XPath NamespaceSupport

Motivation:	Styling and Query need a hint to understand xpath namespaces
Contact:	Jody Garnett
Tracker:	http://jira.codehaus.org/browse/GEOT-3374
Tagline:	xpath across deep waters

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

- Description
- Status
  - Tasks
  - API Changes
    - FilterFactory2
    - Query
    - PropertyName

Children:

Description

The goal is to enable the following code example that uses xpath namespaces:

```
NamespaceSupport namespaceSupport = new
NamespaceSupport();
    namespaceSupport.declairPrefix("foo",
"urn:cgi:xmlns:CGI:GeoSciML:2.0" );

    FilterFactory2 ff =
CommonFilterFactory.getFilterFactory2( null
);
    Filter filter =
ff.greater(ff.property("foo:city/foo:size",n
amespaceSupport),ff.literal(300000));

    FeatureCollection features =
featureSource.getFeatures( filter );
```

Additionally, the interface for PropertyName should be should allow getting namespace information , especially for duplication purposes:

```
NamespaceSupport namespaceSupport =
propertyName.getNamespaceSupport();
        ...
```

Query should be extended to support the PropertyName type rather than just strings, for greater control of properties returned:

```
Query query = new Query( typeName );
    query.setFilter( filter );
    query.setProperties( new PropertyName[]{
            ff.property("foo:x",
namespaceSupport),
            ff.property("foo:y",
namespaceSupport),

ff.property("foo:description/foo:label",
namespaceSupport)});
```

For backward compatibility, the existing getter and setter with String[] rather than PropertyName[] with remain supported.

Assumption: In the above that description will be returned; with only one the "label" value filled in.

This is a straight up API change to roll in the work of FilterFactoryImplNamespaceAware for broader use. This issue also effects the use of Query (another place where NamespaceSupport is required in order to understand propertyName expressions).

```
public class FilterFactoryImplNamespaceAware
extends FilterFactoryImpl {
    private Hints namespaceHints;
    public FilterFactoryImplNamespaceAware()
{
    }
    public
FilterFactoryImplNamespaceAware(NamespaceSup
port namespaces) {
        setNamepaceContext(namespaces);
    }
    public PropertyName property(String
name) {
        return new
AttributeExpressionImpl(name,
namespaceHints);
    }
    public void
setNamepaceContext(NamespaceSupport
namespaces) {
        namespaceHints = new
Hints(FeaturePropertyAccessorFactory.NAMESPA
CE_CONTEXT, namespaces);
    }
}
```

The above code allows a NamespaceSupport instances provided along side an PropertyName. This can be used to sort out any prefix information mentioned by the PropertyName XPath expression.

Discussion:

- The correct solution (fixing FilterFactory2) is complicated as it is a GeoAPI interface
- Hints are not used in the usual CommonFactoryFinder lookup sense; NAMESPACE_CONTEXT could be

used as part of the factory Hints allowing for some reuse (and probably a memory leak)
- Factory is stateful; which may cause problems as it is common practice to both:
  - ignore a provide factory finder and use common factory finder (common in newer code)
  - capture a filter factory for use later when you need it (common in older code)
- NamespaceSupport is a class; and may appear out of the way for someone issuing a Query to fill in a sax support structure. The alternative interface, NamespaceContext is hard to locate a good non deprecated implementation (gabriel was still looking at the time of writing)

Status

Work has completed; the jira was marked as resolved.

Voting is open:

- Andrea Aime +0.5
- Ben Caradoc-Davies
- Christian Mueller +0
- Ian Turton +0
- Justin Deoliveira +0.5
- Jody Garnett +1
- Michael Bedward +0
- 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. ✅ NC: Update Query based on BEFORE/AFTER
2. ✅ Proposal Refactor OpenGIS to pull geo-api pending into gt-opengis module
3. ✅ Move CommonFactoryFinder to gt-api
4. ✅ NC: Update FilterFactory2 based on BEFORE / AFTER
5. ✅ NC: Migrate app-schema to use FilterFactory2 method
6. ✅ NC: review user documentation for any needed changes

API Changes

FilterFactory2

BEFORE:

```
interface FilterFactory2 extends
FilterFactory {
   ...
   PropertyName property(Name);
   ...
}
```

AFTER:

```
interface FilterFactory2 extends
FilterFactory {
   ...
   PropertyName property(Name name);
   PropertyName property( String xpath,
NamespaceSupport namespaceSupport );
   ...
}
```

Query

BEFORE:

```
public class Query {
   String[] properties;

   ...
   public String[] getPropertyNames() {
        return properties;
   }

   public void setPropertyNames(String[]
propertyNames) {
        properties = propertyNames;
   }

   ...
}
```

AFTER:

```
public class Query {
   ...
   PropertyName[] properties;
   ...
   public String[] getPropertyNames() {
        //convert properties to array
strings
   }

   public void setPropertyNames(String[]
propertyNames) {
        //convert strings to property names
   }
   /** Direct access to properties array */
   public List<PropertyName> getProperties()
{
        return properties
   }
   public void setProperties(
List<PropertyName> properties ) {
        this.properties = properties;
   }
   ...
}
```

PropertyName

Pending the acceptance of Refactor OpenGIS.

BEFORE:

```
public interface PropertyName extends
Expression {
    String getPropertyName();
}
```

AFTER:

```
public interface PropertyName extends
Expression {
    String getPropertyName();
    NamespaceContext getNameSpaceContext();
}
```