

FeatureCollection Descriptor for FeatureMembers

Motivation:	Ben noticed that our use of Name was inconsistent, we need to fix that up.
Contact:	Jody Garnett
Tracker:	http://jira.codehaus.org/browse/GEOT-XXXX
Tagline:	a rose by any other name? type or instance

This page represents the **current** plan; for discussion please check the tracker link above.

- [Description](#)
- [Status](#)
 - [Tasks](#)
 - [API Changes](#)
 - [Data Access](#)
 - [FeatureSource](#)
 - [FeatureCollection](#)
 - [Documentation Changes](#)

Description

In the [DataAccess super class for DataStore](#) proposal a List<Name> was introduced to DataAccess. It is not clear if these names refer to the content (ie a Descriptor) or to the type (ie FeatureType) - the language used in the javadocs was "names of the available Resources".

This is an aspect of the feature model that was new to the development community and we did not notice this gap until implementation.

Feature Name	Name used to retrieve the content; like an identifier for the data set
Schema Name	Name of the type or schema the content is available in

Unless a datastore is publishing a formal schema defined else where (or has otherwise separated out an information model that they are publishing against) these two names will be the same.

This may be tricky where two tables in a database have the same structure. The table name would show up as the feature name; the description of the structure would be the type name. By sharing a structural description the two tables would be able to reuse Styles definitions.

Status

This proposal is under attack from all sides.

Voting has not started yet:

- [Andrea Aime](#)
- [Ian Turton](#)
- [Justin Deoliveira](#)
- [Jody Garnett](#)
- [Martin Desruisseaux](#)
- [Simone Giannecchini](#)

Tasks

This section is used to make sure your proposal is complete (did you remember documentation?) and has enough paid or volunteer time lined up to be a success

	no progress	✓	done	✗	impeded	⚠	lack mandate /funds/time	?	volunteer needed
--	-------------	---	------	---	---------	---	--------------------------	---	------------------

1. API changed based on BEFORE / AFTER
2. Update DataAccess interface and AbstractDataStore instances
3. Add test cases to each datastore module to verify functionality
4. Update user guide with example code

API Changes

Data Access

BEFORE

```
interface DataAccess<T extends FeatureType,
F extends Feature>{
    /**
     * Names of the available Resources.
     * <p>
     * For additional information please see
     getInfo( Name ) and getSchema( Name ).
     * </p>
     * @return Names of the available
contents.
     * @throws IOException
     */
    List<Name> getNames() throws
```

```
IOException;
```

```
    /**  
     * Description of the named resource.  
     * <p>  
     * The FeatureType returned describes  
the contents being published. For  
     * additional metadata please review  
getInfo( Name ).  
     *  
     * @param name Type name a the resource  
from getNames()  
     * @return Description of the  
FeatureType being made avaiable  
     * @throws IOException  
     */
```

```
    T getSchema(Name name) throws
    IOException;
}
```

AFTER

```
/**
 * Names of the available Resources.
 * <p>
 * For additional information please see
 getDescriptor(Name) and getInfo( Name ).
 * </p>
 * @return Names of the available
resources.
 * @throws IOException
 */
List<Name> getNames() throws
IOException;

/**
 * Description of a named resource.
 * <p>
 * The Descriptor returned describes the
contents being published.
 *
 * @param featureName Name of a resource
from getNames()
 * @return Description of the resource
being made available
 * @throws IOException
 */
```

AttributeDescriptor getDescriptor(Name
featureName) throws IOException;

```
/**
 * Schema of the named resource as a
FeatureType.
 * <p>
 * The FeatureType returned describes
the content being published. For
 * additional metadata please review
getDescriptor( Name ).
 *
 * @param featureName Name of a resource
from getNames()
 * @return Schema of the resource being
made available
 * @throws IOException
 */
T getSchema(Name name) throws
```

```
IOException;  
    ...  
}
```

FeatureSource

BEFORE

AFTER

FeatureCollection

BEFORE

AFTER

```
/** The featureName being published */  
Name getName();
```

Documentation Changes

- [Home](#)