**Teoria xsd namespaces**

**Scenario: XSD NAMESPACES**Molte persone lavorano su files di tipi xsd che poi vengono composti in modo diverso per vari servizi.

**target namespace in <definition>:**  
Definisce un namespace (come package in java)  
Tutti i types definiti in questa <definition></definition> saranno locali a questo namespace.

**import:**Aggiunge altri elements e types alla definition corrente.  
Per il parser e’ come se venisse fatto un merge dei due <definition> elements e dei loro figli.  
Quando si importa un file e’ necessario specificare il suo namespace.  
Per poterli utilizzare e’ necessario pero’ definire **una reference** (con un acronimo) nel <**main definition**>  
  
**reference ad un namespace** **in <definition>:**Import di un namespace. Ogni volta che si usa quel acronimo il parser cerchera’ un elemento o un type tra tutte le definition caricate, cercandolo nel namespace appropriato (se ne trova uno)

**Steps per definire N files con ns diversi ed importarli tutti in un main.xsd:**

1. Definire il targetnamespace dei tipi di ogni definition in ogni file .xsd  
   Es. Common.xsd,
2. Importare il file ed il suo namespace nel file parent
3. Referenziare il namespace con un acronimo
4. Usare l’acronimo:tipo per usare tipi importati.

**Esempio:**

1. **User.xsd: Definizione targetnamespace**

<xs:schema

xmlns:xs=*"http://www.w3.org/2001/XMLSchema"*

targetNamespace=*"http://ee.spring.corso/ws/UserType"*

elementFormDefault=*"qualified"*>

1. **Main.xsd Import file e namespace esterno.**

<xs:import namespace=[*http://ee.spring.corso/ws/UserType*](http://ee.spring.corso/ws/UserType)

schemaLocation=*"UserType.xsd"*/>

1. **Main.xsd: reference con acronimo al ns importato**

<wsdl:types>

<xs:schema

xmlns:usr=*"http://ee.spring.corso/ws/UserType"*

xmlns:xs=*"http://www.w3.org/2001/XMLSchema"*

elementFormDefault=*"qualified"*

targetNamespace=*"http://ee.spring.corso/ws/UserService"*>

1. **Main.xsd: reference con acronimo al ns importato**

<xs:element name=*"echoUserResponse"*>

<xs:complexType>

<xs:sequence>

<xs:element name=*"user"* type=*"usr:UserType"*/>

</xs:sequence>

</xs:complexType>

</xs:element>