

2007 Q1

[2006 Q4](#)

2007 Q1

[2007 Q2](#)

This report is written in the first quarter of 2007, and outlines community development and research goals.

Users Community

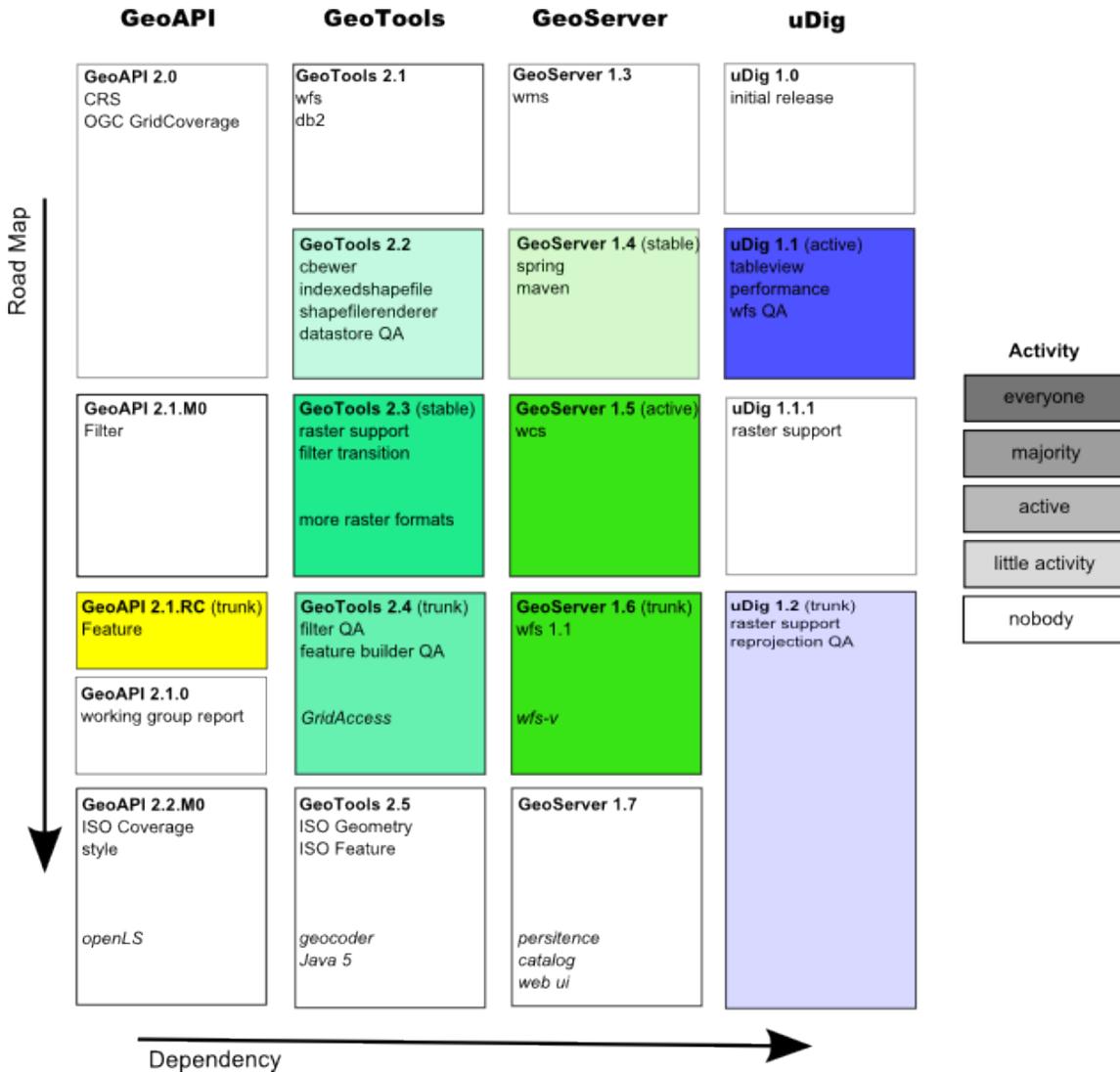
GeoTools has started up production of monthly milestone releases again. The publication of milestone releases makes progress visible and accessible to the user community. We have not been sending announcements out beyond the user list, as milestone releases are strictly for "early access".

- [2.4-M0](#)
- [2.4-M1](#)

We have also updated our [Module Matrix](#) page (and development policies) to reflect how well different modules are supported.

Development

This time around we are balancing between several paid projects (with deadlines) and Martin's presentation to the GeoAPI working group (summer OGC meeting?).



The above diagram is color coded based on project and developer activity (the amount of colour indicates the rate of change).

Development Policies

Since 2006 Q4 the GeoTools project and GeoServer projects have made development policy changes:

- [GeoTools change proposal](#) - Changing an existing API
- [Supporting your module](#) - Adding a new module to GeoTools
- [GSIP 6 - Track GeoTools Trunk](#)

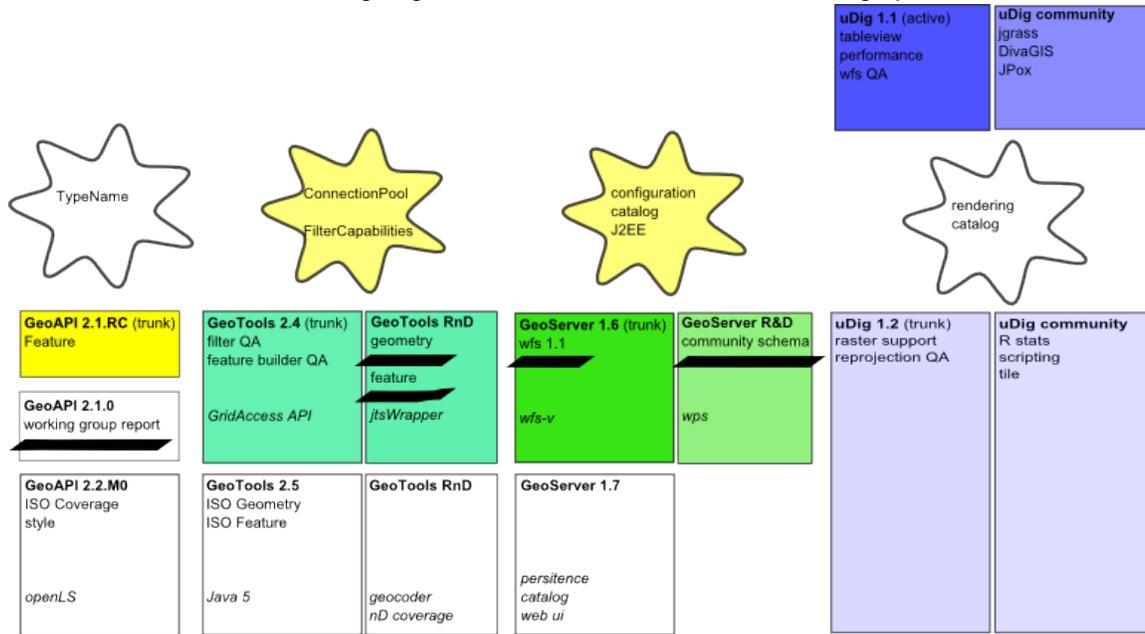
The above policies aim to have a uDig and GeoServer branch available based on GeoTools trunk, without this support we are left with a development community scattered across different versions (all trying to be active).

There is one planned development policy change:

- When a contributor agreement is available (after negotiations with OSGeo) we will require each developer to sign something.

Research Goals

There are several tracks of ongoing research, and a few unresolved design problems:



Research Policies

The GeoTools project and geoserver project made policy changes with respect to new research.

- [Creating your own Module](#)

The establishment of an "unsupported" module space in GeoTools has succeeded in making visible research that was previously "out of sight out of mind". The uDig and GeoServer projects already and community sections meeting this need.

Known Deadlines

In the above diagram the following known deadlines are underlined.

Here are the dates as near as have been made public:

- GeoAPI Report for June 11th (May 11th code freeze)
 - Report to be reviewed during [July 9th Meeting](#) working group meeting
 - The report must be submitted for **June 11th** (ie one month before)
 - Last time this report took a month to prepare so assume a code freeze on **May 11th**
- GeoTools ISO Geometry implementation for Mid March
 - Either jtsWrapper or geometry to be brought up to supported status
 - Module may not be included in download until Java 5 is available

The following are research goals only (no formal release required):

- GeoTools Feature / GeoServer Community schema for Mid March
 - the implementation is being made available as an unsupported module, actual intergration happens later

Outstanding Problems

The following problems are listed as design problems that should be addressed. Often these problems are holding

up existing paid work, but due to the amount of collaboration involved lack specific funding.

- **GeoAPI TypeName** - We are caught between the ISO definition of GenericName (in which the Schema in which the name needs to be known prior to construction) and the need to have a quick "id" when looking up feature content. This is an example of a usability tradeoff .. hard to resolve. The Java class QName is an example of a good compromise (based on the same ISO specification).
 - GeoAPI TypeName interface has been defined in a manner similar to QName, this is in conflict with the interface for GenericName
- **GeoTools ConnectionPool** - GeoTools is starting to be used in more serious J2EE applications, as such we need to use a JNDI look up for our DataSource (Use of an oracle ConnectionPool is most often requested on the user list).
 - We need to define an approach, document it, and roll it out into the database epsg authority plugins and datastores.
- **GeoTools FilterCapabilities** - We have switch over to use GeoAPI Filter interfaces. The interface "Function" does not list the number of required parameters; instead that information is captured as part of the FilterCapabilities information (which we currently do not use).
 - We need to split out the description of available functions from the data structure used to define an expression.
- **GeoServer Persistence** - The ability to persist GeoServers state (currently to an XML file) is hard to maintain, and not extensible.
 - We have several sources of inspiration (using an XML to bean technology), some restrictions (allowing for programatic configuration) and some acceptance tests (ability to add additional configuration elements without changing the object model).
- **GeoServer Catalog** - A catalog is used to manage your resource connections, and allow for additional kinds of resources over time. The existing approach (cut & paste) is not scaling well for GeoServer.
 - We need to bring the GeoTools interfaces up to the level in which they are used by GeoServer and uDig. There are lots of negative examples (on what we don't want to do) and a few good ideas. GeoTools should strive for the minimum common ground.
- **GeoServer J2EE** - GeoServer needs to become a better J2EE application
 - Pay attention to DataSources retrived via JNDI lookup
 - support external configuration and clustering
- **uDig Rendering** - The uDig rendering system has seen enough real world use that we can start to simplify based on experience. Depending on where the line is drawn in the sand we may be able to add OpenGL to our list of targets (currently SWT and AWT are supported).
 - While the work can be backported to GeoTools we need to ensure that there is a good reason (measured in developers) to make the effort worthwhile.
- **uDig Catalog** - The uDig catalog system has seen enough real world use that we can start to simplify based on experience.
 - While the work can be backported to GeoTools we need to ensure that there is a good reason (measured in developers) to make the effort worthwhile.