什么是属性编辑器，作用？   
\* 自定义属性编辑器，spring配置文件中的字符串转换成相应的对象进行注入   
spring已经有内置的属性编辑器，我们可以根据需求自己定义属性编辑器   
  
\* 如何定义属性编辑器？   
\* 继承PropertyEditorSupport类，覆写setAsText()方法，参见：UtilDatePropertyEditor.java   
\* 将属性编辑器注册到spring中，参见：applicationContext.xml   
  
比如：   
有一个类里面有一个Date属性

1. **public** **class** Bean1 {
2. **private** Date dateValue;
3. **public** **void** setDateValue(Date dateValue) {
4. **this**.dateValue = dateValue;
5. }
6. }

applicationContext.xml配置文件如下：

1. <!--将bean1中的Date赋值2008-08-15，spring会认为2008-08-15是String，无法转换成Date，会报错！-->
2. <bean id="bean1" **class**="com.bjsxt.spring.Bean1">
3. <property name="dateValue">
4. <value>2008-08-15</value>
5. </property>
6. </bean>
7. <!-- 于是定义属性编辑器 -->
8. <bean id="customEditorConfigurer" **class**="org.springframework.beans.factory.config.CustomEditorConfigurer">
9. <property name="customEditors">
10. <map>
11. <!—key Date,value是个bean，com.bjsxt.spring.UtilDatePropertyEditor ，其中bean com.bjsxt.spring.UtilDatePropertyEditor 的属性有format，注意写法，当map entry的value是bean的时候，就是下面这样-->
12. <entry key="java.util.Date">
13. <bean **class**="com.bjsxt.spring.UtilDatePropertyEditor">
14. <!--干脆把format也注入，灵活处理格式-->
15. <property name="format" value="yyyy-MM-dd"/>
16. </bean>
17. </entry>
18. </map>
19. </property>
20. </bean>

UtilDatePropertyEditor.java 如下，必须继承java.beans.PropertyEditorSupport类，覆写setAsText()方法

1. **import** java.beans.PropertyEditorSupport;
2. **import** java.text.ParseException;
3. **import** java.text.SimpleDateFormat;
4. **import** java.util.Date;
6. /\*\*
7. \* java.util.Date属性编辑器
8. \* @author Administrator
9. \*
10. \*/
11. **public** **class** UtilDatePropertyEditor **extends** PropertyEditorSupport {
13. **private** String format="yyyy-MM-dd";
15. @Override
16. **public** **void** setAsText(String text) **throws** IllegalArgumentException {
17. System.out.println("UtilDatePropertyEditor.saveAsText() -- text=" + text);
19. SimpleDateFormat sdf = **new** SimpleDateFormat(format);
20. **try** {
21. Date d = sdf.parse(text);
22. **this**.setValue(d);
23. } **catch** (ParseException e) {
24. e.printStackTrace();
25. }
26. }
28. **public** **void** setFormat(String format) {
29. **this**.format = format;
30. }
32. }

这样就可以完成正确解析了，注意customEditors是Spring的类CustomEditorConfigurer提供的属性，是一个Map,里面存放的都是自定义的编辑器（customEditors），比如这里存放的是UtilDatePropertyEditor日期编辑器，看CustomEditorConfigurer源码就知道了。   
  
测试一下：

1. **import** org.springframework.beans.factory.BeanFactory;
2. **import** org.springframework.context.support.ClassPathXmlApplicationContext;
4. **import** junit.framework.TestCase;
6. **public** **class** InjectionTest **extends** TestCase {
8. **private** BeanFactory factory;
10. @Override
11. **protected** **void** setUp() **throws** Exception {
12. factory = **new** ClassPathXmlApplicationContext("applicationContext.xml");
13. }
15. **public** **void** testInjection1() {
16. Bean1 bean1 = (Bean1)factory.getBean("bean1");
17. System.out.println("bean1.dateValue=" + bean1.getDateValue());
18. }
20. }

能打印出日期就OK了。