## 命名空间

命名空间相当于是为前缀设置一个别名(相当于Java中的变量名)。命名空间只能定义在根节点中。

|  |
| --- |
| <students xmlns:a=*"http://www.hwhlxy.com/a"*  xmlns:b=*"http://www.hwhlxy.com/b"*>  <student>  <a:name>小黄</a:name>  <gender>男</gender>  <friend>  <b:name>小黄人</b:name>  </friend>  </student>  <student>  <a:name>小董</a:name>  <gender>男</gender>  <friend>  <b:name>小刘</b:name>  </friend>  </student>  </students> |

定义了命名空间后，前缀a、b不再是单一字母。而是表示其所代表的一个完整域名内容

### 默认命名空间

定义命名空间时，没有指定别名的命名空间就称为默认命名空间。所有没有应用特定命名空间的标签，默认使用的都是默认命名空间。

|  |
| --- |
| <students xmlns=*"http://www.hwhlxy.com/a"*  xmlns:b=*"http://www.hwhlxy.com/b"*>  <student>  <name>小黄</name>  <gender>男</gender>  <friend>  <b:name>小黄人</b:name>  </friend>  </student>  <student>  <name>小董</name>  <gender>男</gender>  <friend>  <b:name>小刘</b:name>  </friend>  </student>  </students> |

## Dtd

### 概述

Document Type Defined的缩写，简称为文档类型定义。同过DTD可以对Xml文档的内容进行约束。目前为止DTD的使用已经较少，更多的是应用Schema。

### 语法

1. 语法

<!DOCTYPE 根元素 [元素声明]>

示例

|  |
| --- |
| <!DOCTYPE students [  ]> |

1. 元素声明

<!ELEMENT 元素名称(子元素名称[+])

* +可以省略，如果带有则表示元素至少出现一次

|  |
| --- |
| <!ELEMENT students (student+)>  <!ELEMENT student (name,age,gender,grade)> |

### 定义形式

1. 内部约束

* 缺点

定义在xml文档内部的约束就称为内部约束。内部约束在所在文档中生效

* 示例

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!DOCTYPE students [  <!ELEMENT students (student+)>  <!ELEMENT student (name,age,gender,grade)>  ]>  <students>  <student>  <name>Mike</name>  <age>20</age>  <gender>男</gender>  <grade>一年一班</grade>  </student>  </students> |

1. 外部约束(独立的约束文件)

* 步骤
  + 创建约束文件（.DTD文件。DTD文件的本质还是一个XML文件）
  + 编写约束内容

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!ELEMENT students (student+)>  <!ELEMENT student (name,age,gender,grade)> |

* + xml文件中引入约束文件
* 引入约束文件
  + 引入项目内的约束文件

语法：

<!DOCTYPE 根节点名称 SYSTEM "DTD约束文件路径">

示例：

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!DOCTYPE students SYSTEM "student.dtd">  <students>  <student>    </student>  </students> |

* + 引入公共的约束文件

语法：（SYSTEM—项目内引入|PUBLIC—文件地址引入）

<!DOCTYPE 根节点名 PUBLIC "唯一标识" "文件地址">

--唯一标识 ：一般用域名进行标识

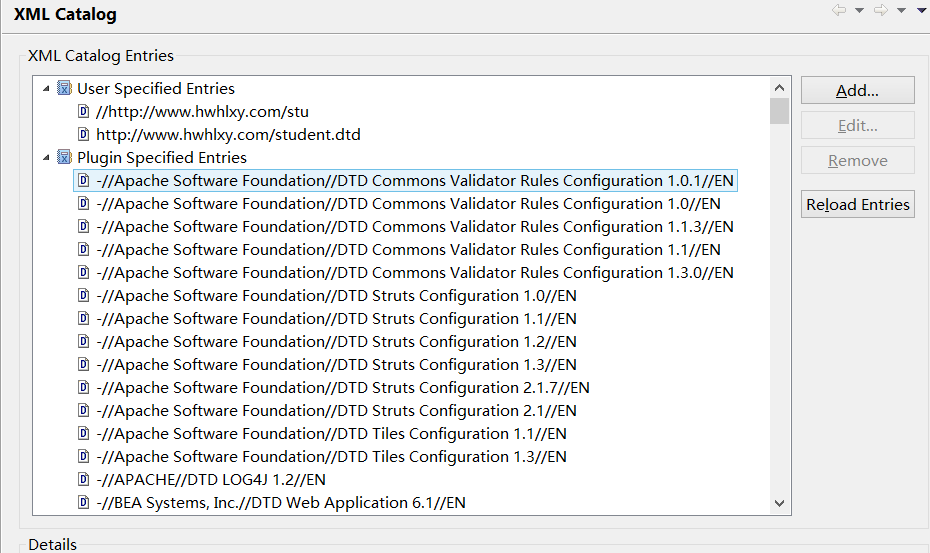
--文件地址 ：可以是本地地址，也可以是URL地址。 如果是URL地址，则同过地址必须可以找到文件

示例

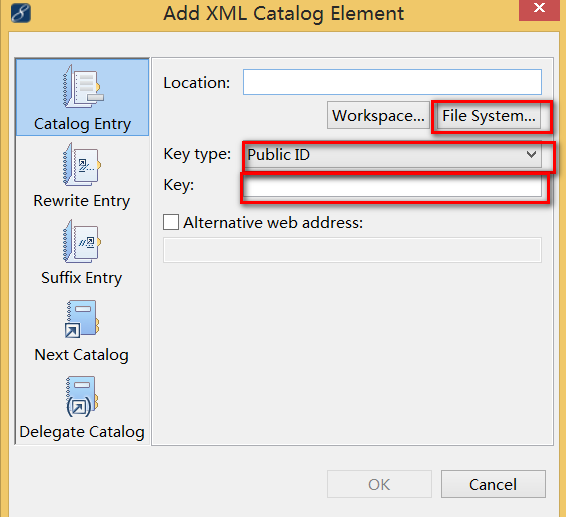
|  |
| --- |
| 本地地址  <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!DOCTYPE students PUBLIC "//http://www.hwhlxy.com/stu" "student.dtd">  <students>  <student>    </student>  </students> |
| 网络地址  <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!DOCTYPE students PUBLIC "//http://www.hwhlxy.com/stu" "http://www.hwhlxy.com/student.dtd">  <students>  <student>    </student>  </students> |

手动引入（重点）

Window🡪preferences🡪Xml Catalog



点击 add

通过file System 找到dtd文件为位置  


两种方法引入：

1. 通过唯一标识符
2. 通过URI



## Schema

### DTD的缺点

* DTD使用了非Xml的语法，适用过程中需要先进性学习
* DTD只能指定简单的规则
* DTD无法设置元素的数据类型
* DTD中无法使用命名空间

### Schema应用步骤

1. 创建Schema文件(.xsd)

* 代码

|  |
| --- |
| <schema xmlns=*"http://www.w3.org/2001/XMLSchema"*  targetNamespace=*"http://www.example.org/student"*  xmlns:tns=*"http://www.example.org/student"*  elementFormDefault=*"qualified"*>      </schema> |

* 说明
* xmlns ：命名空间，不需要修改
* targetNamespace ：目标命名空间
* xmlns:tns ：命名空间，值与targetNamespace同
* elementFormDefault：设置定义规则。qualified---严格的规范(Xml文档中比前必须符合Schema的设定。否则会报错)。unqualified---宽松的。可以不遵守Schema的约定。一般不被使用

1. 编写Schema文件

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <schema xmlns=*"http://www.w3.org/2001/XMLSchema"*  targetNamespace=*"http://www.hwhlxy.com/stu"*  xmlns:tns=*"http://www.hwhlxy.com/stu"*  elementFormDefault=*"qualified"*>  <element name=*"students"*>  <complexType>  <sequence>  <element name=*"student"*></element>  </sequence>  </complexType>  </element>  </schema> |

1. 引入Schema文件

* 代码

|  |
| --- |
| <students xmlns=*"http://www.hwhlxy.com/stu"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://www.hwhlxy.com/stu student.xsd"*>  </students> |

* 说明
* xmlns ：命名空间，其值与Schema的命名空间一致
* xmlns:xsi ：命名空间，其值固定
* xsi:schemaLocation ：定位Schema文件的位置。