Spring的配置里，为什么最好不要配置xsd文件的版本号

**XML的一些概念**

首先来看下xml的一些概念：

xml的schema里有namespace，可以给它起个别名。比如常见的spring的namespace：

**[html]** [view plain](http://blog.csdn.net/hengyunabc/article/details/22295749) [copy](http://blog.csdn.net/hengyunabc/article/details/22295749)

[在CODE上查看代码片](https://code.csdn.net/snippets/264875)

1. xmlns:mvc="http://www.springframework.org/schema/mvc"
2. xmlns:context="http://www.springframework.org/schema/context"

通常情况下，namespace对应的URI是一个存放XSD的地址，尽管规范没有这么要求。如果没有提供schemaLocation，那么Spring的XML解析器会从namespace的URI里加载XSD文件。我们可以把配置文件改成这个样子，也是可以正常工作的：

**[html]** [view plain](http://blog.csdn.net/hengyunabc/article/details/22295749) [copy](http://blog.csdn.net/hengyunabc/article/details/22295749)

[在CODE上查看代码片](https://code.csdn.net/snippets/264875)

1. **<?xml** version="1.0" encoding="UTF-8"**?>**
2. **<beans** xmlns="http://www.springframework.org/schema/beans/spring-beans.xsd"
3. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"**>**

schemaLocation提供了一个xml namespace到对应的XSD文件的一个映射，所以我们可以看到，在xsi:schemaLocation后面配置的字符串都是成对的，前面的是namespace的URI，后面是xsd文件的URI。比如：

**[html]** [view plain](http://blog.csdn.net/hengyunabc/article/details/22295749) [copy](http://blog.csdn.net/hengyunabc/article/details/22295749)

[在CODE上查看代码片](https://code.csdn.net/snippets/264875)

1. xsi:schemaLocation="http://www.springframework.org/schema/beans
2. http://www.springframework.org/schema/beans/spring-beans.xsd
3. http://www.springframework.org/schema/security
4. http://www.springframework.org/schema/security/spring-security.xsd"

**Spring是如何校验XML的**

Spring默认在启动时是要加载XSD文件来验证xml文件的，所以如果有的时候断网了，或者一些开源软件切换域名，那么就很容易碰到应用启动不了。我记得当时Oracle收购Sun公司时，遇到过这个情况。

为了防止这种情况，Spring提供了一种机制，默认从本地加载XSD文件。打开spring-context-3.2.0.RELEASE.jar，可以看到里面有两个特别的文件：

spring.handlers

**[plain]** [view plain](http://blog.csdn.net/hengyunabc/article/details/22295749) [copy](http://blog.csdn.net/hengyunabc/article/details/22295749)

[在CODE上查看代码片](https://code.csdn.net/snippets/264875)

1. http\://www.springframework.org/schema/context=org.springframework.context.config.ContextNamespaceHandler
2. http\://www.springframework.org/schema/jee=org.springframework.ejb.config.JeeNamespaceHandler
3. http\://www.springframework.org/schema/lang=org.springframework.scripting.config.LangNamespaceHandler
4. http\://www.springframework.org/schema/task=org.springframework.scheduling.config.TaskNamespaceHandler
5. http\://www.springframework.org/schema/cache=org.springframework.cache.config.CacheNamespaceHandler

spring.schemas

**[plain]** [view plain](http://blog.csdn.net/hengyunabc/article/details/22295749) [copy](http://blog.csdn.net/hengyunabc/article/details/22295749)

[在CODE上查看代码片](https://code.csdn.net/snippets/264875)

1. http\://www.springframework.org/schema/context/spring-context-2.5.xsd=org/springframework/context/config/spring-context-2.5.xsd
2. http\://www.springframework.org/schema/context/spring-context-3.0.xsd=org/springframework/context/config/spring-context-3.0.xsd
3. http\://www.springframework.org/schema/context/spring-context-3.1.xsd=org/springframework/context/config/spring-context-3.1.xsd
4. http\://www.springframework.org/schema/context/spring-context-3.2.xsd=org/springframework/context/config/spring-context-3.2.xsd
5. http\://www.springframework.org/schema/context/spring-context.xsd=org/springframework/context/config/spring-context-3.2.xsd
6. ...

再打开jar包里的org/springframework/context/config/ 目录，可以看到下面有

spring-context-2.5.xsd  
spring-context-3.0.xsd  
spring-context-3.1.xsd  
spring-context-3.2.xsd

很明显，可以想到Spring是把XSD文件放到本地了，再在spring.schemas里做了一个映射，优先从本地里加载XSD文件。

并且Spring很贴心，把旧版本的XSD文件也全放了。这样可以防止升级了Spring版本，而配置文件里用的还是旧版本的XSD文件，然后断网了，应用启动不了。

我们还可以看到，在没有配置版本号时，用的就是当前版本的XSD文件：

**[html]** [view plain](http://blog.csdn.net/hengyunabc/article/details/22295749) [copy](http://blog.csdn.net/hengyunabc/article/details/22295749)

[在CODE上查看代码片](https://code.csdn.net/snippets/264875)

1. http\://www.springframework.org/schema/context/spring-context.xsd=org/springframework/context/config/spring-context-3.2.xsd

同样，我们打开dubbo的jar包，可以在它的spring.schemas文件里看到有这样的配置：

**[html]** [view plain](http://blog.csdn.net/hengyunabc/article/details/22295749) [copy](http://blog.csdn.net/hengyunabc/article/details/22295749)

[在CODE上查看代码片](https://code.csdn.net/snippets/264875)

1. http\://code.alibabatech.com/schema/dubbo/dubbo.xsd=META-INF/dubbo.xsd

所以，Spring在加载dubbo时，会从dubbo的jar里加载dubbo.xsd。

**如何跳过Spring的XML校验？**

可以用这样的方式来跳过校验：

**[html]** [view plain](http://blog.csdn.net/hengyunabc/article/details/22295749) [copy](http://blog.csdn.net/hengyunabc/article/details/22295749)

[在CODE上查看代码片](https://code.csdn.net/snippets/264875)

1. GenericXmlApplicationContext context = new GenericXmlApplicationContext();
2. context.setValidating(false);

**如何写一个自己的spring xml namespace扩展**

可以参考Spring的文档，实际上是相当简单的。只要实现自己的NamespaceHandler，再配置一下spring.handlers和spring.schemas就可以了。

http://docs.spring.io/spring/docs/current/spring-framework-reference/html/extensible-xml.html

**其它的一些东东**

**防止XSD加载不成功的一个思路**

http://hellojava.info/?p=135

**齐全的Spring的namespace的列表**

http://stackoverflow.com/questions/11174286/spring-xml-namespaces-how-do-i-find-what-are-the-implementing-classes-behind-t

**Spring core**

* aop - [AopNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/aop/config/AopNamespaceHandler.html)
* c - [SimpleConstructorNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/beans/factory/xml/SimpleConstructorNamespaceHandler.html)
* cache - [CacheNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/cache/config/CacheNamespaceHandler.html)
* context - [ContextNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/context/config/ContextNamespaceHandler.html)
* jdbc - [JdbcNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/jdbc/config/JdbcNamespaceHandler.html)
* jee - [JeeNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/ejb/config/JeeNamespaceHandler.html)
* jms - [JmsNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/jms/config/JmsNamespaceHandler.html)
* lang - [LangNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/scripting/config/LangNamespaceHandler.html)
* mvc - [MvcNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/web/servlet/config/MvcNamespaceHandler.html)
* oxm - [OxmNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/oxm/config/OxmNamespaceHandler.html)
* p - [SimplePropertyNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/beans/factory/xml/SimplePropertyNamespaceHandler.html)
* task - [TaskNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/scheduling/config/TaskNamespaceHandler.html)
* tx - [TxNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/transaction/config/TxNamespaceHandler.html)
* util - [UtilNamespaceHandler](http://static.springsource.org/spring/docs/3.1.x/javadoc-api/org/springframework/beans/factory/xml/UtilNamespaceHandler.html)

**Spring Security**

* security - [SecurityNamespaceHandler](http://static.springsource.org/spring-security/site/docs/2.0.x/apidocs/org/springframework/security/config/SecurityNamespaceHandler.html)
* oauth - [OAuthSecurityNamespaceHandler](http://static.springsource.org/spring-security/oauth/apidocs/org/springframework/security/oauth/config/OAuthSecurityNamespaceHandler.html)

**Spring integration**

* int - [IntegrationNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/config/xml/IntegrationNamespaceHandler.html)
* amqp - [AmqpNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/amqp/config/AmqpNamespaceHandler.html)
* event - [EventNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/event/config/EventNamespaceHandler.html)
* feed - [FeedNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/feed/config/FeedNamespaceHandler.html)
* file - [FileNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/file/config/FileNamespaceHandler.html)
* ftp - [FtpNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/ftp/config/FtpNamespaceHandler.html)
* gemfire - [GemfireIntegrationNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/gemfire/config/xml/GemfireIntegrationNamespaceHandler.html)
* groovy - [GroovyNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/groovy/config/GroovyNamespaceHandler.html)
* http - [HttpNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/http/config/HttpNamespaceHandler.html)
* ip - [IpNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/ip/config/IpNamespaceHandler.html)
* jdbc - [JdbcNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/jdbc/config/JdbcNamespaceHandler.html)
* jms - [JmsNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/jms/config/JmsNamespaceHandler.html)
* jmx - [JmxNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/jmx/config/JmxNamespaceHandler.html)
* mail - [MailNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/mail/config/MailNamespaceHandler.html)
* redis - [RedisNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/redis/config/RedisNamespaceHandler.html)
* rmi - [RmiNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/rmi/config/RmiNamespaceHandler.html)
* script - [ScriptNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/scripting/config/jsr223/ScriptNamespaceHandler.html)
* security - [IntegrationSecurityNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/security/config/IntegrationSecurityNamespaceHandler.html)
* sftp - [SftpNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/sftp/config/SftpNamespaceHandler.html)
* stream - [StreamNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/stream/config/StreamNamespaceHandler.html)
* twitter - [TwitterNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/twitter/config/TwitterNamespaceHandler.html)
* ws - [WsNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/ws/config/WsNamespaceHandler.html)
* xml - [IntegrationXmlNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/xml/config/IntegrationXmlNamespaceHandler.html)
* xmpp - [XmppNamespaceHandler](http://static.springsource.org/spring-integration/api/org/springframework/integration/xmpp/config/XmppNamespaceHandler.html)

**总结：**

为什么不要在Spring的配置里，配置上XSD的版本号？  
因为如果没有配置版本号，取的就是当前jar里的XSD文件，减少了各种风险。  
而且这样约定大于配置的方式很优雅。

**参考：**

http://stackoverflow.com/questions/10768873/spring-di-applicationcontext-xml-how-exactly-is-xsischemalocation-used

http://stackoverflow.com/questions/11174286/spring-xml-namespaces-how-do-i-find-what-are-the-implementing-classes-behind-t

http://docs.spring.io/spring/docs/current/spring-framework-reference/html/extensible-xml.html