

## Lecture 1: **Complex Type- qualified or unqualified ?**

The *xsd:complexType* element contains what's known as compositor, which describes the composition of the type's content, also known as its content model. XML Schema defines three compositors that can be used in complex type definitions including *xsd:sequence*, *xsd:choice*, and *xsd:all*.

Compositors contain particles, which includes things like other compositor, element declarations, wildcards, and model groups. Attribute declarations are not considered particles because they don't repeat. Hence, **attribute declarations are not placed within a compositor** but **after the compositor** at the end of the complex type definition.

The elements and attributes declared within the *xsd:complexType* element are considered local to the complex type. Local elements and attributes can only be used within the context where they're defined.

whether local elements/attributes need to be namespace qualified in instance documents?

Since local elements and attributes will always contain an ancestor element (typically a global element) qualified by the target namespace, one could argue that it's not necessary.

Due to this reasoning, in XML Schema **local elements and attributes should be unqualified by default**. Hence, a valid instance of the author element looks like this:

### In XSD:

```
<xsd:schema targetNamespace="http://www.ntou.edu.tw/XML"
xmlns:myNS="http://www.ntou.edu.tw/XML"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
.....
<xsd:complexType name="AuthorType">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string" />
    <xsd:element name="phone" type="myNS:Phone" />
  </xsd:sequence>
  <xsd:attribute name="id" type="myNS:SSN" />
</xsd:complexType>
<xsd:element name="author" type="myNS:AuthorType" />
```

Local elements must be unqualified by default

Global elements are qualified.

**In XML:**

```
<x:author xmlns:x="http://www.ntou.edu.tw/XML"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ntou.edu.tw/XML mySchema_0100.xsd"
id="333-33-3456">
  <name>Araron</name>
  <phone>(02)8662-5925</phone>
</x:author>
```

Local elements are unqualified

XML Schema makes it possible to explicitly control whether a given local element/ attribute should be **qualified** or **unqualified** using the form attribute on *xsd:element/ xsd:attribute* or by using the **elementFormDefault/attributeFormDefault** attributes on *xsd:schema*.

**In XSD:**

```
<xsd:schema targetNamespace="http://www.ntou.edu.tw/XML"
xmlns:myNS="http://www.ntou.edu.tw/XML"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
```

Local elements must be qualified

```
.....
<xsd:complexType name="AuthorType">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string" />
    <xsd:element name="phone" type="myNS:Phone" />
  </xsd:sequence>
  <xsd:attribute name="id" type="myNS:SSN" />
</xsd:complexType>
<xsd:element name="author" type="myNS:AuthorType" />
```

**In XML:**

qualified

```
<x:author xmlns:x="http://www.ntou.edu.tw/XML"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ntou.edu.tw/XML mySchema_0100.xsd"
id="333-33-3456">
```

qualified

```
<x:name>Araron</name>
```

```
<x:phone>(02)8662-5925</phone>  
</x:author>
```

or using the namespace by default:

qualified

```
<author xmlns="http://www.ntou.edu.tw/XML"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://www.ntou.edu.tw/XML mySchema_0100.xsd"  
id="333-33-3456">
```

qualified

```
<name>Araron</name>  
<phone>(02)8662-5925</phone>  
</author>
```

(參考: myXml\_0100.xml 及 mySchema\_0100.xsd)

## Lecture 2: **Complex Type- (Global & Local Declaration)**

### Named Types: (Global & Local Declaration)

You can define a simple or complex type to place the definition- that is, the *xsd:simpleType* or *xsd:complexType* element- directly within the *xsd:schema* element and assign the definition a name. You can then apply the type to one or more elements or attributes by assigning the type's name to the type attribute in the declaration.

```
<?xml version="1.0"?>
<!-- my first Schema file mySchema_0031.xsd -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.ntou.edu.tw/XML"
xmlns:myNS="http://www.ntou.edu.tw/XML"
elementFormDefault="unqualified">
```

這個 simpleType 是  
一個 global type

```
<xsd:simpleType name="PriceType">
  <xsd:restriction base="xsd:decimal">
    <xsd:minExclusive value="0" />
    <xsd:maxExclusive value="500" />
  </xsd:restriction>
</xsd:simpleType>
```

```
<xsd:element name="book">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="title" type="xsd:string"/>
      <xsd:element name="author" type="xsd:string"/>
      <xsd:element name="page" type="xsd:positiveInteger"/>
      <xsd:element name="price" type="myNS:PriceType"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

這四個 element 都是 local  
declaration (因為採用 type="..." 因  
此 ~~or may not~~ 需要作 qualify。

```
</xsd:schema>
```

**(Demo)** myXml\_0031.xml and mySchema\_0031.xsd

**(Demo 2)** myXml\_0100.xml and mySchema\_0100.xsd  
(使用 **type** 這個屬性來設定本 element 的 **type**)

**In XSD:**

```
<?xml version="1.0"?>  
<!-- my first Schema file mySchema_0100.xsd -->  
<xsd:schema targetNamespace="http://www.ntou.edu.tw/XML "  
xmlns:myNS="http://www.ntou.edu.tw/XML "  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
elementFormDefault="unqualified" attributeFormDefault="qualified">
```

所以 所有的 local declaration  
必須是 unqualified

這二個 element 都是 local  
declaration (因為採用 **type="..."**)  
因此 may or may need 要作 qualify。

```
<xsd:complexType name="AuthorType">  
  <xsd:sequence>  
    <xsd:element name="name" type="xsd:string"/>  
    <xsd:element name="phone" type="myNS:Phone"/>  
  </xsd:sequence>  
  <xsd:attribute name="id" type="myNS:SSN"/>  
</xsd:complexType>  
<xsd:element name="author" type="myNS:AuthorType"/>  
  
</xsd:schema>
```

**In XML:**

```
<?xml version="1.0" encoding="Big5"?>  
<!-- my first Xml file myXml_0100.xml -->  
<x:author xmlns:x="http://www.ntou.edu.tw/XML "  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://www.ntou.edu.tw/XML mySchema_0100.xsd"  
x:id="333-33-3456">  
  <name>Araron</name>  
  <phone>(02)8662-5925</phone>  
</x:author>
```

qualified

unqualified

## Lecture 3: Using ref/type attribute

You can also reference global element/attribute declarations from within a complex type using the **ref** attribute.

### In XSD:

```
<xsd:schema targetNamespace="http://www.ntou.edu.tw/XML"
xmlns:myNS="http://www.ntou.edu.tw/XML"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
```

```
.....
<xsd:attribute name="id" type="myNS:SSN" />
<xsd:element name="name" type="xsd:string" />
```

已有一個 element 宣告，採  
用 ref 可以參考到現有的  
element 宣告。(是 global)

```
<xsd:complexType name="AuthorType">
  <xsd:sequence>
    <xsd:element ref="myNS:name"/>
    <xsd:element name="phone" type="myNS:Phone" />
  </xsd:sequence>
  <xsd:attribute name="id" type="myNS:SSN" />
</xsd:complexType>
<xsd:element name="author" type="myNS:AuthorType" />
```

已有一個 type 宣  
告，所以可以直接  
採用。(是 local)

### In XML:

```
<author xmlns="http://www.ntou.edu.tw/XML"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ntou.edu.tw/XML mySchema_0100.xsd"
id="333-33-3456">
  <name>Araron</name>
  <phone>(02)8662-5925</phone>
</author>
```

Qualified  
by default!

Since **id** and **name** are global attribute and element, they always need to be qualified in instance documents. Using **“ref”** specifies that the global element can also be used within the context of AuthorType, but it doesn't change the fact it needs to be qualified. The phone element is still defined locally, which means it **may or may not**

need (就我的實作結論：該作 qualified 就必須是 qualified，該作 unqualified 的 就必須嚴格 unqualified) to be qualified in an instance depending on the form in use.

So assuming `elementFormDefault="unqualified"`, a valid instance would look like this:

```
<x:author xmlns:x="http://www.ntou.edu.tw/XML"
.....
x:id="333-33-3456">
  <x:name>Araron</x:name>
  <phone>(02)8662-5925</phone>
</x:author>
```

(參考： myXml\_0101.xml 及 mySchema\_0101.xsd)

由於 Schema 指定了  
`elementFormDefault="unqualified"` 所以：  
Global 的 element 必須 qualified  
(name 是一個 global element)  
local 的 element 必須是 unqualified  
(phone 是一個 local element)

用 ref 方式，其參照的 element 是 global，所以必定要 prefixed！！

由於 phone 這個 element 是 local，所以是否要作 qualify 要視 `elementFormDefault` 的值而定。

## Lecture 4: Using `xsd:choice` & `xsd:all`

The `xsd:complexType` element contains what's known as a compositor, which describes the composition of the type's content, also known as its content model. XML Schema defines three compositors that can be used in complex type definitions including `xsd:sequence`, `xsd:choice`, and `xsd:all`. Now for a slightly more sophisticated example that uses nested complex types, other compositors, and repeating particles:

### In XSD:

```
.....
<xsd:complexType name="AddressType">
  <xsd:all>
    <xsd:element name="street" type="xsd:string" />
    <xsd:element name="city" type="xsd:string minOccurs="0" />
    <xsd:element name="state" type="tns:State" minOccurs="0" />
    <xsd:element name="zip" type="tns:Zip" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="PublicationsListType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="book" type="xsd:string" />
    <xsd:element name="article" type="xsd:string" />
    <xsd:element name="whitepaper" type="xsd:string" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="AuthorType">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="name" type="xsd:string" />
      <xsd:element name="fullName" type="xsd:string" />
    </xsd:choice>
    <xsd:element name="address" type="tns:AddressType" />
    <xsd:element name="phone" type="tns:Phone" minOccurs="0"
      maxOccurs="unbound" />
    <xsd:element name="recentPublications"
      type="tns:PublicationsListType" />
  </xsd:sequence>
</xsd:complexType>
```



```
</xsd:sequence>  
  <xsd:attribute name="id" type="tns:AuthorId" />  
</xsd:complexType>  
</xsd:element name="author" type="tns:AuthorType" />  
.....
```

In this example, **AuthorType** contains a sequence of another compositor, a **choice**, and is followed by three element declarations. Some of the elements are of other user-defined complex types (**AddressType** and **PublicationsListType**), which effectively define nested structures within the type. The choice means that either the **name** or **fullName** element is allowed to appear at that location. And finally, the all compositor in **AddressType** indicates that the order of the elements is insignificant.

Notice also that the **phone** element declaration specifies occurrence constraints using the **minOccurs** and **maxOccurs** attributes. Occurrence constraints may be applied to any particle in a complex type. The default value for each is 1, which means the given particle must appear exactly once at the specified location. Specifying **minOccurs="0"** makes the given particle optional and specifying **maxOccurs="unbounded"** allows the particle to repeat infinitely.

#### In XML Instance:

```
<x:author xmlns:x="http://www.ntou.edu.tw/XML"  
id="333-33-4444" >  
  <name>Allen</name>  
  <address>  
    <street>123 Main</street>  
    <zip>82493</zip>  
  </address>  
  <phone>801-729-0234</phone>  
  <phone>802-123-2344</phone>  
  <phone>802-342-5899</phone>  
  <recentPublications>  
    <whitepaper>web service abstractions</whitepaper>  
    <book>Essential XML Quick Reference</book>  
    <article>Understanding SOAP</article>  
    <book>Essential XML</book>  
  </recentPublications>  
</x:author>
```

