

9.x

The GeoTools 9.x branch is currently being established, as such no technical direction has been set.

- [Usability Targets](#)
 - [API](#)
 - [Deprecation Cycle](#)
 - [Closable Support](#)
 - [Documentation](#)
 - [Simplification](#)
- [Research Targets](#)
 - [Application Schema](#)

RnD and proposals completed for the 9.x branch:

- [FeatureCollection cleanup](#) — remove methods that assume in-memory model
- [Java 7 try-with-resource compatibility](#) — try-with-resource
- [Partial 3D data support](#) — Handling 3d data in a 2d world
- [Replace Contribution Agreement](#) — contemplating contribution
- [Separate general complex feature classes from gt-app-schema](#) —
- [Support for three-dimensional envelopes and bounding box filters](#) —
- [Timed Release](#) — is it time?
- [Vector grids](#)

Attachments:

Name	Size	Creator	Creation Date	Comment
	Drag files from your desktop onto your browser or select files by clicking "Browse"			

Usability Targets

API

We have a couple of drivers for API change, listed are some highlights expected for 9.x development stream.

Deprecation Cycle

We continue to follow our deprecation cycle, allowing downstream projects one full release to migrate away from deprecated methods.

- FeatureCollection - methods assuming an in memory model are being removed

Closable Support

Java 7 is baking in language syntactic sugar around the Closable interface. The development of the java **fore-each** syntax really messed with our FeatureCollection story, the introduction of **try-with-resource** syntax looks to repair that damage.

Documentation

The transition to Sphinx documentation was completed during 8.0 release cycle. This time out we would like to:

fill in the tutorial story to cover more of the library:

- ContentDataStore tutorial is 50% done
- gt-graph is an often requested tutorial

The other bit of work is to continue to update the existing pages with "live" code examples where ever possible.

Simplification

We missed a few opportunities for simplification in 8.x:

- Transaction - make a concrete class to replace DefaultTransaction

Research Targets

Application Schema

One thing this missed during the Application Schema work was the production of classes for us mere mortals working with Java code (rather than xml schema).

So the shopping list for java developers is:

- A FeatureTypeBuilder - that is able to produce application schema compatible feature types.
- FeatureCollection support classes to assist with implementations
- Something like ContentDataStore that can be used as a starting point