

# Data Module

Module Maintainer:	<a href="#">Jody Garnett</a> , <a href="#">Justin Deoliveira</a>
Status:	★ ★
Email Help:	Geotools-gt2-users@lists.sourceforge.net
Volunteer:	geotools-devel@lists.sourceforge.net

Data access api providing a common interface to several file formats and data sources. This code was revised for the GeoServer 1.2 release as part of GeoTools 2.0, design documentation and research for this effort is available at <http://vwfs.refractions.net/>.

Ideas from this module were included in GeoAPI 2.0, this module is currently being revised in conjunction with GeoAPI 2.1 as part of the FM branch (see below).

## Data Module Status

### [2.2.x](#)

For the [2.2.x](#) branch the data module remained stable:

- Implementation updated for ResourceCollection
- Implementation provided for FeatureList (a random access api)

### [2.4.x](#)

Broken out into a separate module.

## Feature Model

On the FM branch the following has occurred:

- separated out into a distinct module
- transition to GeoAPI feature model

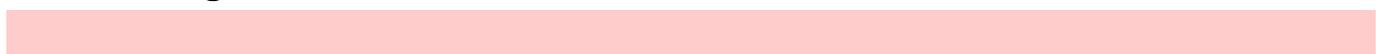
### [2.5.x](#)

Introduced [DataAccess super class for DataStore](#) and parametrized the FeatureSource/Collection/Reader/Writer interfaces to handle GeoAPI Feature as the general case, and SimpleFeature as a specialization.

## Module Status

The data module is in flux as we transition to the FM branch, to protect yourself please use the Query API (and Filter) rather than making direct use of FeatureReader. If you must use FeatureReader

## Outstanding Issues



# Error rendering macro 'jiraissues' : Unable to determine if sort should be enabled.

## Comments

There are several areas in which the data module can be improved:

- support for "joins" has been attempted several times, most recently on the "complex-features" branch, a standards compatible way of defining joins can be seen as part of the Catalog 2 specification in which a Query may hold several filters against different metadata. Starting at the other end of the problem the "community schema" effort concentrates on mapping the joins needed to fulfill a predefined XML schema.
- The exact mechanism for Locking can be improved based on the workflow documented in GeoAPI currently (in which LockResults are returned at the end of a Commit)
- Repository can either be subsumed by Catalog, or can remain as a front end for cross datastore operations (such as client side joins and unlocks). It would make sense to have Repository server as the creator of both Transaction and Locks.
- Coverage support should be accessible in a manner similar to DataStore.