

# JTS & JUMP

## A Foundation for Spatial Processing



Martin Davis, Senior Technical Architect  
[mbdavis@vivid solutions.com](mailto:mbdavis@vivid solutions.com)

# Overview

- **JTS Topology Suite**

- Open-Source API for representing and manipulating **Geometry**

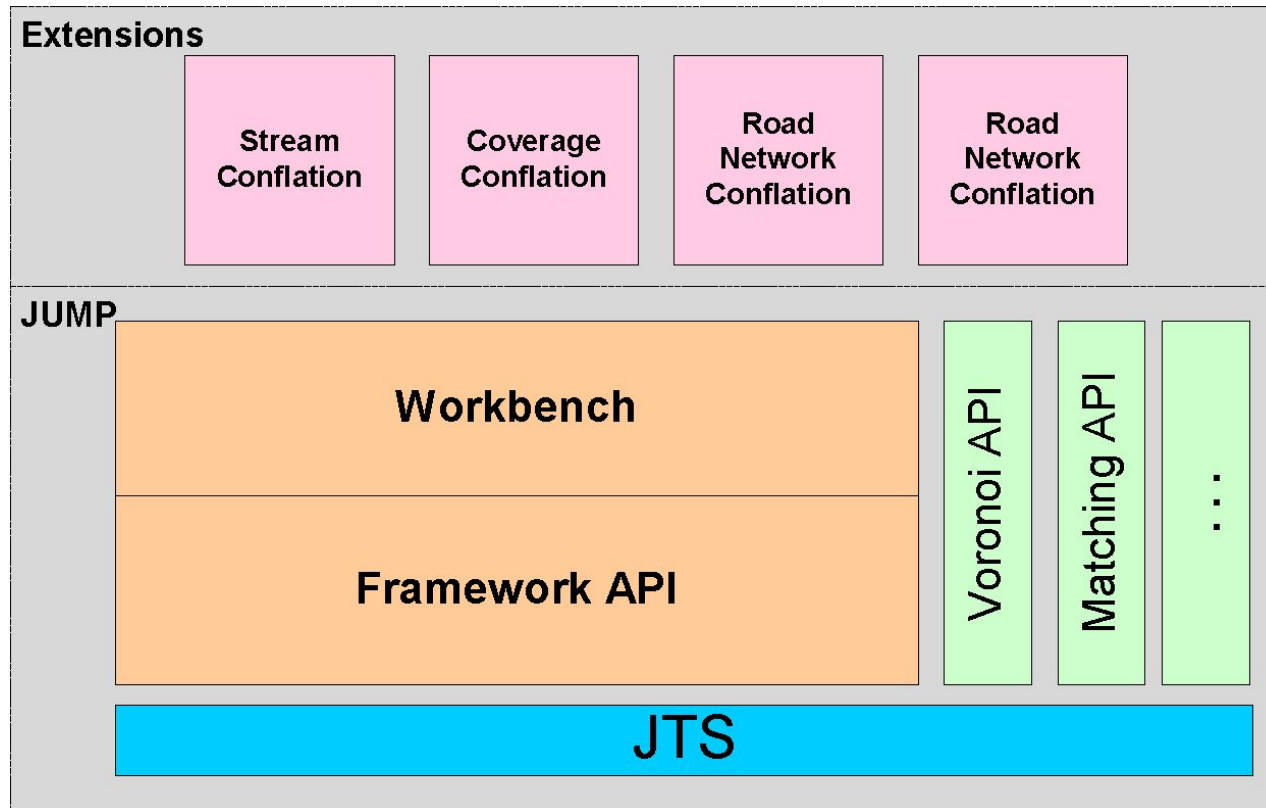
- **JUMP Unified Mapping Platform**

- Open-Source platform for representing, visualizing and manipulating **Geospatial Features**

- **Together: a foundation for spatial processing**

- Algorithm development
- Data Visualization
- Spatial Analysis
- Application deployment

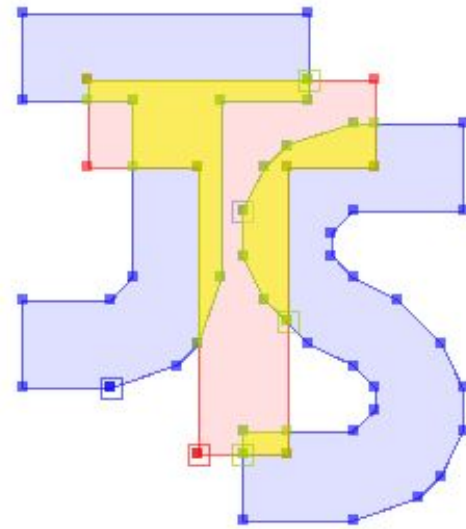
# JUMP & JTS - Architecture



- JUMP core is APIs and GUI Framework
- Applications (such as Conflation) are packaged as JUMP Extensions

# JTS Topology Suite

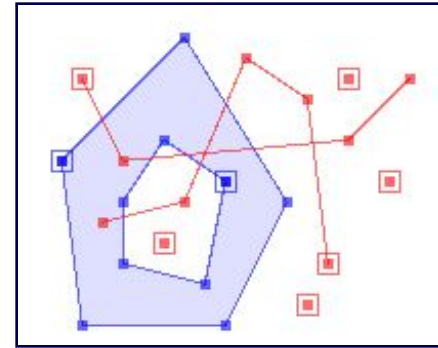
- **Core API for processing Geometry**
- **Full implementation of *OpenGIS Consortium Simple Features for SQL* specification**
- **Open Source, 100% Java**
- **Design Features:**
  - Fast, production quality
  - Robust
  - Explicit precision model
  - All basic geometry operations
- **History:**
  - JTS 1.0 released Feb 2002
  - JTS 1.4 released Nov 2003



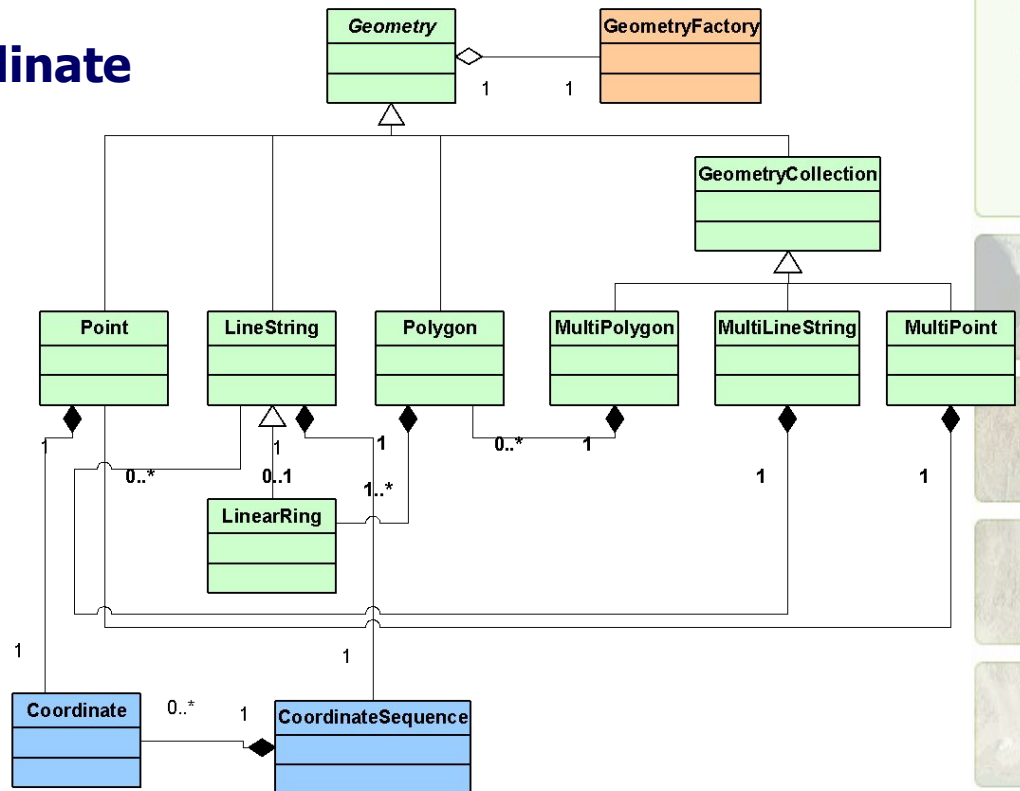
# JTS - Geometry Model

- **Complete model for 2-D linear geometric objects (following OGC SFS model)**

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



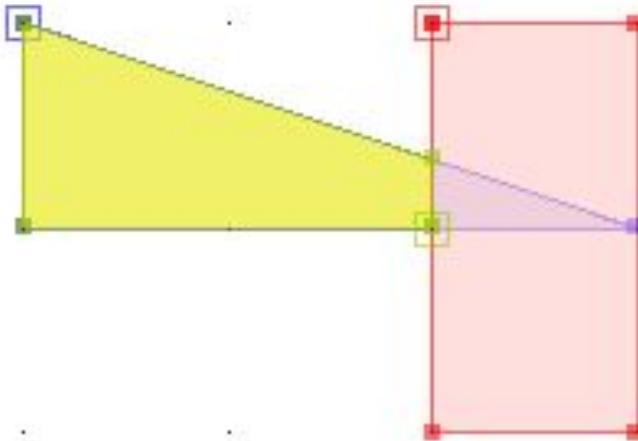
- **Supports user-defined coordinate representation**



# JTS – Precision Model

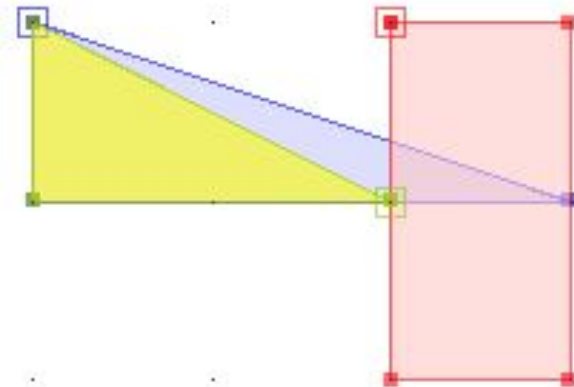
- **JTS provides ability to specify Precision Model of coordinates**
  - Double & Single Precision (IEEE-754)
  - Fixed (# of decimal places)
- **Explicit Precision Model ensures constructive geometry operations are closed over the coordinate space**

*Floating*



```
POLYGON ((3 2, 1 2, 1 3, 3  
2.333333333333333, 3 2))
```

*Fixed*



```
POLYGON ((3 2, 1 2, 1 3, 3 2))
```

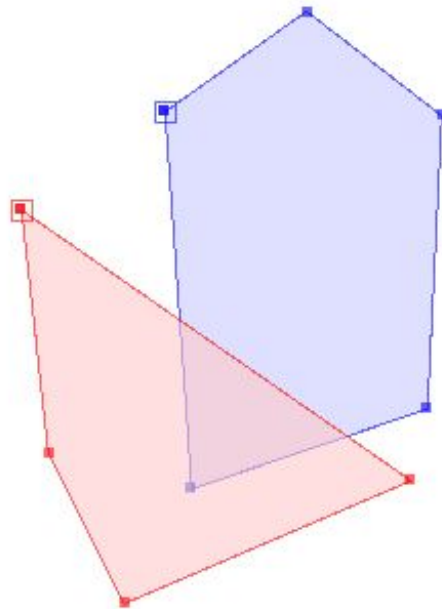
# JTS – Spatial Predicates

- Spatial predicates determine how Geometries interact
- JTS implements the full *Dimensionally Extended 9-Intersection Model (DE-9IM)*
  - General function: **Relate**
  - Named predicates: **Equals, Disjoint, Intersects, Touches, Crosses, Within, Contains, Overlaps**

		B		
		Int	Bdy	Ext
A	Int	2	1	2
	Bdy	1	0	1
	Ext	2	1	2

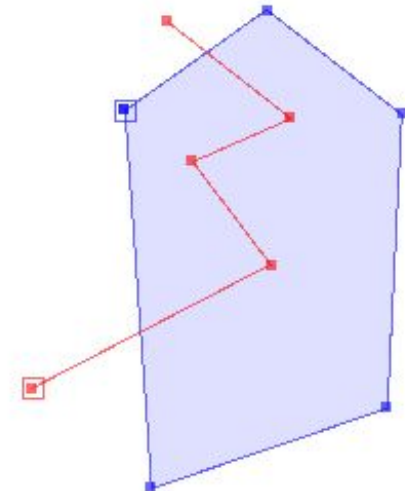
Binary Predicates			AB	BA
Equals			F	F
Disjoint			F	F
Intersects			T	T
Touches			F	F
Crosses			F	F
Within			F	F
Contains			F	F
Overlaps			T	T



		B		
		Int	Bdy	Ext
A	Int	1	F	2
	Bdy	0	F	1
	Ext	1	0	2

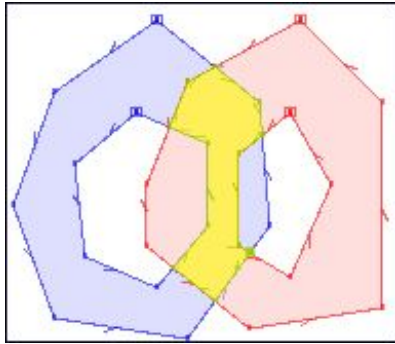
Binary Predicates			AB	BA
Equals			F	F
Disjoint			F	F
Intersects			T	T
Touches			F	F
Crosses			T	T
Within			F	F
Contains			F	F
Overlaps			F	F



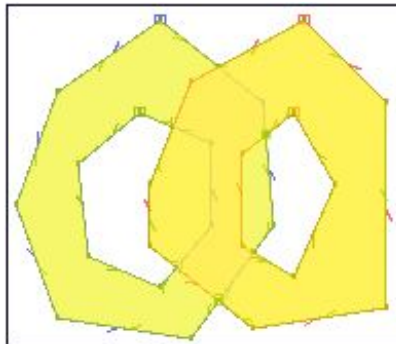
# JTS – Overlay Methods

- **Overlay methods ( = Boolean set-theoretic functions)**

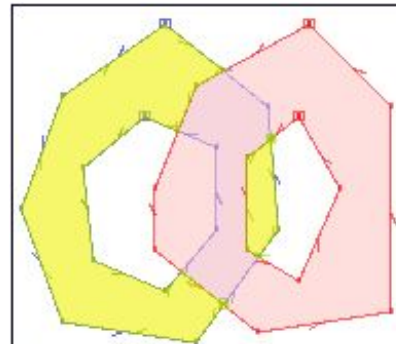
- Intersection
- Union
- Difference
- Symmetric Difference



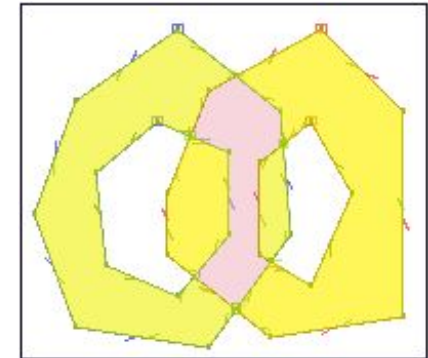
*Intersection*



*Union*

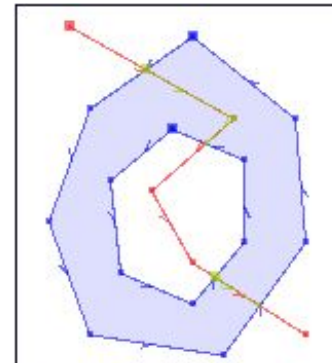


*Difference*



*Symmetric Difference*

- **Overlay methods are heterogeneous**  
– all geometry types supported

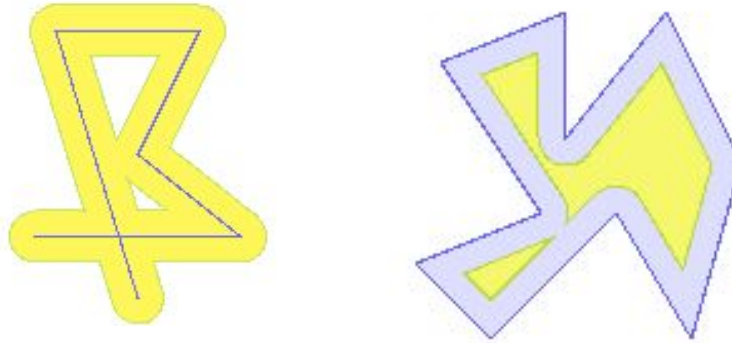




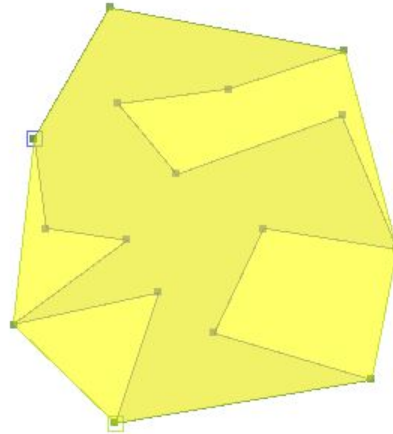
# JTS – Other Constructive Methods

- **Buffer**

- Positive & negative

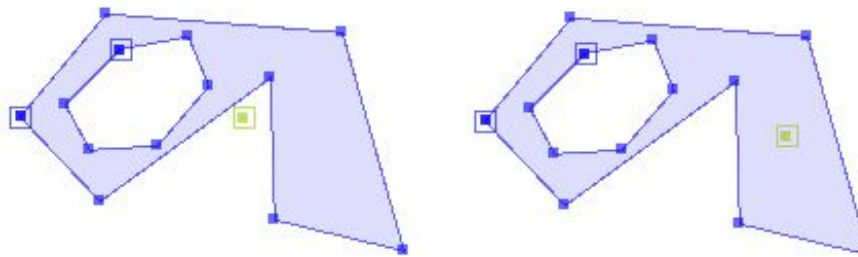


- **Convex Hull**



- **Centroid**

- **InteriorPoint**



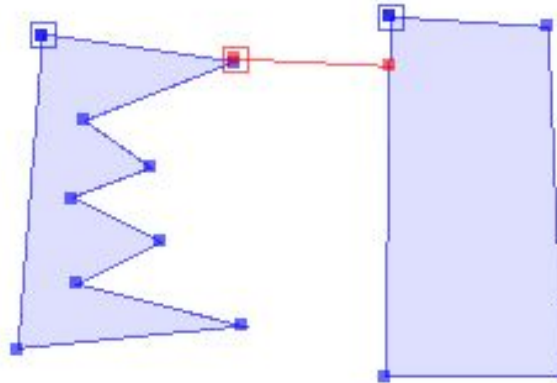
# JTS – Metric Methods

- **Area, Length**

- Length = Perimeter for Area geometries

- **Distance**

- Constructive – computes location of points providing minimum distance

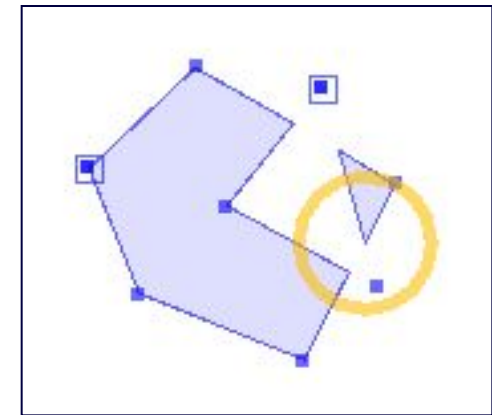
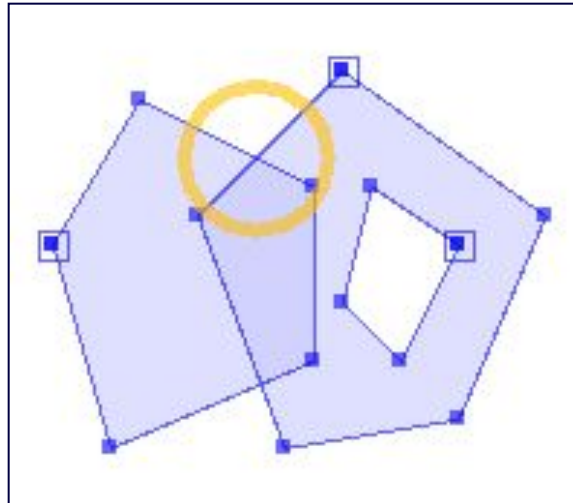
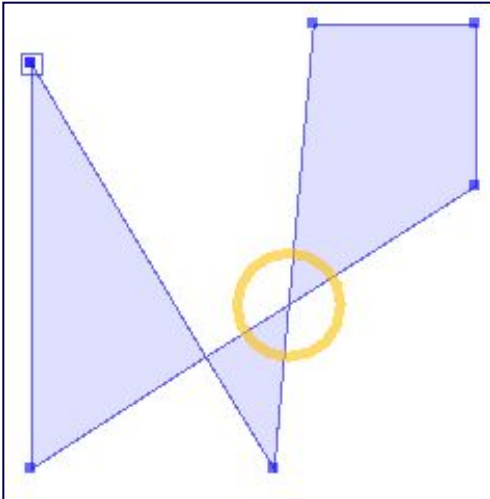


- **WithinDistance**

- “Limited predicate”, allowing optimization of computation

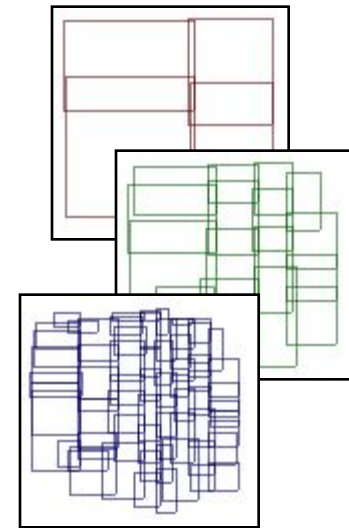
# JTS - Geometry Validation

- **Validation of Geometry topology essential to ensure correct spatial processing**
  - Polygons in particular!
- **JTS provides full Validation of Topology**
  - isValid provides simple good/bad test
  - ValidOp class provides detailed error reporting including location of error



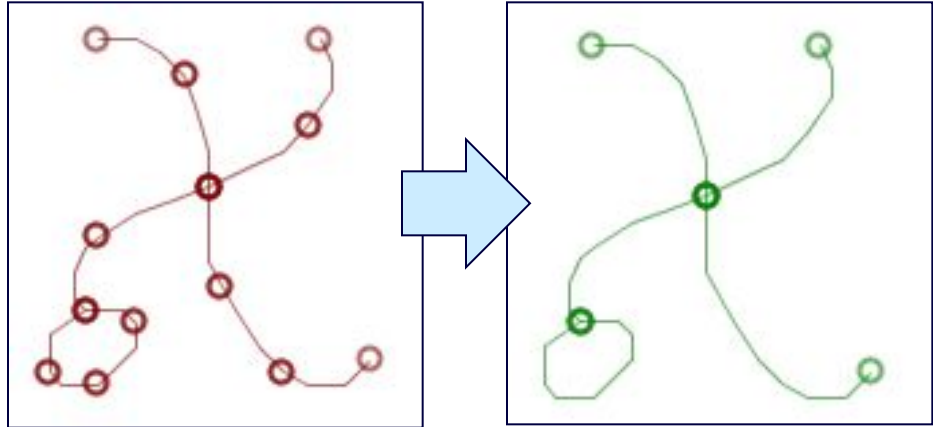
# JTS – Spatial Algorithms & Structures

- **Numerous basic Computational Geometry algorithms**
  - Line segment intersection, ring orientation, point/line orientation, point-line distance, etc.
- **Spatial Indexes**
  - Quadtree, STRtree, Bintree, MonotoneChains, SweepLine
- **Line segment Noding**
  - i.e. find and create all intersections in set of Line Segments
- **Planar Graph framework**
- **Precision Reduction**

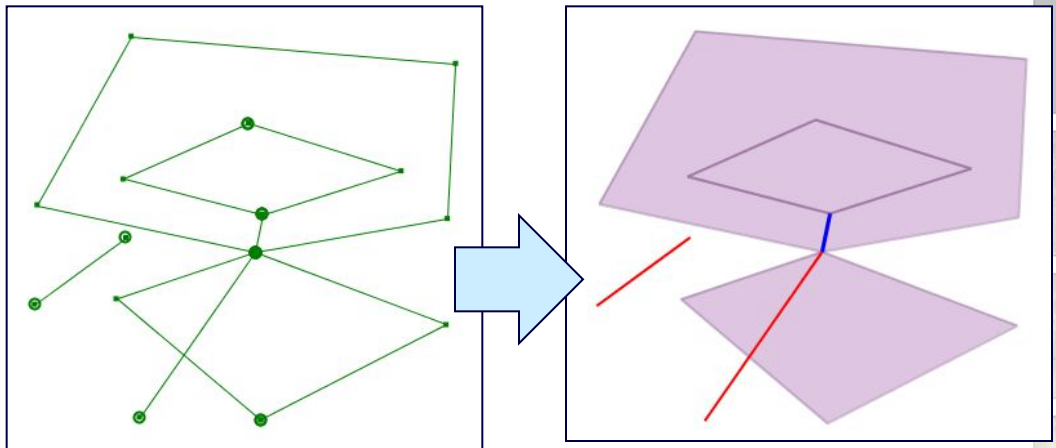


# JTS – Line Merging & Polygonization

- **Line Merging**
  - Removes 2-nodes

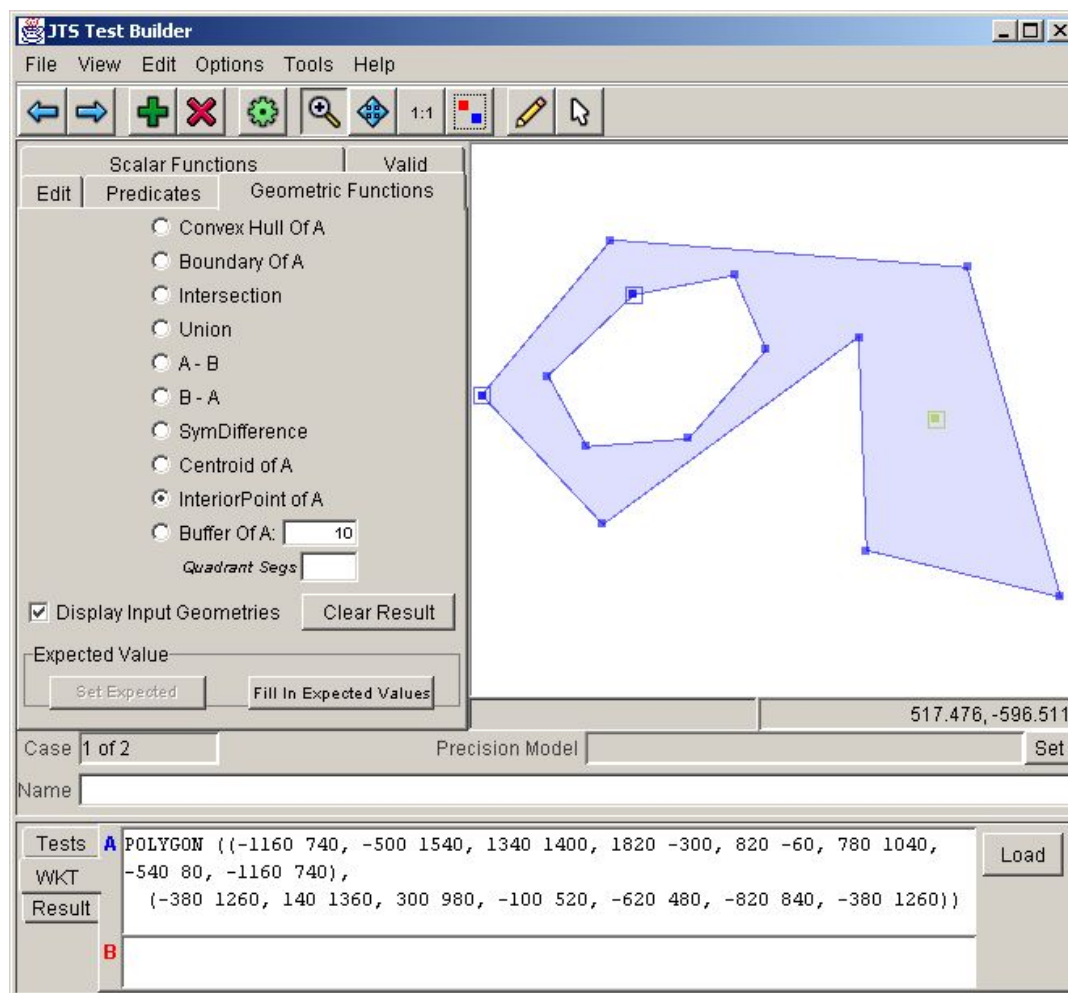


- **Polygonization**
  - Including finding dangles and cutlines



# JTS TestBuilder

- Create/edit/view geometry
- Compute & view results of all JTS methods



# JTS In Use

- **JTS used for core Geometry processing in numerous geo-spatial applications**
  - JUMP
  - BC Gov't Electronic Submission Framework
  - Internet Mapping Framework
  - GeoServer / GeoTools
  - Deegree
  - PostGIS (as GEOS)
  - Tlogica (Bulgaria)
- **Other interesting applications**
  - Font Creator (RobMeek.com)

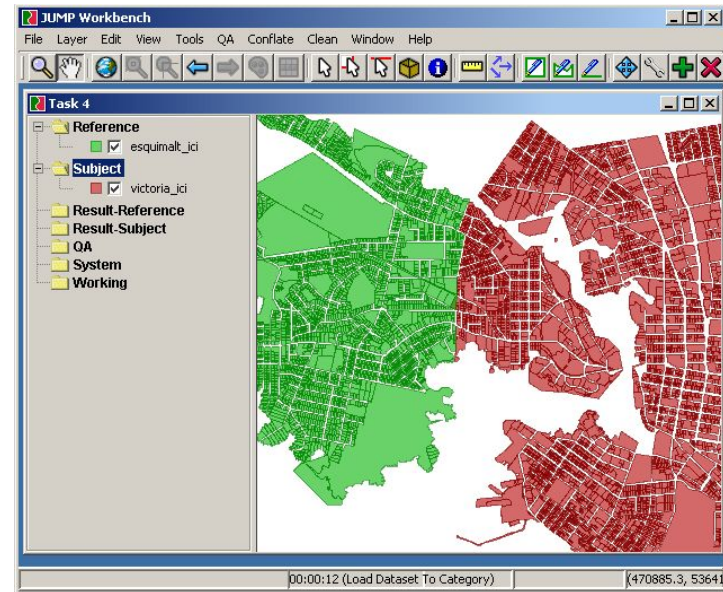
# JTS – Future Work

- **Fully robust Spatial Overlay Operations**
- **Improve performance**
  - e.g. line noding, distance computation
- **Optimize repeated method calls on single Geometry**
  - e.g. “find all geometries which intersect this geometry”
- **Improve internal structure**
- **Geometry simplification / generalization methods**
  - Douglas-Peucker line simplification, etc
- **Distance methods**
  - Hausdorff distance, Frechet distance, etc
- **User-defined Geometry representation**
  - Allows easier adaptation to other Geometry APIs, database structures
- **Affine Transform**
- **Linear Referencing operations**
- **Improved/Extended Spatial Indexes**
  - Updatable Quadtree, R-tree
- **Coverage datatype**



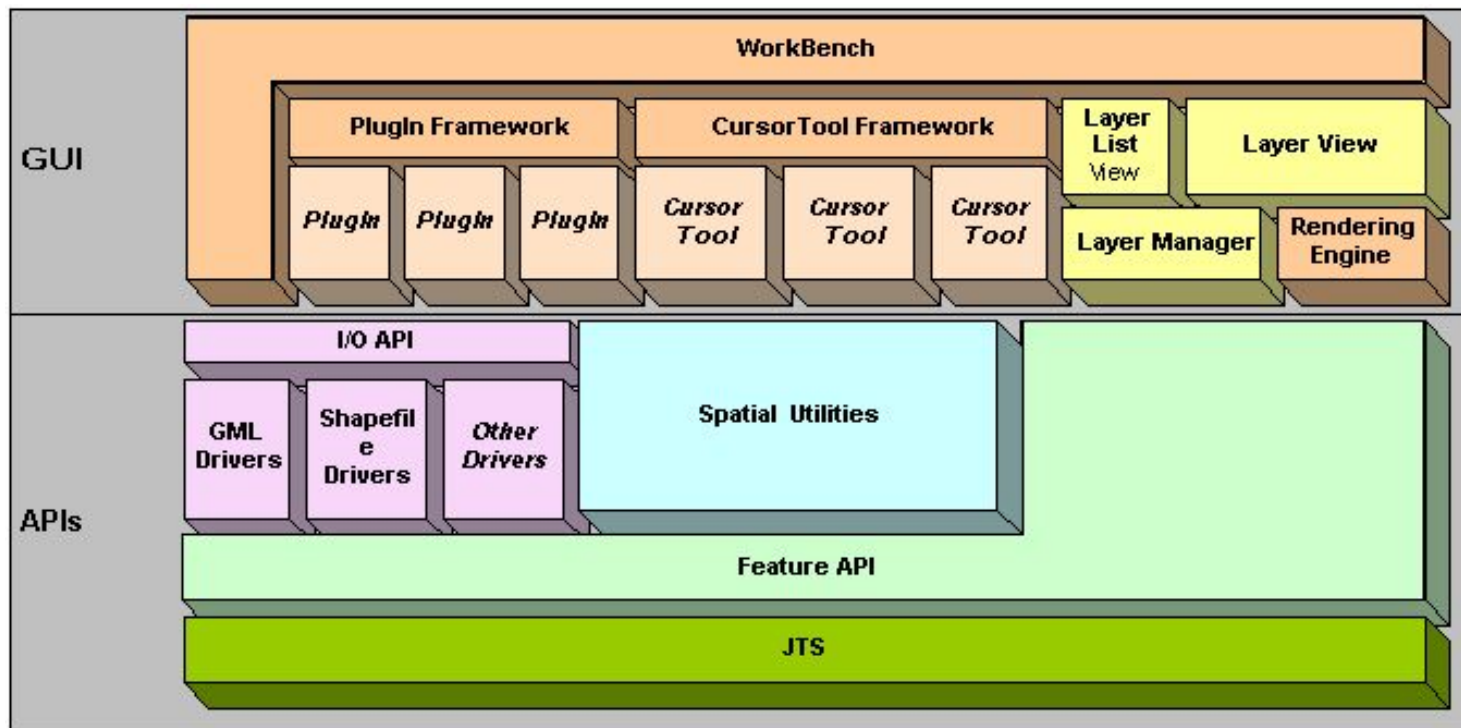
# JUMP Unified Mapping Platform

- 100% pure Java
  - Open Source (*GPL license*)
  - Framework API (*for developers*)
  - Workbench GUI (*for users*)
- 
- Design Features:
    - Rich GUI environment for developing spatial algorithms, visualizing data and output
    - Interactive environment for supporting human-assisted spatial processing
    - Leverage all capabilities of Java platform
    - Highly extensible / reusable



# JUMP - Architecture

- **Highly Extensible**
  - DataSources, Plugins, CursorTools, Renderers
- **Modular, Reusable**
- **Takes full advantage of Java Platform**
  - Dynamic Linking, Java2D graphics, cross-platform, leverages industry standard APIs (e.g. XML)



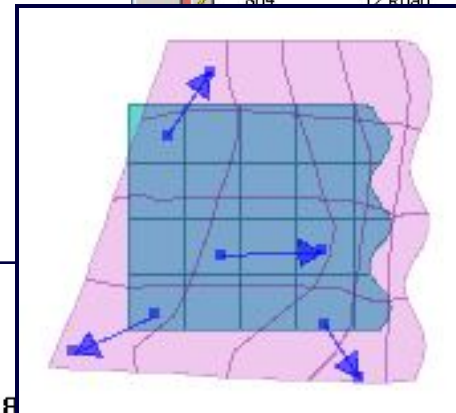
# JUMP – Framework API

- Features with attributes and geometry
- Feature Collections
  - With optional spatial index
- DataSources
  - **File** - Well Known Text, GML, ESRI Shapefile
  - **Spatial DB** - Oracle, PostGIS, ESRI SDE
  - Easy to add new DataSources
- Warping
  - Affine Transform
  - Bilateral Interpolated Triangulation

View Attributes: fernie\_trim

fernietrim (3341 features)

FID	ROAD_ID	CLASS	TYPE
792	0 Road	...	Gravel Road 2 L
793	1 Road	...	Gravel Road 2 L
794	2 Road	...	Rough Road
795	3 Road	...	Gravel Road 1 L
796	4 Road	...	Paved Road 2 L
797	5 Road	...	Gravel Road 2 L
798	6 Road	...	Gravel Road 2 L
799	7 Road	...	Rough Road
800	8 Road	...	Rough Road
801	9 Road	...	Gravel Road 2 L
802	10 Road	...	Paved Road 2 L
803	11 Road	...	Paved Road 2 L
804	12 Road	...	Gravel Road 1 L
			...
			Gravel Road 1 L



```

</FEATURE>
- <FEATURE>
- <GEOMETRY>
- <gml:LineString>
  <gml:coordinates>1697326.4170110226,493927.628378778
  </gml:LineString>
</GEOMETRY>
</FEATURE>
<FEATURE>

```

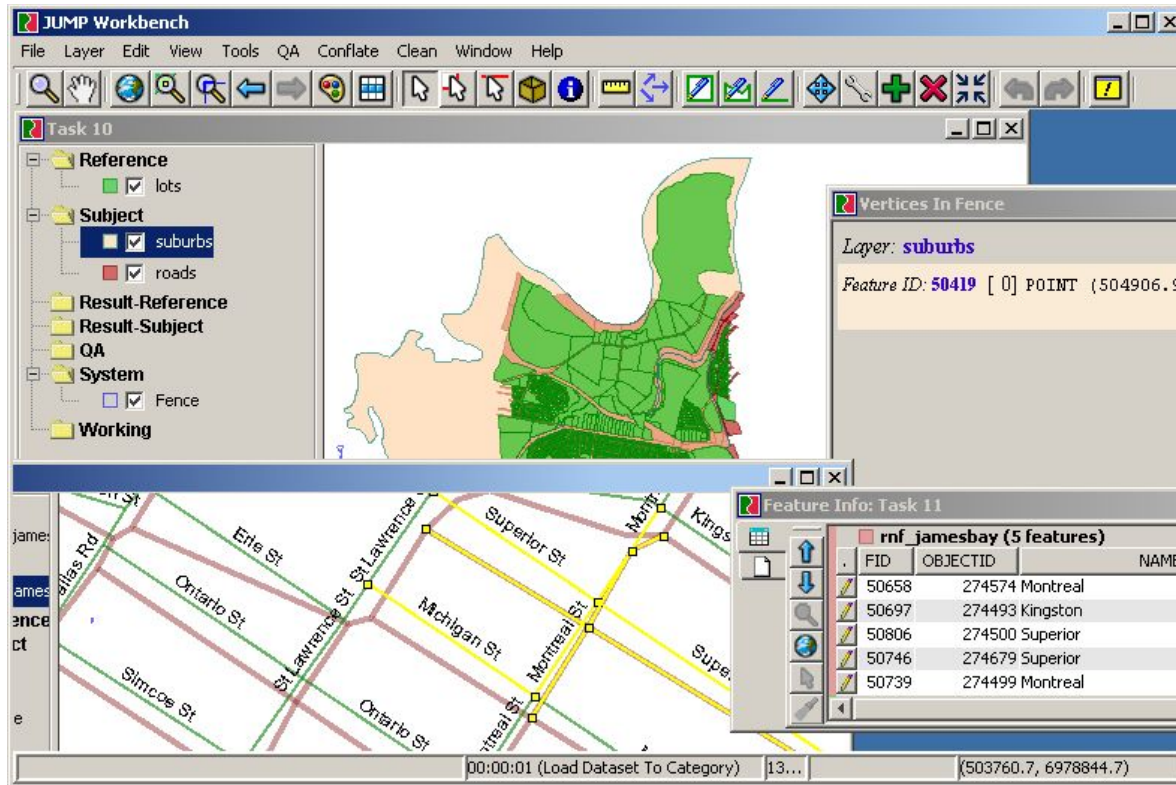
# JUMP & JTS – Demo

- **Load datasets**
- **Zooming / Panning**
- **Symbolization**
  - Line/Fill styles
  - Transparency
  - Labelling
- **WMS Layers**
- **Create Layers / Features / Geometry**
- **Edit Geometry / Attributes**
  - Interactively or Textually
- **Spatial Operations**
  - Overlay, Union, Buffering
- **Polygonization**



# JUMP Workbench

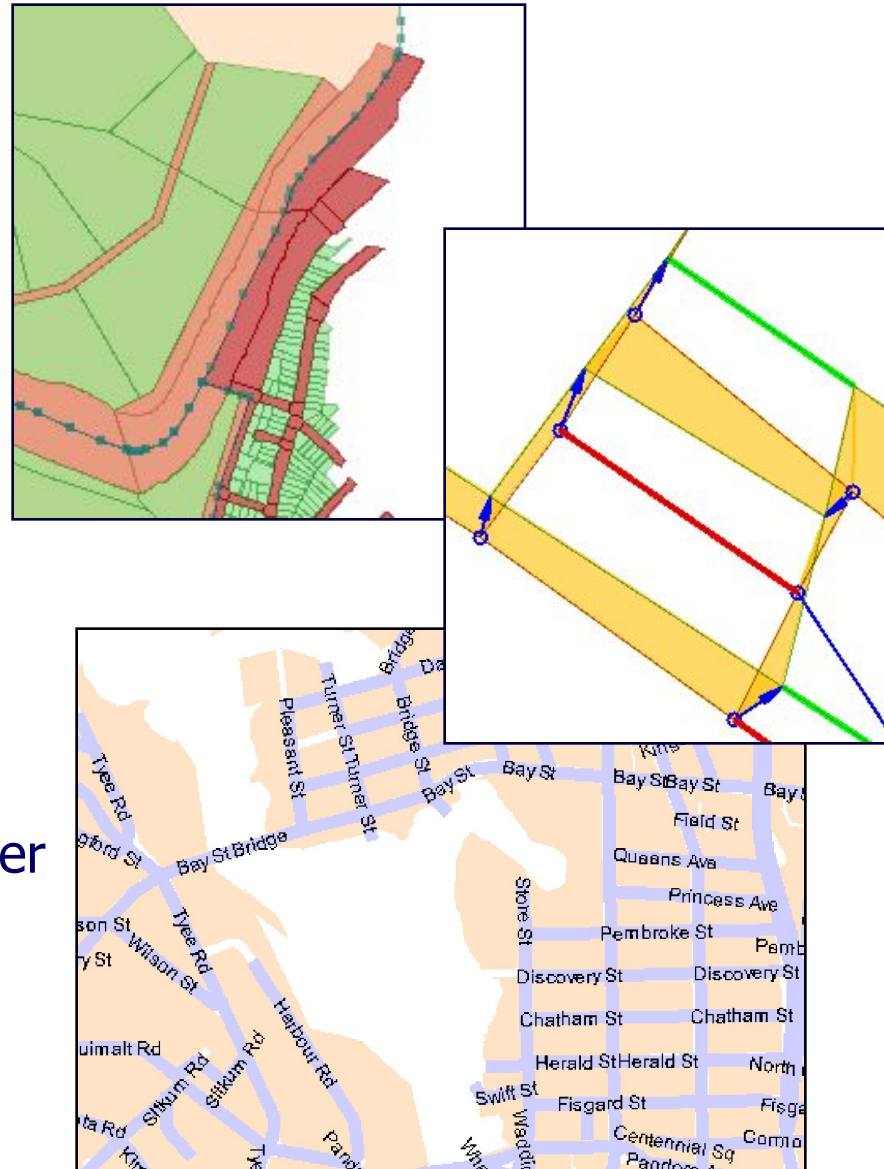
- Multi-Window GUI
- Supports multiple layers of spatial data; rich styling options
- Provides GUI for JUMP API functions
- Geometry & Attribute editing
- Easily extensible via Plugin framework
- Highly modular for easy reusability





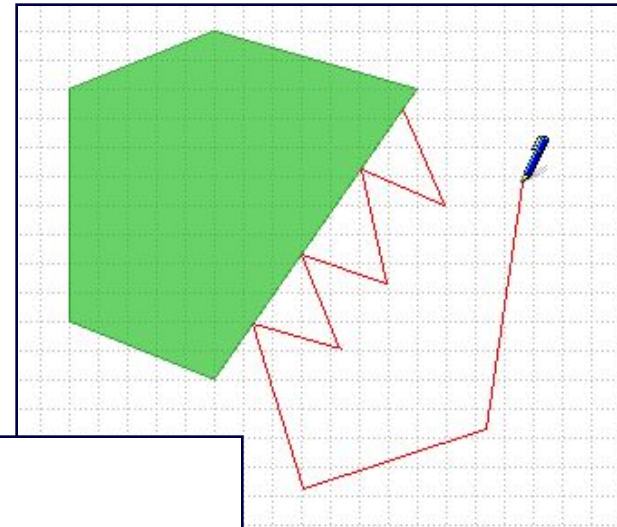
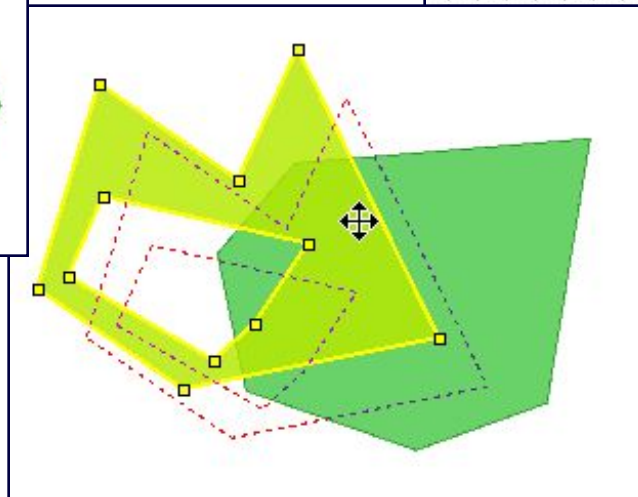
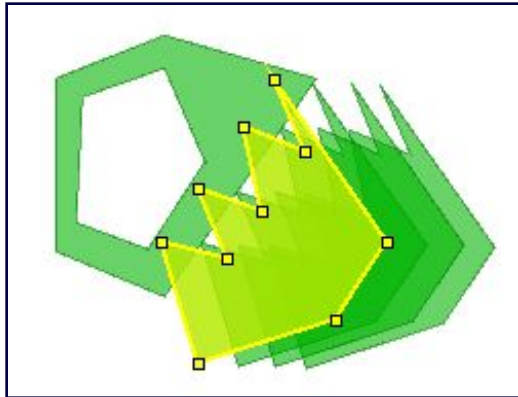
# JUMP - Visualization

- Colour theming
  - Fill / Line colour, size, style
  - Theme by attribute
- Line Styles & Decorations
  - e.g. Dashes, Arrowheads
- Transparency
- Labeling
  - Rotation, scale defined by attribute
  - Scaled / absolute size
  - Collision detection
- Fully customizable via Renderer API



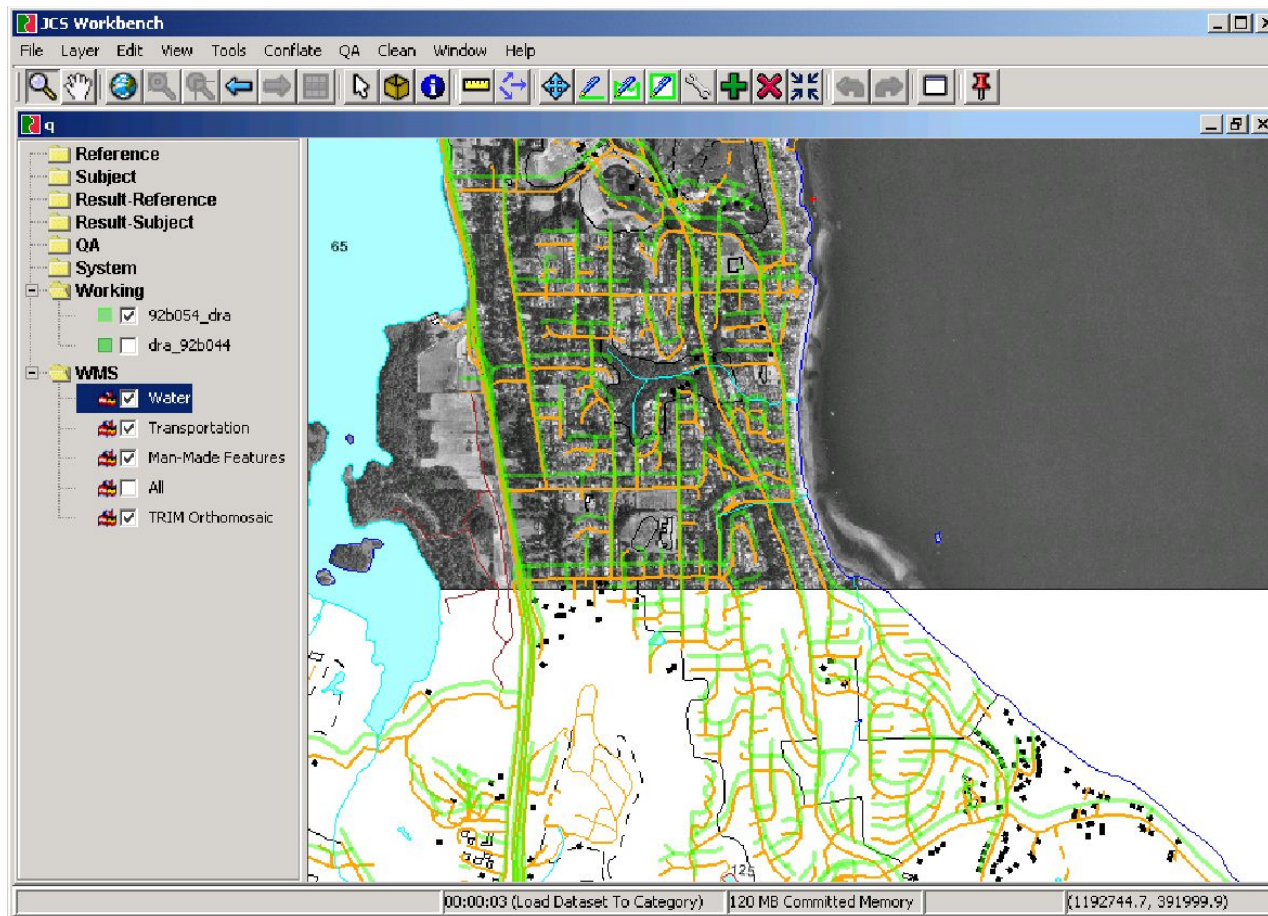
# JUMP – Editing

- Create / Move / Delete Points, Lines, Polygons, Holes
- Combine / Explode to create Geometry Collections
- Snap To Vertex / Line / Grid
- Geometry Validation on Edit
- Multi-Level undo
- Cut / Copy / Paste



# JUMP – Web Map Server Client

- Display images obtained from OGC-compliant Web Map Servers
- Multiple images / servers
- Transparency
- Also exposed as standalone API





# JUMP – Future Work

- **Coordinate Reference System support**
  - Prototype developed; need to improve API
- **Database Connectivity**
  - Live data access (“On-the-fly querying”)
  - Better UI and model for managing database connections
- **Display of geo-referenced imagery**
- **Improved editing tools**
  - Several ideas for richer/easier linework editing tools
- **Harmonization with GeoTools codebase**
  - Start with Coordinate Transformations and DataSources
- **Improve WMS Client**
- **WFS Client**
- **Better Developer documentation**
- **Better cross-platform support, installs**
- *Important to maintain clear, consistent design of UI and codebase!*