Create an XML schema and generate Java beans

Contents

Create an XML schema and generate Java beans

This tutorial provides an overview of XML schema and Java™ beans. There are four lessons that will help you understand how to create and edit these types of documents using the XML schema editor.

Learning objectives

This tutorial is divided into several lessons that you should complete in sequence. While completing the lessons, you will perform these tasks:

- Create an XML schema
- Add components to your XML schema
- Explore the concept of refactoring
- Modify your XML schema's namespace prefix and target namespace
- Generate Java beans from your XML schema

Time Required

This tutorial should take approximately 30 minutes to finish. If you explore other concepts related to this tutorial, it could take longer to complete.

Related information

View the PDF version

Lesson 1.1: Create your XML schema

XML Schema files provide a set of rules which define the structure and semantics of an XML document.

Before you can create an XML schema and begin adding components to it, you must first create a project to contain your schema.

Switch to the Resource perspective

When developing XML projects, you will need to work in the Resources perspective. Follow these steps to switch to the Resource perspective:

1. From the menu bar, select **Window** → **Open Perspective** → **Other**. The Open Perspective dialog box opens.
2. Click **Resource**.
3. Click **OK**. The Resource perspective will open.

Create a project

You will now create a project to contain your XML schema by following these steps:

1. Click **File** → **New** → **Project**.
2. Expand the **General** folder and click **Project**, then click **Next**.
3. In the Project name field, type `XMLProject`.
4. Click **Finish**. Your project is displayed in the Navigator view.

Create an XML schema

You will now create a project to contain your XML schema by following these steps:

1. Click **File** → **New** → **Other**.

2. Expand the XML folder and click **XML Schema**. If you do not see the XML folder, select the **Show All Wizards** check box.
3. Click **Next**. If you are prompted to enable XML Development capabilities, click **OK**.
4. Select XMLProject and in the **File name** field, type `University.xsd`.
5. Click **Finish**.

The University.xsd file is created and opened in the XML schema editor.

Lesson checkpoint

The wizard allows you to quickly create a starting template for an XML Schema.

Now that you have completed this lesson, you should be able to:

- Switch perspectives
- Create a new project
- Create an XML Schema

Lesson 1.2: Add schema components

The XML schema editor provides the tools to easily generate XML schema components without possessing an in depth knowledge of XML schema semantics.

Before you begin, you must complete "Lesson 1.1: Create your XML schema" on page 1

The XML schema specification defines a large number of components such as complex types, simple types, groups, annotations, elements, attributes, and so on. To create a valid schema, you must understand the relationships between these components. For example, the <include>, <import> or <redefine> elements must appear before any other children of the schema element. An attribute can be added to a complex type, but not a simple type. There are many other relationships.

The XML schema editor removes the burden of remembering all these details. You can use the editor's Design view to add all of your schema components.

Add schema components

In these next steps, you will add several components to your schema, including simple and complex types, and global elements:

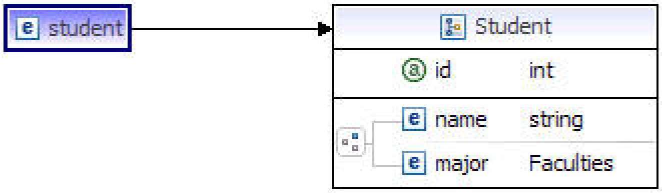
1. The University.xsd file needs to contain a complex type for defining student information and a simple type for listing the faculties in the university:
   a. In the Design view, right-click the Types category, and click **Add Complex Type**.
   b. Click NewXSDComplexType, and change its name to `Student`.
   c. Right-click the Types category, and click **Add Simple Type**.
   d. Click XSDSimpleType, and change its name to `Faculties`.
2. The Student complex type contains the student's name, the student's major, and the student's ID number. Add this information to the Student complex type by following these steps:
   a. To drill down to Student details, double-click the Student complex type.
   b. In the Student details view, right-click the Student complex type, and select **Add Sequence**. This will create a new content model object.
   c. In the Properties view, select all from the **Kind** drop-down list.

      Note: To view properties, click **Window** → **Show View** → **Properties**.
   d. Right-click the content model object, and click **Add Element**. A new element is created.

   e. Select the element and change its name to `name`. Note that the type of this element defaults to string.
   f. Right-click the Student complex type, and click **Add Attribute**. Change its name to `id`.
   g. Double-click the attribute type, and click **int** from the drop-down list.
   h. Right-click the content model object, and click **Add Element**. Change its name to `major`.
   i. Double-click the element type, and click **Browse** from the drop-down list.
   j. Select Faculties from the **Types** list, and click **OK**.

3. The simple type, Faculties, is derived by restriction from the simple type string and its value is limited to one of the faculties in the university. To define this list, you will create the enumeration facet as follows:
   a. Select the Faculties simple type.
   b. In the Properties view, click the **Constraints** tab.
   c. Click **Add** and type the new enumeration `science`.
   d. Repeat steps a through c to create two more enumerations, `arts` and `dentistry`, for the Faculties simple type.
4. To create an instance document from this XML schema, the XML schema must have a global element. You will now add a global element student as follows:
   a. In the Design view of the schema, right-click the Elements category and click **Add Element**. This will create a new element.
   b. Click NewElement, and change the name to `student`.
   c. Right-click the student element and select **Set Type** → **Browse**. Click Student from the **Types** list and click **OK**. Double-click the student element.

Your Design view should appear as shown below:



Lesson checkpoint

With the XML Schema editor it is possible to create valuable XML schema files without knowing all of the details and rules surrounding XML schema.

Now that you have finished, you should be able to complete the following:
- Add components to your XML schema
- Add information to a simple or complex type
- Create the enumeration facet to define a list of restrictions for a simple type.
- Add a global element to your XML schema

Lesson 1.3: Edit and validate your XML schema

Before you begin, you must complete “Lesson 1.2: Add schema components” on page 2.

Make changes and see how refactoring works

As a schema becomes bigger and more complex, it will have more type definitions, and references to those types. So what happens after you have defined a type, created ten references to that type, and you want to change the name of the type? The XML schema editor has a built-in refactoring mechanism that will propagate the changes automatically, meaning you do not have to do any manual updating. The following steps illustrate this feature.

In your University.xsd file, you have defined a simple type called **Faculties**. There is a reference to this type in the **Student** complex type. You have decided to change the name of the simple type to Faculty:

1. In the Design view, right-click the Faculties simple type, and select **Refactor** → **Rename**.
2. In the **New name** field, type `Faculty` and click **OK**.
3. Now switch to the Source view (this view enables you to see your source code). Notice that the type for the element major is changed to tns:Faculty automatically.

Change the namespace prefix and target namespace

A namespace provides a way to identify where an element or attribute comes from.

In the University.xsd schema, the target namespace is http://www.example.org/University. This is indicated by the targetNamespace attribute in the schema element. This means that all the types that are defined in this schema belong to the target namespace http://www.example.org/University.

The following line in the schema element defines the prefix tns for this target namespace:

```
xmlns:tns="http://www.example.org/University"
```

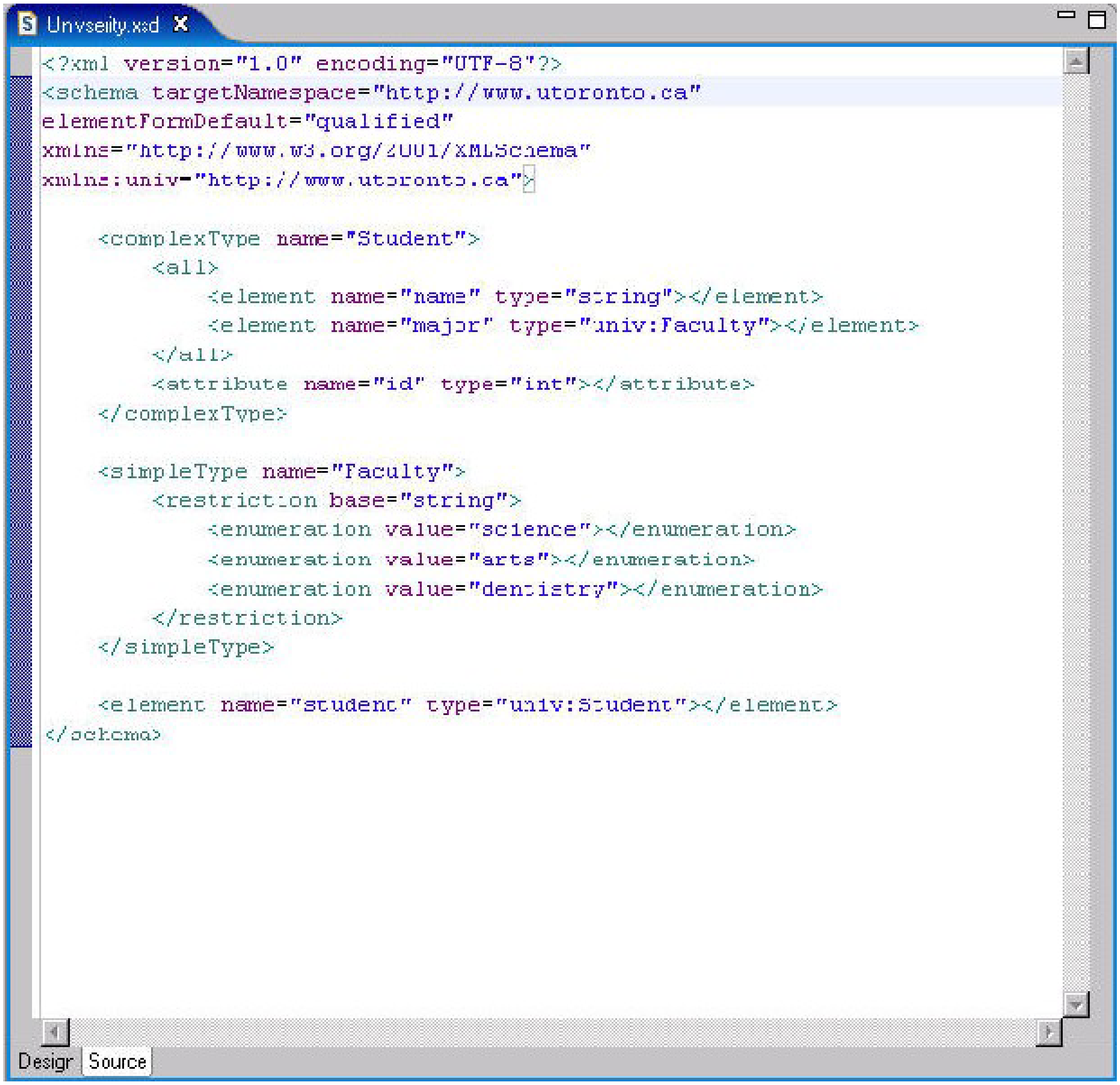
To refer to a type defined in this schema, you must use this defined prefix. Look at the Source view and note how the major element and the student element refers to the type as follows:

```
<element name="major" type="tns:Faculty"/>
<element name="student" type="tns:Student"/>
```

If you want to change the namespace prefix or the target namespace for your schema, you can use the Design view. Follow these steps:

1. In the Design view select the University schema.
2. Go to the Properties view and change the Prefix to `univ`.
3. Change the Target namespace to `http://www.utoronto.ca`.
4. Look at the Source View. Note that the attributes for the schema element and all the prefixes for the types are automatically changed for you.

Your code should now look similar to this:



```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.utoronto.ca"
elementFormDefault="qualified"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:univ="http://www.utoronto.ca">

    <complexType name="Student">
        <all>
            <element name="name" type="string"></element>
            <element name="major" type="univ:Faculty"></element>
        </all>
        <attribute name="id" type="int"></attribute>
    </complexType>

    <simpleType name="Faculty">
        <restriction base="string">
            <enumeration value="science"></enumeration>
            <enumeration value="arts"></enumeration>
            <enumeration value="dentistry"></enumeration>
        </restriction>
    </simpleType>

    <element name="student" type="univ:Student"></element>
</schema>
```

Validate your schema

Another useful feature of the XML schema editor is the incremental validation feature. After you save your XML schema file, you can right-click it in the Navigator view and click **Validate**. Any validation errors are reported in the Problems view with a red marker for the corresponding line in the Source view.

Lesson checkpoint

XML Schema Editor provides the tools to easily make changes which affect multiple dependencies, in a single step. In addition, the editor provides simple methods to specify namespace details, and verify file syntax.

In this lesson, you learned how to perform the following actions:

- Make changes and see how refactoring works
- Change the namespace prefix and target namespace
- Validate your XML schema

Lesson 1.4: Generate Java beans for your XML schema

To allow developers to quickly build an XML application, the XML schema editor supports the generation of Java beans from an XML schema. Using these Java beans, you can quickly create an instance document or load an instance document that conforms to the XML schema without coding directly to the Document Object Model (DOM) APIs.

Before you begin, you must complete "Lesson 1.3: Edit and validate your XML schema" on page 3.

Create a Java project to contain the beans

The beans you generate from University.xsd file must be contained in a project that is configured to work with Java source code.

1. Click **File** → **New** → **Other**. In the New window, select **Java** → **Java Project**. Click **Next**.
2. Type `UniversityJava` in the **Project name** field and click **Finish**.
3. You will be prompted to switch to the Java perspective. Click **Yes**.

Generate the beans

1. In the Navigator view, right-click University.xsd and select **Generate** → **Java**.
2. Ensure that SDO Beans Generator is selected in the Generator list and the UniversityJava project you just created is selected as the Container, then click **Finish**.

The beans appear in the Navigator view in the UniversityJava project. You can ignore any warnings that appear in the Problems view.

Lesson checkpoint

The XML Schema editor provides the necessary tools to utilize Java beans without necessarily having an in depth understanding of the DOM API.

This exercise teaches the basics of Java beans. Now that you have completed this lesson, you should be able to:

1. Create a Java project to contain the beans.
2. Generate the Java beans.

Summary

Congratulations! You have successfully created an XML schema, modified it, and generated Java beans from it.

Completed learning objectives

If you have completed all of the exercises, you should now understand how to:

- Create an XML schema
- Add components to your XML schema
- Understand the concept of refactoring
- Modify your XML schema's namespace prefix and target namespace
- Generate Java beans from your XML schema

More information

If you want to learn more about the topics covered in this tutorial, consider the following sources:

- The w3.org Web site: XML Schema Part 0: Primer is oriented towards quickly understanding how to create schemas using the XML schema language.
- The online help (Help > Help Contents) includes documentation for the XML schema editor.