Mad Staging Area Idea

Recent email and blog activity have brought out the hardest area to collaborate on: Data formats.

It seems silly that we are stuck on what should be a simple task; and most of the reason we are stuck is over arguments on what makes a good feature model. Enough is Enough! Let's focus on slurping up the data; we won't even bother to put it into a Feature yet. By taking it a step at a time we can manage to work together first ...

These areas are similar to:

- Hibernate UserTypes - heck we could reuse the hibernate staging area
- The AttributeReader design, there was a clean separation between the AttributeReader who read from the file, and the FeatureReader who combined the results into a Feature
- Andrea has talked about populating a cache either "by rows" (ie entire features at a time) or "by columns" ie only grab the attributes needed for the current query

Hibernate and UserTypes

For background (before we talk), have a look at how hibernate hoists stuff into it's a staging area (basically raw objects) before using those objects to create a POJO and populate its fields.

If you look at the hibernate "UserType" you can just see how it is designed as a callback object for the staging area. The "data accessors" would be punting up raw arrays of primitive objects, and this UserType part would be processing them into good objects for the cache.

- http://www.hibernate.org/hib_docs/v3/api/org/hibernate/usertype/UserType.html

We have implemented a UserType for JTS Geometry previously.

Questions:

- can someone find me the hibernate cache information? They use several but ship with one ...
- can someone find me the name of the staging area in hibernate

The BIG Idea

COMPLETE READ

So here is the Workflow:

- FILE or DATABASE FORMAT -> primitives -> USERTYPES -> objects -> STAGING AREA -> features

```
APPLICATION
=========================================
Feature (id| geom, name, age, range )

STAGING AREA
====================================
ID | Geom  | Attrb1 | Attrb2 | Attrb3
------------------------------------
 1 | Point | john   | 24     | Range

RAW FORMAT
=========================================
line | Geom  | A1     | A2     | A3 | A4 |
-----------------------------------------
   1 | bytes |"john  "|"    24"| 1  | 3  |
```

So you can see the raw format combining 1 and 3 into a Range in the staging area. Using the information in the staging area the complete feature can be created.

SPARSE READ

There is a common GeoTools use case that involves doing a "sparse read". When doing rendering we often want ask for "half of a feature" (maybe only the geometry and a single attribute are mentioned in the Style?). It will do us no good to read everything, and stage everything, and then only use half of it

```
APPLICATION
=========================================
Feature (id| geom, null, 24, null )

STAGING AREA
===================================
ID | Geom  | Attrb1 | Attrb2 | Attrb3
-----------------------------------
 1 | Point | null   | 24     | null

RAW FORMAT
=======================================
line | Geom  | A1     | A2     | A3 | A4 |
---------------------------------------
   1 | bytes |"john  "|"    24"| 1  | 3  |
```

In this case we only bothered to read the information needed for the task at hand.