Change Log for JSR-245 JavaServer^{TM} Pages 2.1

The purpose for the proposed changes to JSR-245, JavaServer^{TM} Pages 2.1 Specification (JSP 2.1) is to take the expression language (EL) out of JSP 2.1 and make it a self-contained specification. The next version of the JSP specification will remove the description of the expression language and will instead reference and depend on the EL specification.

This change log has been created according to the Maintenance step of the JCP process, and will be posted for a 30 day public review before being declared final.

Proposed Changes

1. Version

The EL that is part of JSP 2.1 will be EL 1.0, and the proposed standalone EL will be EL 1.1.

2. New Methods in EL 1.1

To make EL 1.1 self-contained, two factory methods have been added.

2.1 javax.el.ExpressionFactory.newInstance()

Creates a new instance of a `ExpressionFactory`. This method uses the following ordered lookup procedure to determine the `ExpressionFactory` implementation class to load:

- Use the Services API (as detailed in the JAR specification). If a resource with the name of `META-INF/services/javax.el.ExpressionFactory` exists, then its first line, if present, is used as the UTF-8 encoded name of the implementation class.
- Use the properties file "lib/el.properties" in the JRE directory. If this file exists and it is readable by the `java.util.Properties.load(InputStream)` method, and it contains an entry whose key is "javax.el.ExpressionFactory", then the value of that entry is used as the name of the implementation class.
- Use the `javax.el.ExpressionFactory` system property. If a system property with this name is defined, then its value is used as the name of the implementation class.
- Use a platform default implementation.

2.2 javax.el.ExpressionFactory.newInstance(Properties)

Create a new instance of a `ExpressionFactory`, with optional properties. This method uses the same lookup procedure as the one used in `newInstance()`.

If the argument `properties` is not null, and if the implementation contains a constructor with a single parameter of type `java.util.Properties`, then the constructor is used to create the instance.

Properties are optional and can be ignored by an implementation.

The name of a property should start with "javax.el."

The following are some suggested names for properties.

- javax.el.cacheSize