

Load from XML Schema

Load from XML Schema

http://en.wikibooks.org/wiki/XForms/Load_from_XML_Schema

This Book Is Generated By [Wb2PDF](#)

using

[RenderX XEP](#), XML to PDF XSL-FO Formatter

Table of Contents

1. Load from XML Schema.....	4
Motivation.....	4
Screen Image.....	4
Link to XForms Application.....	5
XML Schema diagram.....	5
Sample Program.....	6
XML Schema.....	8
XML Instance.....	8
Testing.....	9
Discussion.....	9
Known bugs.....	9

Load from XML Schema

Motivation

Many times the data types of your data are defined in an external XML Schema file. To use this information, rather than having to explicitly bind the instance to a data type, you can just load the XML Schema file to initialize instance data in the model. The data types will be inferred from the XML Schema file.

To do this you must add a **schema** attribute to the model where the value of the argument is the XML Schema source file. Note that the argument is `schema` NOT `src`. `src` is used to read in the instance data.

```
<xf:model id="test" schema="data-types.xsd">
  <xf:instance src="instance-data.xml"/>
</xf:model>
```

Screen Image

Here is a screen image of a sample program. Note that input form for the date and boolean data types are automatically inferred from the XML Schema and different controls are placed in the user interface.

Date:	<input type="text" value="2006-09-21"/>	<input type="button" value="Calendar"/>
Required String:	<input type="text" value="The quick brown fox jum"/>	
Optional String:	<input type="text" value="This string is optional."/>	
Unbounded String:	<input type="text" value="One of many."/>	
Unbounded String:	<input type="text" value="Two of many."/>	
Unbounded String:	<input type="text" value="Three of many."/>	
Boolean:	<input checked="" type="checkbox"/>	
Integer:	<input type="text" value="-123"/>	
Positive Integer:	<input type="text" value="2"/>	
Short:	<input type="text" value="4096"/>	
Byte:	<input type="text" value="127"/>	
Unsigned Byte:	<input type="text" value="255"/>	

Note that in this example, the order of the controls is also changed.

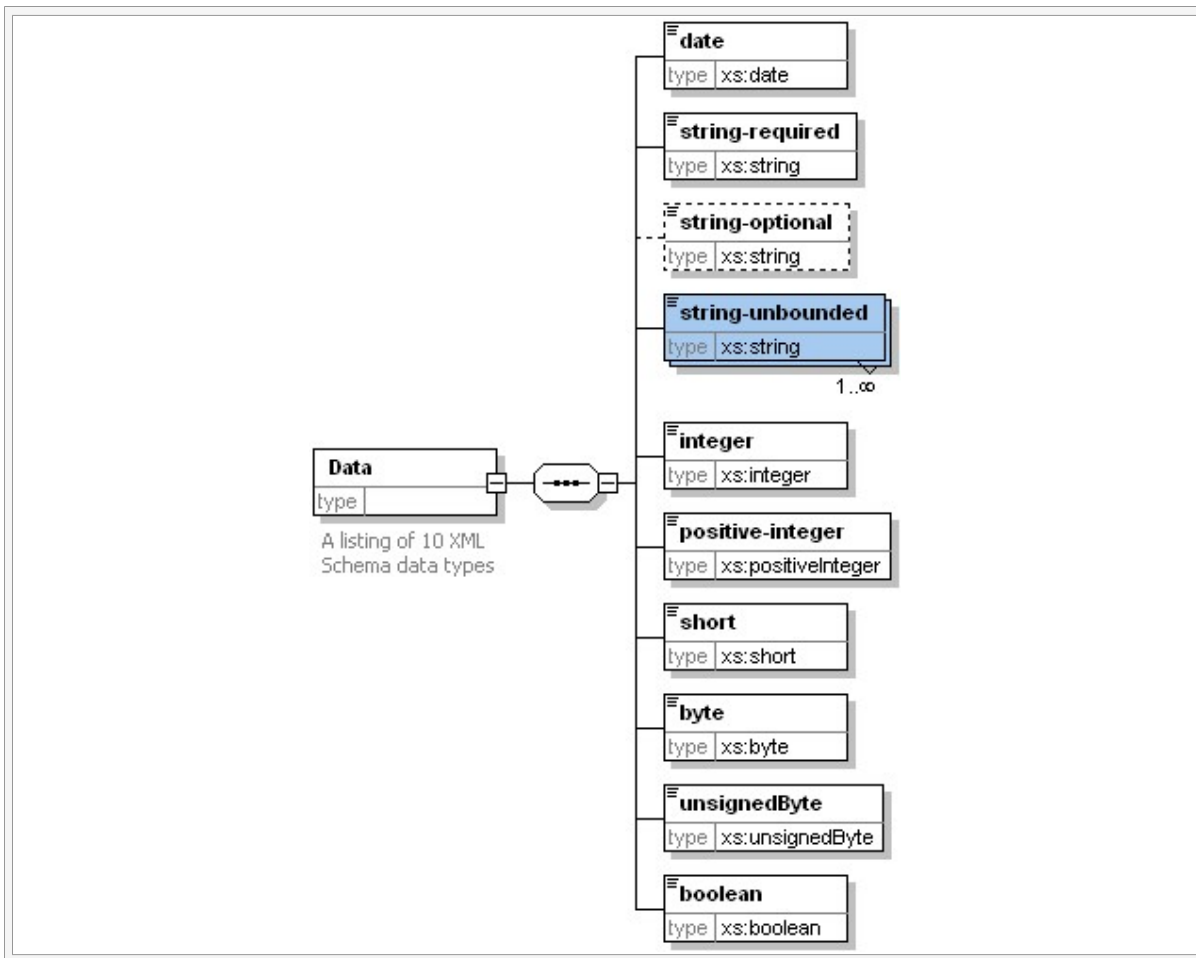
Link to XForms Application

[Load Datatypes from XML Schema](#)

XML Schema diagram

The following is an XML Schema diagram with data types visible. Note that the data type is displayed directly under the data element. This diagram also shows optional data elements using a dashed line and the cardinality of the data elements.

Load from XML Schema



Sample Program

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:xf="http://www.w3.org/2002/xforms"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <head>
    <title>Loading data from an XML Schema file</title>
    <style type="text/css">
      @namespace xf url("http://www.w3.org/2002/xforms");
      xf|input {
        display: table-row;
        line-height: 2em;
      }

      xf|label {
        display: table-cell;
        text-align: right;
        font-family: Ariel, Helvetica, sans-serif;
      }
    </style>
  </head>
  <body>
    <table border="1">
      <tr>
        <td>date</td>
        <td>xs:date</td>
      </tr>
      <tr>
        <td>string-required</td>
        <td>xs:string</td>
      </tr>
      <tr>
        <td>string-optional</td>
        <td>xs:string</td>
      </tr>
      <tr>
        <td>string-unbounded</td>
        <td>xs:string</td>
      </tr>
      <tr>
        <td>integer</td>
        <td>xs:integer</td>
      </tr>
      <tr>
        <td>positive-integer</td>
        <td>xs:positiveInteger</td>
      </tr>
      <tr>
        <td>short</td>
        <td>xs:short</td>
      </tr>
      <tr>
        <td>byte</td>
        <td>xs:byte</td>
      </tr>
      <tr>
        <td>unsignedByte</td>
        <td>xs:unsignedByte</td>
      </tr>
      <tr>
        <td>boolean</td>
        <td>xs:boolean</td>
      </tr>
    </table>
  </body>
</html>
```

```
        font-weight: bold;
        font-size: small;
        padding-right: 5px;
        width: 150px;
    }

    *:required {
        background-color: yellow;
    }

    *:invalid {
        background-color: pink;
    }
</style>
<xf:model id="test" schema="data-types.xsd">
    <xf:instance src="instance-data.xml"/>
</xf:model>
</head>
<body>
    <xf:group model="test" nodeset="/Data">
        <xf:input ref="date">
            <xf:label>Date:</xf:label>
        </xf:input>
        <xf:input ref="string-required">
            <xf:label>Required String:</xf:label>
        </xf:input>
        <xf:input model="test" ref="string-optional">
            <xf:label>Optional String:</xf:label>
        </xf:input>
        <xf:repeat nodeset="string-unbounded">
            <xf:input ref=".">
                <xf:label>Unbounded String:</xf:label>
            </xf:input>
        </xf:repeat>
        <xf:input ref="integer">
            <xf:label>Integer:</xf:label>
        </xf:input>
        <xf:input ref="positive-integer">
            <xf:label>Positive Integer:</xf:label>
        </xf:input>
        <xf:input ref="short">
            <xf:label>Short:</xf:label>
        </xf:input>
        <xf:input ref="byte">
            <xf:label>Byte:</xf:label>
        </xf:input>
        <xf:input ref="unsignedByte">
            <xf:label>Unsigned Byte:</xf:label>
        </xf:input>
        <xf:input ref="boolean">
            <xf:label>Boolean:</xf:label>
        </xf:input>
    </xf:group>
```

Load from XML Schema

```
</body>
</html>
```

XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Data">
    <xs:annotation>
      <xs:documentation>A listing of 10 XML Schema data types</xs:document-
ation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="date" type="xs:date"/>
        <xs:element name="string-required" type="xs:string"/>
        <xs:element name="string-optional" type="xs:string" minOccurs="0"/>
        <xs:element name="string-unbounded" type="xs:string" maxOccurs="un-
bounded"/>
        <xs:element name="integer" type="xs:integer"/>
        <xs:element name="positive-integer" type="xs:positiveInteger"/>
        <xs:element name="short" type="xs:short"/>
        <xs:element name="byte" type="xs:byte"/>
        <xs:element name="unsignedByte" type="xs:unsignedByte"/>
        <xs:element name="boolean" type="xs:boolean"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

XML Instance

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML instance file-->
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceS-
chemaLocation="data-types.xsd">
  <date>2006-09-21</date>
  <string-required>The quick brown fox jumped over the lazy dog.</string-re-
quired>
  <string-optional>This string is optional.</string-optional>
  <string-unbounded>One of many.</string-unbounded>
  <string-unbounded>Two of many.</string-unbounded>
  <string-unbounded>Three of many.</string-unbounded>
  <integer>-123</integer>
  <positive-integer>2</positive-integer>
```



```
<short>4096</short>
<byte>127</byte>
<unsignedByte>255</unsignedByte>
<boolean>true</boolean>
</Data>
```

Testing

One of the ways to test if the data types are loaded correctly from the XML Schema file is to attempt to enter an invalid value of a specific data type. XForms should automatically check to see if the input field meets the data requirements specified in the XML Schema file.

For example in the positive integer fields you can attempt to enter a "-1". After you do this and enter a "tab" the background of the form should display in pink as a warning. This is how the screen should look:

The way you display invalid controls is controlled by the following line in the CSS file:

```
*:invalid {
    background-color: pink;
}
```

This is known as a pseudo element. Although there are no data elements called "invalid", XForms just adds this property to each input control and the style sheet then adds whatever properties you give it in the style sheet.

Discussion

This program also uses a CSS to display the data elements.

Known bugs

Note that under the bindings do not work if there is no sample instance data.

In addition to this section, it may not be supported in other Xform platforms. Tested in XSLTForms, it does not work, with latest version as of 06/09/10 Used FormFaces, it worked, but could not render the css correctly.

There are also cross-browser compatibility issues as well.