spring.jar是包含有完整发布的单个jar 包，spring.jar中包含除了spring-mock.jar里所包含的内容外其它所有jar包的内容，因为只有在开发环境下才会用到 spring-mock.jar来进行辅助测试，正式应用系统中是用不得这些类的。

除了spring.jar文件，Spring还包括有其它13个独立的jar包，各自包含着对应的Spring组件，用户可以根据自己的需要来选择组合自己的jar包，而不必引入整个spring.jar的所有类文件。

(1) spring-core.jar

　　这个jar文件包含Spring框架基本的核心工具类，Spring其它组件要都要使用到这个包里的类，是其它组件的基本核心，当然你也可以在自己的应用系统中使用这些工具类。

(2) spring-beans.jar

　　这个jar文件是所有应用都要用到的，它包含访问配置文件、创建和管理bean以及进行Inversion of Control / Dependency Injection（IoC/DI）操作相关的所有类。如果应用只需基本的IoC/DI支持，引入spring-core.jar及spring-beans.jar文件就可以了。

(3) spring-aop.jar

　　这个jar文件包含在应用中使用Spring的AOP特性时所需的类。使用基于AOP的Spring特性，如声明型事务管理（Declarative Transaction Management），也要在应用里包含这个jar包。

(4) spring-context.jar

　　这个jar文件为Spring核心提供了大量扩展。可以找到使用Spring ApplicationContext特性时所需的全部类，JDNI所需的全部类，UI方面的用来与模板（Templating）引擎如Velocity、FreeMarker、JasperReports集成的类，以及校验Validation方面的相关类。

(5) spring-dao.jar

　　这个jar文件包含Spring DAO、Spring Transaction进行数据访问的所有类。为了使用声明型事务支持，还需在自己的应用里包含spring-aop.jar。

(6) spring-hibernate.jar

　　这个jar文件包含Spring对Hibernate 2及Hibernate 3进行封装的所有类。

(7) spring-jdbc.jar

　　这个jar文件包含对Spring对JDBC数据访问进行封装的所有类。

(8) spring-orm.jar

　　这个jar文件包含Spring对DAO特性集进行了扩展，使其支持 iBATIS、JDO、OJB、TopLink，因为Hibernate已经独立成包了，现在不包含在这个包里了。这个jar文件里大部分的类都要依赖spring-dao.jar里的类，用这个包时你需要同时包含spring-dao.jar包。

(9) spring-remoting.jar

　　这个jar文件包含支持EJB、JMS、远程调用Remoting（RMI、Hessian、Burlap、Http Invoker、JAX-RPC）方面的类。

(10) spring-support.jar

　　这个jar文件包含支持缓存Cache（ehcache）、JCA、JMX、邮件服务（Java Mail、COS Mail）、任务计划Scheduling（Timer、Quartz）方面的类。

(11) spring-web.jar

　　这个jar文件包含Web应用开发时，用到Spring框架时所需的核心类，包括自动载入WebApplicationContext特性的类、Struts与JSF集成类、文件上传的支持类、Filter类和大量工具辅助类。

(12) spring-webmvc.jar

　　这个jar文件包含Spring MVC框架相关的所有类。包含国际化、标签、Theme、视图展现的FreeMarker、JasperReports、Tiles、Velocity、XSLT相关类。当然，如果你的应用使用了独立的MVC框架，则无需这个JAR文件里的任何类。

(13) spring-mock.jar

　　这个jar文件包含Spring一整套mock类来辅助应用的测试。Spring测试套件使用了其中大量mock类，这样测试就更加简单。模拟HttpServletRequest和HttpServletResponse类在Web应用单元测试是很方便的。

如何选择这些发布包，决定选用哪些发布包其实相当简单。

如果你正在构建Web应用并将全程使用Spring，那么最好就使用单个全部的spring.jar文件；

如果你的应用仅仅用到简单的Inversion of Control / Dependency Injection（IoC/DI）容器，那么只需spring-core.jar与spring-beans.jar即可；

如果你对发布的大小要求很高，那么就得精挑细选了，只取包含自己所需特性的jar文件了。

采用独立的发布包你可以避免包含自己的应用不需要的全部类。当然你可以采用其它的一些工具来设法令整个应用包变小，节省空间的重点在于准确地找出自己所需的Spring依赖类，然后合并所需的类与包就可以了。

Eclispe有个插件叫ClassPath Helper可以帮你找找所依赖的类。

Spring包依赖说明:

1) spring-core.jar需commons-collections.jar，spring-core.jar是以下其它各个的基本。

2) spring-beans.jar需spring-core.jar，cglib-nodep-2.1\_3.jar

3) spring-aop.jar需spring-core.jar，spring-beans.jar，cglib-nodep-2.1\_3.jar，aopalliance.jar

4) spring-context.jar需spring-core.jar，spring-beans.jar，spring-aop.jar，commons-collections.jar，aopalliance.jar

5) spring-dao.jar需spring-core.jar，spring-beans.jar，spring-aop.jar，spring-context.jar

6) spring-jdbc.jar需spring-core.jar，spring-beans.jar，spring-dao.jar

7) spring-web.jar需spring-core.jar，spring-beans.jar，spring-context.jar

8) spring-webmvc.jar需spring-core.jar/spring-beans.jar/spring-context.jar/spring-web.jar

9) spring-hibernate.jar需spring-core.jar，spring- beans.jar，spring-aop.jar，spring-dao.jar，spring-jdbc.jar，spring- orm.jar，spring-web.jar，spring-webmvc.jar

10) spring-orm.jar需spring-core.jar，spring-beans.jar，spring-aop.jar，spring-dao.jar，spring-jdbc.jar，spring-web.jar，spring-webmvc.jar

11) spring-remoting.jar需spring-core.jar，spring-beans.jar，spring- aop.jar，spring-dao.jar，spring-context.jar，spring-web.jar，spring- webmvc.jar

12) spring-support.jar需spring-core.jar，spring-beans.jar，spring-aop.jar，spring-dao.jar，spring-context.jar，spring-jdbc.jar

13) spring-mock.jar需spring-core.jar，spring-beans.jar，spring-dao.jar，spring-context.jar，spring-jdbc.jar

一、只是使用[spring](http://lib.csdn.net/base/javaee" \o "Java EE知识库" \t "_blank)框架  
dist\spring.jar  
lib\jakarta-commons\commons-logging.jar

如果使用到了切面编程(AOP)，还需要下列jar文件  
lib\aspectj\aspectjweaver.jsr 和 aspectjrt.jar  
lib\cglib\cglib-nodep-2.1\_3.jar

如果使用了JSR-250中的注解如@Resource/@PostConstruct/@PreDestroy  
还需下列jar文件  
lib\j2ee\common-annotations.jar

二、只是使用spring框架若使用注解方式  
只要加一些命名空间和开启解析器  
并且@Resource需要加lib\j2ee\common-annotations.jar  
<context:annotation-config/>打开处理器

三、要自动扫描  
只要开启解析器和一些命名空间  
<context:component-scan base-package=""/>

四、spring框架采用aop编程  
需要导入一些命名空间  
xmlns:aop.......  
<aop:aspectj-autoproxy/>开启解析器  
如果使用到了切面编程(AOP)，还需要下列jar文件  
lib\aspectj\aspectjweaver.jsr 和 aspectjrt.jar  
lib\cglib\cglib-nodep-2.1\_3.jar

五、spring+Jdbc开发  
需要数据源文件 lib\jakarta-commons\commons-pool.jar、lib\jakarta-commons\commons-dbcp.jar  
1、spring文件 ：

dist\spring.jar、  
 lib\jakarta-commons\commons-logging.jar  
2、jdbc驱动文件 [MySQL](http://lib.csdn.net/base/mysql)\*\*\*\* sql\*\*\* ：  
导入tx命名空间  
<tx:annotation-driven transaction-manager=""/>  
对事务注解的解析器

六、spring2.5+hibernate3.3+struts1.3

（1）、[hibernate](http://lib.csdn.net/base/javaee" \o "Java EE知识库" \t "_blank)核心包

hibernate3.jar  
lib\required\\*.jar  
lib\optional\ehcache-1.2.3.jar (二级缓存文件)  
hibernate注解包  
lib\test\slf4j-log4j12.jar

（2）、spring包  
dist\spring.jar  
lib\jakarta-commons\commons-logging.jar 和commons-pool.jar、commons-dbcp.jar(后为数据源文件)  
dist\modules\spring-webmvc-struts.jar (与struts1的集成)  
lib\aspectj\aspectjweaver.jsr 和 aspectjrt.jar (Aop文件)  
lib\cglib\cglib-nodep-2.1\_3.jar     (Aop文件)  
lib\j2ee\common-annotations.jar     (注解文件)  
lib\log4j\log4j-1.2.15.jar

（3）、sturs1.3.8  
建议将jstl-1.0.2.jar和standard-1.0.2.jar文件更换为1.1版本 此时JSTL文件  
还有spring已存在antlr-2.7.6.jar文件所以将struts中的antlr-2.7.6.jar文件删除以免冲突

[数据库](http://lib.csdn.net/base/mysql)驱动  
mysql\*\*\* sql\*\*\*\*根据需求换

如果在web容器实例spring容器  
加文件到web.xml......

如果action让容器管理，则path(Struts的配置文件)和name(Spring配置文件)要一致  
还必须在struts的配置文件里加一个控制器

spring为struts1.3解决乱码问题  
在web.xml配置一个filter

七、spring2.5+hibernate3.3+struts2整合开发

struts2文件lib里面所有不带-plugin结尾的jar文件，但除了struts2-spring-plugin-2.0.11.1.jar

http://docs.spring.io/spring-framework/docs/current/spring-framework-reference/html/overview.html#overview-modules

| **GroupId** | **ArtifactId** | **Description** |
| --- | --- | --- |
| org.springframework | spring-aop | Proxy-based AOP support |
| org.springframework | spring-aspects | AspectJ based aspects |
| org.springframework | spring-beans | Beans support, including Groovy |
| org.springframework | spring-context | Application context runtime, including scheduling and remoting abstractions |
| org.springframework | spring-context-support | Support classes for integrating common third-party libraries into a Spring application context |
| org.springframework | spring-core | Core utilities, used by many other Spring modules |
| org.springframework | spring-expression | Spring Expression Language (SpEL) |
| org.springframework | spring-instrument | Instrumentation agent for JVM bootstrapping |
| org.springframework | spring-instrument-tomcat | Instrumentation agent for Tomcat |
| org.springframework | spring-jdbc | JDBC support package, including DataSource setup and JDBC access support |
| org.springframework | spring-jms | JMS support package, including helper classes to send and receive JMS messages |
| org.springframework | spring-messaging | Support for messaging architectures and protocols |
| org.springframework | spring-orm | Object/Relational Mapping, including JPA and Hibernate support |
| org.springframework | spring-oxm | Object/XML Mapping |
| org.springframework | spring-test | Support for unit testing and integration testing Spring components |
| org.springframework | spring-tx | Transaction infrastructure, including DAO support and JCA integration |
| org.springframework | spring-web | Web support packages, including client and web remoting |
| org.springframework | spring-webmvc | REST Web Services and model-view-controller implementation for web applications |
| org.springframework | spring-webmvc-portlet | MVC implementation to be used in a Portlet environment |
| org.springframework | spring-websocket | WebSocket and SockJS implementations, including STOMP support |