[Spring中ApplicationContext和beanfactory区别](http://blog.csdn.net/hi_kevin/article/details/7325554)

|  |
| --- |
| BeanFacotry是spring中比较原始的Factory。如XMLBeanFactory就是一种典型的BeanFactory。原始的BeanFactory无法支持spring的许多插件，如AOP功能、Web应用等。    ApplicationContext接口,它由BeanFactory接口派生而来，因而提供BeanFactory所有的功能。ApplicationContext以一种更向面向框架的方式工作以及对上下文进行分层和实现继承，ApplicationContext包还提供了以下的功能：    • MessageSource, 提供国际化的消息访问     • 资源访问，如URL和文件     • 事件传播     • 载入多个（有继承关系）上下文 ，使得每一个上下文都专注于一个特定的层次，比如应用的web层   1.利用MessageSource进行国际化     BeanFactory是不支持国际化功能的，因为BeanFactory没有扩展Spring中MessageResource接口。相反，由于ApplicationContext扩展了MessageResource接口，因而具有消息处理的能力(i18N)，具体spring如何使用国际化，以后章节会详细描述。   2.强大的事件机制(Event)     基本上牵涉到事件(Event)方面的设计，就离不开观察者模式。不明白观察者模式的朋友，最好上网了解下。因为，这种模式在java开发中是比较常用的，又是比较重要的。  ApplicationContext的事件机制主要通过ApplicationEvent和ApplicationListener这两个接口来提供的，和java swing中的事件机制一样。即当ApplicationContext中发布一个事件的时，所有扩展了ApplicationListener的Bean都将会接受到这个事件，并进行相应的处理。   Spring提供了部分内置事件，主要有以下几种：   ContextRefreshedEvent ：ApplicationContext发送该事件时，表示该容器中所有的Bean都已经被装载完成，此ApplicationContext已就绪可用  ContextStartedEvent：生命周期 beans的启动信号   ContextStoppedEvent: 生命周期 beans的停止信号   ContextClosedEvent：ApplicationContext关闭事件，则context不能刷新和重启，从而所有的singleton bean全部销毁(因为singleton bean是存在容器缓存中的)     虽然，spring提供了许多内置事件，但用户也可根据自己需要来扩展spriong中的事物。注意，要扩展的事件都要实现ApplicationEvent接口。    3.底层资源的访问     ApplicationContext扩展了ResourceLoader(资源加载器)接口，从而可以用来加载多个Resource，而BeanFactory是没有扩展ResourceLoader   4.对Web应用的支持     与BeanFactory通常以编程的方式被创建不同的是，ApplicationContext能以声明的方式创建，如使用ContextLoader。当然你也可以使用ApplicationContext的实现之一来以编程的方式创建ApplicationContext实例 。    ContextLoader有两个实现：ContextLoaderListener和ContextLoaderServlet。它们两个有着同样的功能，除了listener不能在Servlet 2.2兼容的容器中使用。自从Servelt 2.4规范，listener被要求在web应用启动后初始化。很多2.3兼容的容器已经实现了这个特性。使用哪一个取决于你自己，但是如果所有的条件都一样，你大概会更喜欢ContextLoaderListener；关于兼容方面的更多信息可以参照ContextLoaderServlet的JavaDoc。  这个listener需要检查contextConfigLocation参数。如果不存在的话，它将默认使用/WEB-INF/applicationContext.xml。如果它*存在*，它就会用预先定义的分隔符（逗号，分号和空格）分开分割字符串，并将这些值作为应用上下文将要搜索的位置。ContextLoaderServlet可以用来替换ContextLoaderListener。这个servlet像listener那样使用contextConfigLocation参数。  5.其它区别     1).BeanFactroy采用的是延迟加载形式来注入Bean的，即只有在使用到某个Bean时(调用getBean())，才对该Bean进行加载实例化，这样，我们就不能发现一些存在的Spring的配置问题。而ApplicationContext则相反，它是在容器启动时，一次性创建了所有的Bean。这样，在容器启动时，我们就可以发现Spring中存在的配置错误。     2).BeanFactory和ApplicationContext都支持BeanPostProcessor、BeanFactoryPostProcessor的使用，但两者之间的区别是：BeanFactory需要手动注册，而ApplicationContext则是自动注册 |