ReferencingFactoryFinder incompatible change

Motivation:	Fix the "URN factory not found" bug
Contact:	Martin Desruisseaux
Tracker:	http://jira.codehaus.org/browse/GEOT-1659

This proposal has been withdrawn
Implementation was attempted and we realized that it leads to confusion. It was not always obvious which `ReferencingFactoryFinder` instance should be used, `STRICT` or `DEFAULT`. The risk for greater confusion on user side was high. We selected an other approach which preserve backward compatibility, at the cost of being less explicit and more "magic".

Description

Initial investigation suggests that GEOT-1659 is caused by global (system-wide) hints. Sometime the referencing module really wants a factory for a specific set of hints, without merge with global hints.

The incompatible change proposal:

- Turns every `ReferencingFactoryFinder` static methods into non-static methods.

Note that only `ReferencingFactoryFinder` (and maybe other `FactoryFinders` if we want to uniformize this approach) would be affected. The `CRS` convenience class would not be affected.

If this proposal is accepted, I can declare two `ReferencingFactoryFinder` instances: `DEFAULT` and `STRICT`. Users would need to replace every calls to:

```
ReferencingFactoryFinder.getFooFactory(...)
```

by:

```
ReferencingFactoryFinder.DEFAULT.getFooFactory(...)
```

We can do that with a search-and-replace tool like we did for `Logging` (actually it is easier than `Logging`, since there is no import statement or class name to change).

Transition path

`ReferencingFactoryFinder` has been introduced on 2.4, which is not yet released. On the 2.4 branch, the deprecated `FactoryFinder` class would stay as-is: static methods delegating to `ReferencingFactoryFinder.DEFAULT`. So the "deprecate" cycle would be respected, but very late since it is so close to 2.4 release.

Status

This proposal is currently underconstruction:

- Andrea Aime
- Chris Holmes
- Ian Turton
- Jody Garnett
- Martin Desruisseaux
- Richard Gould
- Simone Giannecchini

Tasks

1. Fix up ReferencingFactoryFinder with new methods.
2. Writes a Ant task performing global search-and-replace. This is a very trivial and low-risk replacement.
3. Update GeoTools library to use new methods using the Ant task.
4. Update the Referencing User docs: 07 Referencing

API Change

BEFORE

```
class ReferencingFactoryFinder {
   ReferencingFactoryFinder() {
        // singleton
   }

   public static Set<CRSAuthorityFactory> getCRSAuthorityFactories(Hints hints);

   public static CRSAuthorityFactory getCRSAuthorityFactory(String authority, final
Hints  hints);
   ...
   public static synchronized void addAuthorityFactory(AuthorityFactory authority);
}
```

Client code using global settings with ReferencingFactoryFinder:

```
CRSAuthorityFactory epsgAuthority =
ReferencingFactoryFinder.getCRSAuthorityFactory("EPSG", null);
CRSAuthorityFactory autoAuthority =
ReferencingFactoryFinder.getCRSAuthorityFactory("AUTO", null);
```

GeoTools code using `ReferencingFactoryFinder` internally:

```
Hints hints = new Hints(Hints.FORCE_LONGITUDE_FIRST_AXIS_ORDER, Boolean.FALSE);
CRSAuthorityFactory crsAuthority =
ReferencingFactoryFinder.getCRSAuthorityFactory("URN", hints);
```

AFTER

```
class ReferencingFactoryFinder {
   /** The default factory finder instance. User-provided hints are merged with global
hints. */
   public static final ReferencingFactoryFinder DEFAULT = new
ReferencingFactoryFinder(null, true);

   /** The factory finder instance that do not merge user hints with global hints. */
   public static final ReferencingFactoryFinder STRICT = new
ReferencingFactoryFinder(null, false);

   /** Hints used by create methods, or null if none. */
   final Hints hints;

   /** true if global hints should be merged with user-specified hints. */
   final boolean merge;

   /** Creates a factory finder with the specified hints. */
   public ReferencingFactoryFinder(Hints hints, boolean merge) {
      this.hints = new Hints(hints);
      this.merge = merge;
   }

    /** Returns all providers of the specified type. */
    private static Set<Factory> getFactories(Class type, Hints userHints) {
        Hints mergedHints;
        if (merge) {
            mergedHints = GeoTools.addDefaultHints(this.hints);
        } else {
            mergedHints = this.hints;
        }
        mergedHints.addAll(userHints);
        return new ... (mergedHints);
    }

   public Set<CRSAuthorityFactory> getCRSAuthorityFactories(Hints hints) {
       return getFactories(CRSAuthorityFactory.class, hints);
   }

   public CRSAuthorityFactory getCRSAuthorityFactory(String authority, Hints hints) {
       return getAuthorityFactory(CRSAuthorityFactory.class, authority, hints,
Hints.CRS_AUTHORITY_FACTORY);
   }
   ...
   // non create methods remain unchanged
   public static synchronized void addAuthorityFactory(final AuthorityFactory
authority);
}
```

Client code using global settings with `ReferencingFactoryFinder`:

```
CRSAuthorityFactory epsgAuthority =
ReferencingFactoryFinder.DEFAULT.getCRSAuthorityFactory("EPSG");
CRSAuthorityFactory autoAuthority =
ReferencingFactoryFinder.DEFAULT.getCRSAuthorityFactory("AUTO", null);
```

GeoTools code using `ReferencingFactoryFinder` instance:

```
Hints hints = new Hints(Hints.FORCE_AXIS_ORDER_HONORING, Boolean.TRUE);
CRSAuthorityFactory crsAuthority =
ReferencingFactoryFinder.STRICT.getCRSAuthorityFactory("EPSG");
```