

Cleanup Style Interface Deprecations

Motivation:	A previous withdrawn proposal introduced geoapi symbology encoding 1.1 interfaces to geotools.
Contact:	Jody Garnett
Tagline:	Clean up deprecations in style interfaces
Jira:	GEOT-2468 used to report progress

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

- [Description](#)
- [Status](#)
 - [Tasks](#)
 - [API Changes](#)
 - [Current GeoTools 2.5 Interface](#)
 - [AFTER GeoTools 2.5 Change](#)
 - [AFTER: Final GeoAPI 2.6 interface](#)
 - [Documentation Changes](#)

Description

A previous withdrawn proposal, [Upgrading styles with GeoAPI Interfaces](#), introduced GeoAPI interfaces for Symbology Encoding into the GeoTools. This work was incomplete leaving us with many deprecated methods; and no alternative workaround.

This "proposal" (more case of technical dept) will clean up the style interfaces by:

- Introduce read-write GeoTools interfaces for any new Symbology Encoding constructs
- Restoring setters methods; the only deprecations will remain those where a clear alternative is available (example setTitle is now getDescription().setTitle)
- TypeNarrow getters to be explicit about the GeoTools interface returned for any aggregations. Example: getDescription() will return the geotools Description interface
- Completing the StyleFactoryImpl2 implementation; this is an implementation of geoapi style factory where the created results are type-narrowed to explicitly be the GeoTools interfaces
- Updating StyleFactoryImpl to delegate to StyleFactoryImpl2 for the new "create" methods

Good ideas not included in this proposal:

- There is no new builders defined by this proposal
- The renderer will not be updated to account for new Symbology Encoding ideas like UoM
- Symbolizer GeometryPreProcessing will not be added

While the GeoAPI interfaces will be left in the mix as super classes (providing a read-only interface) the GeoTools codebase will continue to make use of the GeoTools interfaces which will offer a read-write extension. This arrangement allows us to use the excellent work that went into producing Symbology Encoding 1.1 interfaces while preserving our current work flow.

Status

This proposal is ready to go; however there is no paid contract work behind it; as such it will need to be a community effort.

I would appreciate a timely response on this as:

- a group is working on UOM support which is one of the features of the SE 1.1 interfaces
- I am putting together a GeoTools tutorial and I cannot do so with the current number of deprecations

This proposal has received the following votes:

- [Andrea Aime](#) +1
- [Ian Turton](#) +0
- [Justin Deoliveira](#) +0
- [Jody Garnett](#) +1
- [Simone Giannecchini](#) +1

Community support:

- jeichar

Tasks

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

1. ✓ Restore Setter methods for 2.5.x; and prepare patch
2. ✓ Apply patch to 2.6.x; and proceed with extending any geoapi classes required (this is a formal review of the geoapi interfaces so some changes to the interfaces will be required - some of the visit method for example are inconsistent)
3. ✓ Go over StyleFactory from geoapi and make sure it is complete; it is complete when it can completely copy an existing style using a StyleVisitor
4. ✓ With 2.6.x cleaned up the documentation examples can be updated; the documentation examples are stored in demo

Unexpected developments:

- Andrea started a StyleBuilder in the "literate" of AttributeTypeBuilder rather than the existing StyleBuilder class which basically will fill in null values for you.
- Clone - the style classes are clonable; but not very well done. I have reports of people using this in the field so I am hard pressed to know how to kill it

API Changes

The following lists one example (ContrastEnhancement); the same approach will be used to update all of the interfaces including the Symbolizers and FeatureTypeStyle.

Current GeoTools 2.5 Interface

Currently all set methods are deprecated; and any method with a planned replacement is also deprecated.

```

public interface ContrastEnhancement {
    @Deprecated // to be replaced by
    ContrastMethod in 2.6
    public void setType(Expression type);

    @Deprecated // setter deprecated
    public Expression getType();

    @Deprecated // setter deprecated
    public void setGammaValue(Expression
gamma);

    public Expression getGammaValue();
    ...
}

```

AFTER GeoTools 2.5 Change

Minimal change will be performed to line up GeoTools 2.5 with intended result; no sense deprecating a method that is going to be maintained. We also need to restore any method for which the replacement is not ready (getType in this example).

```

public interface ContrastEnhancement extends
org.opengis.style.ContrastEnhancement{
    public void setType(Literal type);
    /**
     * Returns a literal expression (one of
    NORMALIZE, HISTOGRAM, NONE)
     * indicating which ContrastMethod value
    is to be used.
     *

```

```
    * @deprecated Please use getMethod
    */
    public Literal getType();

    /**
     * @param gamma How much to brighten
     (greater than 1) or dim (less than 1) this
     channel; use 1.0 to indicate no change.
     */
    public void setGammaValue(Expression
gamma);

    /**
     * How much to brighten (values greater
     than 1.0) or dim (values less than 1.0) an
     image. The default GammaValue is 1.0
     * (no change).
     *
     * @return Expression, if
     <code>>null</code> a value of 1.0 is assumed
     indicating no change
```

```

    */
    public Expression getGammaValue();
}

```

AFTER: Final GeoAPI 2.6 interface

Any geoapi interfaces that need to be extended (or type narrowed) are sorted out. Any overrides are supplemented with our own javadoc (getters) and a mutability allowed (addition of a setter method). Any honest mistakes (like `getType` and `setType`) remain deprecated.

```

public interface ContrastEnhancement extends
org.opengis.style.ContrastEnhancement{
    /**
     * Contrast method used to enhance this
     channel.
     * @param method ContrastMethod used to
     enhance this channel
     */
    public void setMethod( ContrastMethod
method );

    public ContrastMethod getMethod();

    @Deprecated
    public void setType(Literal type);

    /**
     * Returns a literal expression (one of
    NORMALIZE, HISTOGRAM, NONE)
     * indicating which ContrastMethod value
    is to be used.
     *

```

```
    * @deprecated Please use getMethod
    */
    @Deprecated
    public Literal getType();

    /**
     * @param gamma How much to brighten
     (greater than 1) or dim (less than 1) this
     channel; use 1.0 to indicate no change.
     */
    public void setGammaValue(Expression
gamma);

    /**
     * How much to brighten (values greater
     than 1.0) or dim (values less than 1.0) an
     image. The default GammaValue is 1.0
     * (no change).
     *
     * @return Expression, if
     <code>null</code> a value of 1.0 is assumed
     indicating no change
     */
```

```
public Expression getGammaValue();  
}
```

Documentation Changes

The rendering and style section of the user guide can be updated after this work is complete.