

# Spring Framework Reference Card

**NOTE:** This is a quick summary of popular features. Many more classes/methods exist. Complete documentations is at <http://www.springframework.org>

## Factory

org.springframework.beans.factory

### XmlBeanFactory(FileSystemResource(File/String))

```
getBean(String beanName)
containsBean(String beanName)
getType(String beanName) returns Class
isSingleton(String beanName)
isTypeMatch((String beanName, class)
```

Bean factory implementations should support the standard bean lifecycle interfaces as far as possible. The full set of initialization methods and their standard order is:

1. BeanNameAware's setBeanName
2. BeanClassLoaderAware's setBeanClassLoader
3. BeanFactoryAware's setBeanFactory
4. ResourceLoaderAware's setResourceLoader (only applicable when running in an application context)
5. ApplicationEventPublisherAware's setApplicationEventPublisher (only applicable when running in an application context)
6. MessageSourceAware's setMessageSource (only applicable when running in an application context)
7. ApplicationContextAware's setApplicationContext (only applicable when running in an application context)
8. ServletContextAware's setServletContext (only applicable when running in a web application context)
9. postProcessBeforeInitialization methods of BeanPostProcessors
10. InitializingBean's afterPropertiesSet
11. a custom init-method definition
12. postProcessAfterInitialization methods of BeanPostProcessors

On shutdown of a bean factory, the following lifecycle methods apply:

1. DisposableBean's destroy
2. a custom destroy-method definition

## ApplicationContext

org.springframework.context.support

### ClassPathXmlApplicationContext(String pathname)

### FileSystemXmlApplicationContext(String pathname)

Also supports array of pathnames

### XmlWebApplicationContext

The default location for the root context is "/WEB-INF/applicationContext.xml", and "/WEB-INF/test-servlet.xml" for a context with the namespace "test-servlet" (like for a DispatcherServlet instance with the servlet-name "test").

## ApplicationContext (Continued)

In **web.xml**:

```
<context-param id="spring-beans">
  <description>
    This is a comma-separated list of Spring bean definition files. The Spring web dispatcher servlet(s) are not configured here, since they are inferred by the servlet by constructing a filename using the servlet's logical name as a basis.
  </description>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    /WEB-INF/applicationContext.xml</param-value>
</context-param>
```

## Beans

```
<beans [default-autowire=]>
  <bean name="" class="" [singleton="true"] [autowire=]>
    <property name="name">value/ref</property>
  </bean>
</beans>
```

**Note:** Simple elements may be collapsed into their parents.

## Value/Reference

```
<value>value</value>
<ref bean="beanname"/>
<null/> Null value
```

## Collections

```
<list>[value/ref]*</list>
<set> same as list
<map>[<entry key="name">value/ref</entry>]*</map>
<property> same as map
```

## Construction

```
<constructor-arg [index="n"]
  [type="class"]>value/ref</constructor-arg>
```

Repeat for each constructor argument used to build bean.

## Property Editor Support

### Built-in

org.springframework.beans.propertyeditors

**ClassEditor** – String classname

**CustomDateEditor** – DateFormat

**FileEditor** – String pathname

**LocaleEditor** – String localename

**StringArrayPropertyEditor** – csv to String[]

**StringTrimmerEditor** – removes spaces, sets "" to null.

## Autowiring

**autowire**=byName,byType,constructor,autodetect

Autodetect tries **constructor**, then **byType**

## Postprocessing

### BeanPostProcessor

org.springframework.beans.factory.config

```
public Object postProcessAfterInitialization(
  Object bean, String beanName)
  throws BeanException;
```

```
public Object postProcessBeforeInitialization(
  Object bean, String beanName)
  throws BeanException;
```

### BeanFactoryPostProcessor

org.springframework.beans.factory.config

```
Public void postProcessBeanFactory(
  ConfigurableListableBeanFactory)
  throws BeansException;
```

## Special Beans

org.springframework.beans.factory.config

### PropertyPlaceholderConfigurer

**property:** pathname value or <list> of pathname values

Supports ant-style (\${name}) substitution

## Event Listeners

org.springframework.context.ApplicationEvent[C]

### ContextClosedEvent

### ContextRefreshedEvent

**RequestHandledEvent** (web application context)

## Awareness

org.springframework.beans.factory

### BeanNameAware

```
setBeanName(String)
```

### BeanFactoryAware

### ApplicationContextAware

## JEE and JPA Support

### JNDI DataSource

```
<bean id="myDataSource"
  class="org.springframework.jndi.JndiObjectFactoryBean">
  <property name="jndiName"
    value="java:comp/env/jdbc/myDS"/>
</bean>
```

OR

```
<jee:jndi-lookup id="myDataSource"
  jndi-name="jdbc/myDS"/>
```

### Apache commons-dbc Data Source

```
<bean id="myDataSource"
  class="org.apache.commons.dbcp.BasicDataSource"
  destroy-method="close"
  >
  <property name="driverClassName"
    value="org.hsqldb.jdbcDriver" />
  <property name="url"
    value="jdbc:hsqldb:sample" />
  <property name="username" value="myself" />
  <property name="password" value="secret" />
</bean>
```

### JPA EntityManagerFactory

```
<bean id="entityManagerFactory"
  class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean"
  >
  <property name="dataSource" ref="myDataSource" />
  <property name="persistenceXmlLocation"
    value="/WEB-INF/classes/META-INF/persistence.xml" />
  <property name="loadTimeWeaver"><!-- Tomcat 6 -->
  <bean
    class="org.springframework.instrument.classloading.InstrumentationLoadTimeWeaver" />
  </property>
</bean>
```

**NOTE:** Requires spring-agent and spring-tomcat-weaver in TOMCAT\_HOME/lib and Tomcat startup with agent. Exact details are subject to change with improvements in Tomcat.

### JPA Annotation Support

```
<bean
  class="org.springframework.orm.jpa.support.PersistenceAnnotationBeanPostProcessor" />
```

### JPA TransactionManager

```
<bean id="transactionManager"
  class="org.springframework.orm.jpa.JpaTransactionManager">
  <property name="entityManagerFactory" ref="entityManagerFactory" />
</bean>
```

**JTA version:**

```
org.springframework.orm.jta.JtaTransactionManager
```

### Annotation support

```
<tx:annotation-driven />
```

## AOP

```
<!-- the transactional advice (what 'happens'; see the <aop:advisor /> bean
below) -->
<tx:advice id="txAdvice" transaction-manager="txManager">
  <!-- the transactional semantics... -->
  <tx:attributes>
    <!-- all methods starting with 'get ' are read-only -->
    <tx:method name="get" read-only="true"/>
    <!-- other methods use the default transaction settings (see below) -->
    <tx:method name="*" />
  </tx:attributes>
</tx:advice>
```

```
<!-- ensure that the above transactional advice runs for any execution
of an operation defined by the FooService interface -->
<aop:config>
  <aop:pointcut id="fooServiceOperation" expression="execution(*
x.y.service.FooService.*(..))"/>
  <aop:advisor advice-ref="txAdvice" pointcut-ref="fooServiceOperation"/>
</aop:config>
```

## XML Namespaces:

### Spring Version 2.5

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:aop="http://www.springframework.org/schema/aop"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xmlns:oxm="http://www.springframework.org/schema/oxm"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
    http://www.springframework.org/schema/tx
    http://www.springframework.org/schema/tx/spring-tx-2.0.xsd
    http://www.springframework.org/schema/aop
    http://www.springframework.org/schema/aop/spring-aop-2.0.xsd
    http://www.springframework.org/schema/oxm
    http://www.springframework.org/schema/oxm/spring-oxm-1.5.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-2.5.xsd"
  >
```