

# Provide Parsers in a consistent fashion

<b>Motivation:</b>	Be consistent about parsing.
<b>Contact:</b>	Jody Garnett
<b>Tracker:</b>	<a href="http://jira.codehaus.org/browse/GEOT-1200">http://jira.codehaus.org/browse/GEOT-1200</a>
<b>Tagline:</b>	lexify this (bad object pun)

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

## Description

We have both some text parsing (CQL and WKT) and XML parsing (filter, gml) incoming. This proposal outlines how we can be consistent in terms of package structure.

- CQL - Catalogue Query Language - currently an unsupported module
- XML - Extensible Markup Language - we are forced to run our own parser due to scaling issues
- WKT - Well Known Text - we already support CRS definition, we need one for Parsing Geometry

How can we be consistent:

1. Make a "default" useful parser available as a utility class
  - CQL
  - CRS.parseWKT
  - JTS.parseWKT
  - Spatial.parseWKT (should be method compatible with JTS utility class)
2. Hide details in a parser package (that casual users do not need to import)
3. Be consistent with package names (not sure how to handle version differences)
  - org.geotools.filter.text.cql2 - parser code for CQL
  - org.geotools.geometry.text.wkt - parser code for WKT
  - org.geotools.filter.xml.filter1\_0 - filter 1.0 bindings (requires gml2)
  - org.geotools.filter.xml.filter1\_1 - Filter 1.1 bindings (requires gml3)
  - org.geotools.geometry.xml.gml2 - geometry bindings for latest GML2
  - org.geotools.geometry.xml.gml3 - geometry bindings for latest GML3

### package names for the gory details

Of the format org.geotools.SUBJECT.PARSER.SPECIFICATION:

- SUBJECT - is the **output** being produced (ie style, geometry, referencing, feature, filter, etc...)
- PARSER - is the kind of "input" being considered (ie text or xml)
- SPECIFICATION - (optional) if you need to get more specific on the kind of "input" you can quote the specification here.

Please note this is for the gory details **only**; your users should not have to import anything from these packages. You may be stuck making some of the content public (especially for XML callbacks) - but none of your example/user code should be forced into an import.



Module breakdown of xml binding plugins is not in scope - we would need a separate proposal.

## Status

Voting in process - closes April 2nd.

Discussed on the email list and in a weekly IRC meeting ( [2007/03/19/IRC Meeting - 19 March 2007](#) ).

Votes are currently being collected:

- [Andrea Aime](#) +1
- [Chris Holmes](#)
- [Ian Turton](#)
- [Jody Garnett](#) +1
- [Martin Desruisseaux](#)
- [Richard Gould](#) +1
- [Simone Giannecchini](#)

It would be nice to have this approved this week so the work can be in the release

this month

dynamictasklist: task list macros declared inside wiki-markup macros are not supported

## Resources

File

Modified ▲

No files shared here yet.

## Tasks

no progress	✓	done	✗	impeded	⚠	lack mandate/funds/time	?	volunteer needed
-------------	---	------	---	---------	---	-------------------------	---	------------------

A target release is also provided for each milestone.

Milestone 1	2.4-M1	
	✓	Move cql to library
	✓	Added org.geotools.geometry.WKTParser
	✓	test and release
	✓	update documentation
Milestone 2	2.5-M1	
		test and release
		update documentation
Milestone 3	2.5.0	
		verify user documentation
		include in release

## API Changes

### BEFORE

This change introduces new API:

#### Example.java

```
public void exampleMethod( ... ){
    Point point = primitiveFactory.createPoint( 32.0, 32.0 );
}
```

### AFTER

## Example.java

```
import org.geotools.geometry.iso.Spatial;

// holds the WKT parser but is not needed by client code
// import org.geotools.geometry.text.wkt

/**
 * Note geometryFactory includes CRS information already
 */
public void exampleMethod( ... ){
    Spatial spatial = new Spatial( geometryFactory, primitiveFactory, positionFactory
);
    Point = (Point) Spatial.parseWKT( "POINT 32 32" );
}
```

## Documentation Changes

Website:

- Update [Upgrade to 2.4](#) instructions (if xml parsing packages are changed)

Developers Guide:

- Update description package convention to describe how additional parsers can be added.

User Guide:

- CQL examples look great!
- CRS examples needed
- Spatial examples needed

User Manual:

- Check with Acuster to see if demo can be updated

Issue Tracker:

- Close jira when completed