

GeomeryFactoryFinder Proposal

Motivation:	Introduce GeometryFactoryFinder and GeometryBuilder API
Contact:	Graham Davis
Tracker:	http://jira.codehaus.org/browse/GEOT-XXXX
Tagline:	Do you know where your geometry is?

This page represents the **current** plan; for discussion please check the tracker link above.

Description

We now have two implementations of ISO19107 Geometry interfaces, and we need to add a FactoryFinder and Builder implementation to make them available as part of the library.

The Geometry interfaces are not hooked into the the GeoTools feature model at this time. As such, it will not break or disrupt other modules on trunk.

- **Geometry Plugin**
The geometry module is an implementation of the GeoAPI interfaces under the ISO 19107 specification. It offers the ability to use objects not found in the JTS Wrapper module, such as curves, surfaces, and multiprimitives. Future work on the module should also add support for 3D geometries. A guide to using this module can be [found here](#).
- **JTS-Wrapper**
An unsupported module donated by SYS Technologies, this work remained on a branch for years and is now available as as beta software.

Status

GeometryFactoryFinder and GeometryBuilder are available on trunk, we need to ask for a vote and accept any remaining feedback.

Voting is open:

- [Andrea Aime](#)
- [Ian Turton](#) +1
- [Justin Deoliveira](#)
- [Jody Garnett](#) +1
- [Martin Desruisseaux](#)
- [Simone Giannecchini](#) +1

API Changes

To make GeometryFactory instances available in GeoTools we need a "Factory Finder":

```
public class GeometryFactoryFinder {
    PositionFactory getPositionFactory(
CoordinateReferenceSystem, Hints );
    GeometryFactory getGeometryFactory(
CoordinateReferenceSystem, Hints );
    ...
}
```

The GeoAPI geometry factories are very complicated to use out of the box. Usually we create a builder to simplify the common use case for a factory, in this case we have created a `GeometryBuilder` to simplify every use.

```

public class GeometryBuilder {
    public GeometryBuilder(
CoordinateReferenceSystem );
    public GeometryBuilder(
CoordinateReferenceSystem, Hints );
    public GeometryBuilder( String code );
    public GeometryBuilder( Hints );

    CoordinateReferenceSystem
getCoordinateReferenceSystem();
    void setCoordinateReferenceSystem(
CoordinateReferenceSystem );

    PointArray createPositionList(
Collection<Position> positions );
    PointArray createPositionList( Object
rest[] );
    ...
}

```

The major "improvement" with the use of a Builder is that you can set the CoordinateReferenceSystem as you go, it will look up the correct factory implementations as needed. Please note that GeometryBuilder cannot implement the Factory interfaces directly (as builders are not stateless or threadsafe).

Code Example

```
GeometryBuilder build = new
GeometryBuilder("EPSG:4326");

Position position = build.createPosition(
0.0, 0.0 );
List<Position> list = build =
createPositionList( position,
build.createPosition(1.0, 0.0) );

DirectPosition here = build.createPosition(
48.44, -123.37 );
Point point = build.createPoint(here);
Point point2 = build.createPoint( 48.44,
-123.37 );
```

Documentation Changes

Website:

- Update [Home](#) with new "library plugin" section
- Updated [Module matrix](#) page
- Updated [geometry status](#) page
- Updated [Geometry Plugin](#) guide