What is a Complex Element?

A complex element is an XML element that contains other elements and/or attributes.

There are four kinds of complex elements:

* empty elements
* elements that contain only other elements
* elements that contain only text
* elements that contain both other elements and text

**Note:** Each of these elements may contain attributes as well!

Examples of Complex Elements

A complex XML element, "product", which is empty:

<product pid="1345"/>

A complex XML element, "employee", which contains only other elements:

<employee>  
 < firstname>John</firstname>  
 < lastname>Smith</lastname>  
< /employee>

A complex XML element, "food", which contains only text:

<food type="dessert">Ice cream</food>

A complex XML element, "description", which contains both elements and text:

<description>  
It happened on <date lang="norwegian">03.03.99</date> ....  
< /description>

How to Define a Complex Element

Look at this complex XML element, "employee", which contains only other elements:

<employee>  
  <firstname>John</firstname>  
  <lastname>Smith</lastname>  
</employee>

We can define a complex element in an XML Schema two different ways:

1. The "employee" element can be declared directly by naming the element, like this:

<xs:element name="employee">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="firstname" type="xs:string"/>  
      <xs:element name="lastname" type="xs:string"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>

If you use the method described above, only the "employee" element can use the specified complex type. Note that the child elements, "firstname" and "lastname", are surrounded by the <sequence> indicator. This means that the child elements must appear in the same order as they are declared. You will learn more about indicators in the XSD Indicators chapter.

2. The "employee" element can have a type attribute that refers to the name of the complex type to use:

<xs:element name="employee" type="personinfo"/>  
  
<xs:complexType name="personinfo">  
  <xs:sequence>  
    <xs:element name="firstname" type="xs:string"/>  
    <xs:element name="lastname" type="xs:string"/>  
  </xs:sequence>  
</xs:complexType>

If you use the method described above, several elements can refer to the same complex type, like this:

<xs:element name="employee" type="personinfo"/>  
<xs:element name="student" type="personinfo"/>  
<xs:element name="member" type="personinfo"/>  
  
<xs:complexType name="personinfo">  
  <xs:sequence>  
    <xs:element name="firstname" type="xs:string"/>  
    <xs:element name="lastname" type="xs:string"/>  
  </xs:sequence>  
</xs:complexType>

You can also base a complex element on an existing complex element and add some elements, like this:

<xs:element name="employee" type="fullpersoninfo"/>  
  
<xs:complexType name="personinfo">  
  <xs:sequence>  
    <xs:element name="firstname" type="xs:string"/>  
    <xs:element name="lastname" type="xs:string"/>  
  </xs:sequence>  
</xs:complexType>  
  
<xs:complexType name="fullpersoninfo">  
  <xs:complexContent>  
    <xs:extension base="personinfo">  
      <xs:sequence>  
        <xs:element name="address" type="xs:string"/>  
        <xs:element name="city" type="xs:string"/>  
        <xs:element name="country" type="xs:string"/>  
      </xs:sequence>  
    </xs:extension>  
  </xs:complexContent>  
</xs:complexType>