<http://www.cnblogs.com/ysw-go/p/6010646.html>

文章转载自：<https://yq.aliyun.com/articles/40353>

相信很多人和我一样，在编写Spring或者Maven或者其他需要用到XML文档的程序时，通常都是将这些XML文档头拷贝过来，并没有理解其中元素（比如xmlns，xmlns:xsi，xsi:schemaLocation）的真正含义，不知道哪些元素是多余的，也不知道为什么要加那些元素。这样当有时候网上Copy的XML头有错的时候自己却不知道怎么下手。我也是这样的，于是今天花了点时间好好的理解了一下这些元素及其用法，现整理与此，在此谢谢各位前辈的经验，如有总结的不对或者不好的地方，欢迎留言提出各位的宝贵意见。

话不多说，先来一段Spring的XML样本，相信大家都很眼熟：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | <?**xml** version="1.0" encoding="UTF-8"?>  <**beans** xmlns="http://www.springframework.org/schema/beans"         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"         xmlns:context="http://www.springframework.org/schema/context"         xmlns:mvc="http://www.springframework.org/schema/mvc"         xsi:schemaLocation="http://www.springframework.org/schema/beans                             http://www.springframework.org/schema/beans/spring-beans.xsd                             http://www.springframework.org/schema/context                             http://www.springframework.org/schema/context/spring-context.xsd                             http://www.springframework.org/schema/mvc                             http://www.springframework.org/schema/mvc/spring-mvc.xsd">      <**context:component-scan** base-package="xxx.xxx.controller" />        <**context:annotation-config**/>      <**mvc:default-servlet-handler**/>      <**mvc:annotation-driven**/>        <**mvc:resources** mapping="/images/\*\*" location="/images/" />        <**bean** id="xxx" class="xxx.xxx.xxx.Xxx">          <**property** name="xxx" value="xxxx"/>      </**bean**>  </**beans**> |

    这个文档中，根元素<beans/>就不用说了，接下来是xmlns。那么什么是xmlns呢？xmlns其实是XML Namespace的缩写，可译为“XML命名空间”，但个人觉得，翻译后的名字反而不好理解，所以我们就叫它为XML Namespace吧。

### 为什么需要xmlns？

    考虑这样两个XML文档：表示HTML表格元素的<table/>：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <**table**>    <**tr**>      <**td**>Apples</**td**>      <**td**>Bananas</**td**>    </**tr**>  </**table**> |

    和描述一张桌子的<table/>：

|  |  |
| --- | --- |
| 1  2  3  4  5 | <**table**>    <**name**>African Coffee Table</**name**>    <**width**>80</**width**>    <**length**>120</**length**>  </**table**> |

    假如这两个 XML 文档被一起使用，由于两个文档都包含带有不同内容和定义的 <table> 元素，就会发生命名冲突。XML 解析器是无法确定如何处理这类冲突。为了解决上述问题，xmlns就产生了。

### ****如何是用xmlns？****

    很简单，使用语法：**xmlns:namespace-prefix="namespaceURI"**。其中namespace-prefix为自定义前缀，只要在这个XML文档中保证前缀不重复即可；namespaceURI是这个前缀对应的XML Namespace的定义。例如，

|  |  |
| --- | --- |
| 1 | xmlns:context="http://www.springframework.org/schema/context" |

    这一句定义了一个http://www.springframwork.org/schema/context的Namespace（这和Java类中的包的声明很相似），并将其和前缀context绑定。所以上面的Spring XML文档中会有这么一句：

|  |  |
| --- | --- |
| 1 | <**context:component-scan** base-package="xxx.xxx.controller" /> |

    这里的<component-scan/>元素就来自别名为context的XML Namespace，也就是在http://www.springframework.org/schema/context中定义的。

    我们还可以将前缀定义为abc：

|  |  |
| --- | --- |
| 1 | xmlns:abc="namespaceURI" |

    这样再使用这个namespaceURI中的元素时，需要以abc为前缀，例如：<abc:xxx/>。再拿上面的例子解释怎么使用xmlns：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <!-- 这里xmlns:h="url1"表示这个table是用h作为标记，table的写法在url1中定义 -->  <**h:table** xmlns:h="url1">    <**h:tr**>      <**h:td**>Apples</**h:td**>      <**h:td**>Bananas</**h:td**>    </**h:tr**>  </**h:table**> |

    和：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <!-- 这里xmlns:f="url2"表示这个table是用f作为标记，table的写法在url2中定义 -->  <**f:table** xmlns:f="url2">    <**f:name**>African Coffee Table</**f:name**>    <**f:width**>80</**f:width**>    <**f:length**>120</**f:length**>  </**f:table**> |

    后者与前者仅仅使用不同前缀，我们为 <table> 标签添加了一个 xmlns 属性，这样就为前缀赋予了一个与某个命名空间相关联的限定名称。此时再把它们放在一起，XML解析器就不会报错了。

    注意：当xmlns被定义在元素的开始标签中（如这里的<f:table/>）时，所有带有相同前缀的子元素都会与同一个Namespace相关联（即<f:table/>里面的<f:name/>和<f:width/>也会使用url2定义的写法）。

### xmlns和xmlns:xsi有什么不同？

    xmlns表示默认的Namespace。例如Spring XML文档中的

|  |  |
| --- | --- |
| 1 | xmlns="http://www.springframework.org/schema/beans" |

    这一句表示该文档默认的XML Namespace为http://www.springframwork.org/schema/beans。**对于默认的Namespace中的元素，可以不使用前缀**。例如Spring XML文档中的

|  |  |
| --- | --- |
| 1  2  3 | <**bean** id="xxx" class="xxx.xxx.xxx.Xxx">    <**property** name="xxx" value="xxxx"/>  </**bean**> |

    xmlns:xsi表示使用xsi作为前缀的Namespace，当然前缀xsi需要在文档中声明。

### ****xsi:schemaLocation有何作用？****

    xsi:schemaLocation属性其实是Namespace为http://www.w3.org/2001/XMLSchema-instance里的schemaLocation属性，正是因为我们一开始声明了

|  |  |
| --- | --- |
| 1 | xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" |

    这里才写作xsi:schemaLocation（当然一般都使用这个前缀）。它定义了XML Namespace和对应的XSD（Xml Schema Definition）文档的位置的关系。它的值由一个或多个URI引用对组成，两个URI之间以空白符分隔（空格和换行均可）。第一个URI是定义的XML Namespace的值，第二个URI给出Schema文档的位置，Schema处理器将从这个位置读取Schema文档，**该文档的targetNamespace必须与第一个URI相匹配**。例如：

|  |  |
| --- | --- |
| 1  2 | xsi:schemaLocation="http://www.springframework.org/schema/context                      http://www.springframework.org/schema/context/spring-context.xsd" |

    这里表示Namespace为http://www.springframework.org/schema/context的Schema的位置为<http://www.springframework.org/schema/context/spring-context.xsd>。这里我们可以打开这个Schema的位置，下面是这个文档的开始部分：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <**xsd:schema** xmlns="http://www.springframework.org/schema/context"              xmlns:xsd="http://www.w3.org/2001/XMLSchema"              xmlns:beans="http://www.springframework.org/schema/beans"              xmlns:tool="http://www.springframework.org/schema/tool"                <!-- 这里的targetNamespace和上方xsi:schemaLocation中的第一个URI匹配 -->              targetNamespace="http://www.springframework.org/schema/context"              elementFormDefault="qualified"              attributeFormDefault="unqualified"> |

    有了上面的说明后，再去理解开始的Spring XML文档，一定会有不一样的感觉！

    最后再次感谢各位前辈的宝贵经验。