MBeanRegistrationAspect

Purpose

The purpose of this aspect is to allow you to easily *expose* your objects as JMX mbeans.

Usage

In order to be able to use this aspect, you need to define two pointcuts: one specifying which classes to transform (named **mbeansClasses**) as **DynamicMBean** and **NotificationEmitter** and another one (named **mbeansToRegister**) indicating when should the registration of your objects as managed beans should be done.
Usually you use a *class* pointcut for **mbeansClasses** and a *method* pointcut for **mbeansToRegister** in order to intercept constructor calls to the classes you picked in **mbeansClasses**.

Classes you would like to weave need a XML file describing the JMX constructors, operations, attributes and notifications you'd like to expose. This aspect for each class weaved (let's say class **foo.bar.C**) looks for a resource named with the name of the class suffixed by **-mbean.xml** (so **C-mbean.xml** for example) in the same package (so **foo.bar**) as the one of the class.

Here is a sample class definition file:

```
<mbean description="Dummy Object"
type="org.codehaus.aware.jmx.DummyObject">
  <constructor name="default"
description="default constructor"/>
  <constructor name="default_int_string"
description="constructor with int and
string">
    <parameter name="IntProperty"
description="int property" type="int"/>
    <parameter name="StringProperty"
description="string property"
type="java.lang.String"/>
  </constructor>
  <attribute name="IntProperty"
description="int property" type="int"/>
  <attribute name="StringProperty"
description="string property"
type="java.lang.String"/>
  <operation name="doNothing"
description="Do nothing" returnType="void"/>
  <operation name="doNothingWithAString"
description="Do nothing with a string"
returnType="void">
    <parameter name="aString" description="a
string" type="java.lang.String"/>
  </operation>
  <notification
name="javax.management.Notification"
description="Attribute change notification"
type="org.codehaus.aware.jmx.DummyObject"/>
</mbean>
```

Here is a sample *aspectwerkz.xml* file:

```
<aspectwerkz>
  <system id="aspectwerkz-aware">
    <package name="org.codehaus.aware.jmx">
      <aspect
class="MBeanRegistrationAspect">
        <param name="domain" value="test"/>
        <pointcut name="mbeansClasses"
expression="class(org.codehaus.aware.jmx.Dum
myObject)"/>
        <pointcut name="mbeansToRegister"
expression="execution(org.codehaus.aware.jmx
.DummyObject.new(..))"/>
      </aspect>
    </package>
  </system>
</aspectwerkz>
```

See Also

MBeanNotificationAspect.