

Dealing with Multiple XML Schema Files

Roger L. Costello

May 2011

1. It is okay to have multiple elements with the same name, provided they are locally declared and have the same data type. Two identical `classification` elements are declared within document:

```
<xs:element name="document">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="classification" type="xs:string" />
      <xs:element name="para" type="xs:string" />
      <xs:element name="classification" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

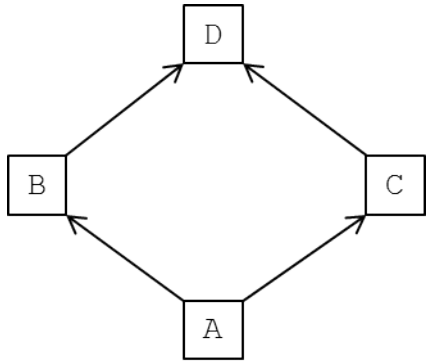
2. It is not okay to have multiple elements with the same name if they are globally declared. Two `altitude` elements are declared globally (*not valid*):

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="altitude" type="xs:int" />
  <xs:element name="altitude" type="xs:int" />

</xs:schema>
```

3. It is okay to have multiple schemas reuse one schema file. Both `B.xsd` and `C.xsd` reuse (*i.e.*, `xs:include`) `D.xsd`:



Here are the schema files:

A.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="B.xsd" />
  <xs:include schemaLocation="C.xsd" />

  <xs:element name="A">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="B" />
        <xs:element ref="C" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

B.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="D.xsd" />

  <xs:element name="B">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="D" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

C.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```

<xs:include schemaLocation="D.xsd" />

<xs:element name="C">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="D" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>

```

D.xsd

```

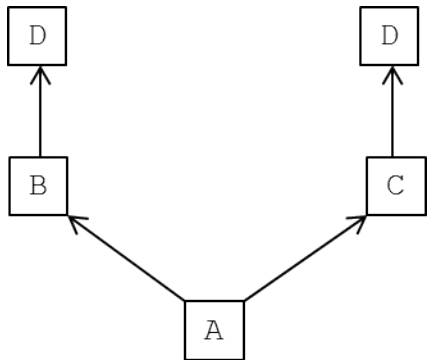
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="D" type="xs:int" />

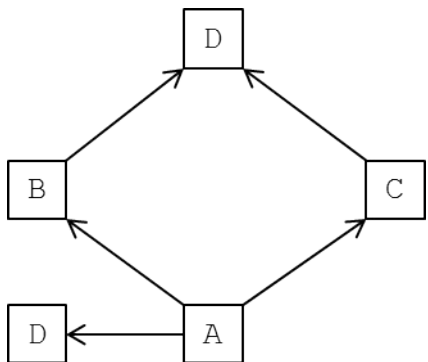
</xs:schema>

```

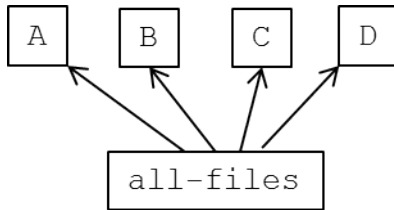
4. It is not okay to make multiple copies of a schema file and have multiple schemas reuse the multiple copies. D.xsd is placed in folder1 and a copy is placed in folder2. B.xsd uses the copy in folder1 and C.xsd uses the copy in folder2 (*not valid*):



5. Likewise, it is not okay to do this:



6. It is okay to create a schema file that uses other schema files even though it doesn't `xs:include` them. Here's how: create one file that `xs:includes` all the files:



Here are the schema files and an instance document. Notice that there are no `xs:include` elements in `A.xsd`, `B.xsd`, and `C.xsd` even though they are using elements in other schema files:

all-files.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="A.xsd" />
  <xs:include schemaLocation="B.xsd" />
  <xs:include schemaLocation="C.xsd" />
  <xs:include schemaLocation="D.xsd" />

</xs:schema>
```

A.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="A">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="B" />
        <xs:element ref="C" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

B.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="B">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="D" />
      </xs:sequence>
    </xs:complexType>

</xs:schema>
```

```
    </xs:element>
</xs:schema>
```

C.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

    <xs:element name="C">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="D" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>

</xs:schema>
```

D.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

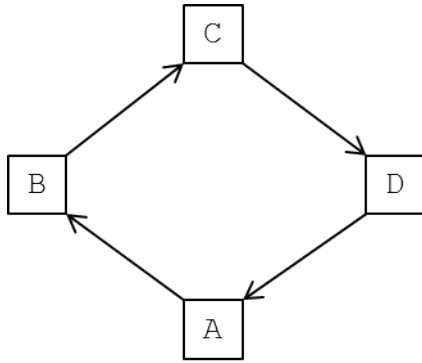
    <xs:element name="D" type="xs:int" />

</xs:schema>
```

Instance.xml

```
<A>
    <B>
        <D>10</D>
    </B>
    <C>
        <D>99</D>
    </C>
</A>
```

7. It is okay to have circular references. A.xsd references (*i.e.*, `xs:includes`) B.xsd, which references C.xsd, which references D.xsd, which completes the circle by referencing A.xsd:



Here are the schema files and an instance document. It is important that somewhere in this circle an element (or a reference to an element) is optional. Otherwise there is an infinite loop and you will get an error when you validate your schemas. In this set of schemas the reference to element A (from D.xsd) is optional.

A.xsd

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="B.xsd" />

  <xs:element name="A">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="B" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>

```

B.xsd

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="C.xsd" />

  <xs:element name="B">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="C" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>

```

C.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="D.xsd" />

  <xs:element name="C">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="D" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

D.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="A.xsd" />

  <xs:element name="D">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="A" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

Instance.xml

```
<A>
  <B>
    <C>
      <D>
    </D>
    </C>
  </B>
</A>
```