

GeoTools Steering Document

Author: Jody Garnett

Review/Revise: Andrea, Justin, Simone, Martin, Paul

Background: The GeoTools library is going through a transition, from a project centered around the research and development of spatial standards, to one used as the foundation for production systems. This document outlines a priorities and goals for this period of transition, allowing us to balance current needs against a realistic schedule.

Project Status

<i>Release</i>	<i>Status</i>	<i>Notes</i>
GeoTools 2.0	archived	no longer relevant to day to day operations
GeoTools 2.1	archived	used in GeoServer 1.3.0
GeoTools 2.2	released	stable target for GeoServer 1.4.x and uDig 1.1,
GeoTools 2.3	code freeze	target for GeoServer 1.5.x WCS
GeoTools 2.4	unstable	
Complex-Features	orphaned	live development continues, waitin on Feature Model to merge

We are also currently hosting to the following Research and Development work.

<i>RnD</i>	<i>Status</i>	<i>Notes</i>
nD Coverages	2.5 research	Support for GeoAPI
Coverage Formats	2.3 devel	Additional Grid Coverage formats against 2.3 api
Go-1 Renderer		Support for Go-1 based rendering
ISO Renderer	2.4 research	Implementing ISO portrayal
uDig Catalog	development	Backport of uDig catalog, used by GeoServer 1.5
uDig Renderer		Backport of uDig renderer framework, to enable ISO Renderer
GTXML	research	Definition of bindings for common xml artifacts
QA FactoryFinder	2.4 QA	Allow GeoTools to function with equinox or spring
QA Factory	2.4 QA	
QA DataStore	2.4 QA	
Hibernate DataStore	2.2 devel	Produce Features from an Object Model
JPOX DataStore	2.4 research	Ability to service up Pojo from a JDO implementation
Feature Model	stalled	for“non flat” features, balance ISO of GML3, stalled on QA
GeoAPI Geometry	stalled	GeoAPI geometry implementation, stalled on feature model

Project Community

One sign of life has always been the number of new developers showing up, over the last year the number of developers has mostly remained flat, which is a cause for concern. We have witnessed a refresh of our core “Project Management Committee” and are operating with a pool of motivated developers.

The roll of the “Module Maintainer” has tapered off the last year, while this is expected as many modules are implementing stable APIs, it has left us with a gap of volunteer and QA effort we previously enjoyed.

- go with what we have, break out modules into a community section that do not meet QA
- seek new modules and development for inclusion in the library

User Community

The GeoTools user community continues to grow, in recent weeks email activity has focused around the use of WFSDataStore. We can surmise that WFS 1.0 is now taking root around the world, and currently we are one of the only toolkits to support the service.

Over the next year I expect to see a growing body of GeoServer and uDig developers asking for support, the challenge will be turn these developers into contributors (accepting patches and providing svn access).

- set up an unsupported svn repository folder with minimal procedure for access
- encourage development of WFS 1.1 client implementation based on “GTXML” is produced
- encourage development of WFS 1.0 client implementation onto new Feature Model

OSGeo

GeoTools is participating in a broader Open Source Geospatial Foundation. We are in the process of completing our incubation requirements (issues of documentation license, and copyright).

We should be able to take part in marketing opportunities, and keep our eye (but not hopes) on OSGeo as a mechanism to collect funding for maintenance.

- collect signed contribution letters from as many contributors as we can find
- produce marketing materials, focus on white papers as benefits a technical project like a toolkit

Standards

GeoTools continues to offer client implementations of available public standards. As we become more popular we are pushed towards ISO standards (which due to cost are difficult to usefully collaborate on). Where possible we will continue to seek guidance of other projects such as GeoAPI.

Research is focused on GML3, ISO Grid Coverage, ISO Geometry, ISO 19115 and ISO 19115, CSW, WFS 1.1, Filter 1.1, and SLD 1.1.

GeoTools RoadMap

The GeoTools timeline is based on community consensus and is mutable to reflect project deadlines.

	<i>GeoTools</i>	<i>GeoAPI</i>	<i>GeoServer</i>	<i>uDig</i>	
Oct	GeoTools 2.2 2.2.1	2.0	1.4.0-RC1	1.1.0	Stable - point release in conjunction with uDig 1.1.0
Oct	GeoTools 2.3 -2.3.0-M0 -2.3.0-RC1 -2.3.0	2.1.M1	1.5-M0 1.5-M1	1.1.1	Stable - GeoSolutions will produce point releases are new grid coverage formats are made available. - Refractions will migrate uDig 1.1.1
Oct Nov Dec	GeoTools 2.4 -2.4.0-M0 -2.4.0-M1 -2.4.0-RC0 -2.4.0	2.1.M2 2.1.M3 2.1.M4		1.2-M0 1.2-M1	Development (QA Mandate) - QA Factory and DataStore use. - “builders” with existing feature model - GridCoverage data access API
Dec Jan	GeoTools 2.5 2.5.0-M0 2.5.0-M1 2.5.0-RC0	2.1.RC0	1.6-M0 1.6-M1 1.6-RC0		Research - switch to simple feature model - switch nD coverage

To interact with the roadmap, for example to scheduling a point release for bug fixes, ask on IRC.

GeoTools Project Deadlines

I am still trying to catch up and balance the above timeline against project goals, here are the **deadlines** (often coming from commercial work based on the listed projects):

- October 2006
 - **GeoServer 1.4.0-RC1** (GeoTools 2.2.1) stable release for commercial work
 - **uDig 1.1** (GeoTools 2.2.1) stable release target for commercial work
 - **GeoServer 1.5.M0** (GeoTools 2.3.0) for additional GC format work
- November 2006
 - **uDig 1.2-M0** (GeoTools 2.4.M2) deadline for JPOX research
- December 2006
 - **GeoServer 1.6-M0** deadline for prototype WFS 1.1
- January 2006
 - **GeoServer 1.6.RC0** deadline for nD coverages

This list is based on deadlines, so if the RnD project you are interested in is not listed it is because either I do not know the deadline, or plan for intergration with the community has not been provided. Work that falls into this category includes Go-1 renderer implementations, any geotools catalog work or feature model work.

GeoTools Project Structure

We need to balance needs of those working on research projects with those working on improving quality of the existing code base. Traditionally GeoTools has operated on a branch-and-merge policy, breaking out an unsupported community section will isolate new research and unsupported plug-ins.

<i>GeoTools Library</i>	<i>GeoTools Plugins</i>	<i>GeoTools Extentions</i>	<i>Unsupported</i>	
referencing	Arcgrid	brewer	arcsde	oraclespatial
api	epsg-access	graph	db2	referencing3D
coverage	epsg-hsql	mappane	directory	tiger
main	epsg-postgresql	openoffice	epsg-ext-esri	vpf
render	epsg-wkt	shaperenderer	geomedia	oracle
sample-data	geotiff	validation	geometryless	go
	gtopo30	widgets-swing	gml	xml
wms	image		hsql	xml-filter
data	imagemosaic		mif	xml-gml2
jdbc	imagepyramid		mysql	xml-gml3
	postgis			jpox
	shapefile			portrayal
	wfs			viewport

The GeoTools Project Management Committee would reduce the release contents to modules meeting the developer guidelines at some initial loss of functionality.

Keeping these modules in svn as unsupported will provide a starting place for those wishing to restore capabilities to the toolkit in the future. Making community modules available as an unsupported download would also be nice visibility for community members.

Please note the above table also represents some book keeping in **bold**:

- combining shape and shapefile, breaking out data and jdbc implementations (as performed on the FM branch), and positioning wms support as part of the library
- creation of new research modules

GeoTools 2.4

The following steps are to be taken as part of the 2.4 development effort.

- Reduction of supported library to working data sources
 - **test:** DataStore Conformance Test
 - **test:** coverage check – developers guide requirement
 - **test:** GeoResourceInfo implementation
 - move other plugins to community section
- Clear support for use of factories at at the datastore level
 - **test:** ability to create POJO from shapefile
- Creation of Grid Coverage data access API
 - **test:** Service and GeoResource handles

The Refractions/uDig focus is on data source quality – specifically:

- Performance suitable for analysis work
 - $O(\log N)$ performance on data source operations
 - $O(1)$ on metadata requests such as bounds
- Event notification
- Storage of common types (boolean, TM_Instant, etc...)

For GeoServer/GeoSolutions the focus is on GridCoverage formats:

- Additional Formats (shared with 2.3.x branch)
- Definition of a API for Grid Coverage discovery and access
 - need to ensure Service/GeoResource handle compatibility

For GeoServer/Social Change Australia (transitional work to revised FM):

- need Factory injection situation cleared up
- delegate to “FeatureBuilder” interface

For TOPP/GeoServer the focus is again on quality:

- QA and consistency to keep the library from falling apart

For Axios/GeoServer/uDig there are a few key fixes:

- Parsing support for common literal types (boolean, temporal, etc)

Un-sponsored goals for 2.4:

- User Guide
- OSGeo Incubation completion
- GTXML QA checks and Documentation requirements for inclusion in 2.4

Resource Allocation

The following table lists interested individuals, that may be made available.

	<i>GeoSolutions</i>	<i>Refractions</i>	<i>TOPP</i>	
2.2.1 2.2.2		support support	support	Refractions is very active in maintain this branch for uDig 1.1 development
2.3.0-M0 2.3.0-RC1 2.3.0 2.3.x	support support devel	support	support support	GeoSolutions will be active on this branch once released, producing new grid coverage plug-ins.
2.4.0-M0 2.4.0-M1 2.4.0-RC0 2.4.0	research research support	devel devel devel support	devel devel support support	Jody - oracle datastore quality Andrea - code reviews, QA support Justin – factory / builder QA Martin – factory / hint QA Simone – Coverage Data Access API
2.5.0-M0 2.5.0-M1 2.5.0-RC0	simone		feature model	Simone – research into nD coverages Jody – feature model support Justin – feature model transition

Release Overhead

The overhead of providing GeoTools releases has traditionally fallen on those projects making use of the library when a release is needed. Given this burden all major associated projects (uDig / GeoServer / WCS) have dodged the issue by packaging up GeoTools in an addhoc manner.

This is something we need to control:

- the release process needs to be stable, and lighten up
- in terms of IP the risk of unversioned jars should not be placed on associated projects

The GeoTools PMC has the mandate to produced releases, here are two options to consider:

- Stewardship: associate PMC member with a branch of interest to their organization:
 - Richard: GeoTools 2.2.x, stable target for uDig 1.1 development
 - Simone: GeoTools 2.3.x, stable target for GeoServer 1.5x and WCS development
 - Jody: GeoTools 2.4.x, target for uDig trunk development
 - Justin: GeoTools 2.5.x, target for GeoServer 1.6 research
- Rotation: rotate release responsibility between PMC members
 - research: schedule milestone release every month
 - development: schedule official releases every quarter
 - production: facilitate creation of point releases as bug-fixes are required by projects