClosed

Version 0.8

Release Date: 14 December 2001

- Replaced former geom interfaces with geom.sfs interfaces
- Renamed implementation classes from Ximpl to simply X, and moved them to the geom package
- Fixed bug in RobustLineIntersector (intersections at endpoints were not being computed exactly)

Version 0.7

Release Date: 5 December 2001

- Fixed bug in relate code

Version 0.6

Release Date: 10 September 2001

- improved HTML formatting
- TestRunner accepts xml filenames as program arguments
- code defect #5 fixed in relate code

Version 0.5

Release Date: 6 September 2001

- code defects #3 & 4 fixed in relate code
- normalize method added
- convexHull method implemented
- spatial analysis methods intersection, union, difference, symDifference implemented EXCEPT for LA case

Version 0.4

Release Date: 13 July 2001

- com.vividsolutions.jts renamed to com.vividsolutions.jtstest
- ca.gc.cgdi renamed to com.vividsolutions

Version 0.3

Release Date: 4 July 2001

JTS Framework

- GeometryImpl#equalsExact created to do pointwise comparison

Predicates

- Fixed Defect #1: Point-wise equal lines with self-intersections

Test Harness

- TestBuilder allows saving tests in XML format
- TestBuilder allows testing of spatial functions

Version 0.2

Release Date: 15 June 2001

- Relate code cleaned up, and handling of GeometryCollections clarified
- Minor changes to TestBuilder
 - Ø Test Case Text Viewer added, displaying TestCaseList Java code
 - Ø bugs in input of GeometryCollections fixed

Version 0.1

Release Date: 13 June 2001

- Wrote JavaDoc for geom, geom.impl, io, io.impl and util classes
- Implemented PointImpl.isSimple
- Reduced length of exception messages
- Added, removed, moved and renamed various classes and methods
- Envelopes are now created when needed for Geometrys
- Relate works for all combinations of Geometrys, with the following exceptions:
 - Ø one of the arguments is a Geometry Collection and contains Geometrys which share boundaries
- TestCaseList allows running validation TestCase classes from the commandline.
- com.vividsolutions.jts.testsuite contains TestCases for relate
- TestBuilder allows creating Geometrys, running relate tests, and loading TestCaseLists

Version 0.0

Release Date: 30 May 2001

- Baseline version