



Generate and transform an XML file

Contents

Generate and transform an XML file. . . 1

Lesson 1.1: Create your DTD	1
Switch to the Resource perspective	1
Create a project	1
Create a DTD	1
Lesson checkpoint	2
Lesson 1.2: Customize your DTD	2
Create and name elements.	2
Edit an element's content model.	3
Edit an element's attributes	3
Edit a child element's content model	3
Add attributes.	4

Lesson checkpoint	4
Lesson 1.3: Create and edit your XML file	5
Generate an XML file based on your DTD	5
Edit in the Design view.	5
Edit in the Source view.	6
Lesson checkpoint	7
Lesson 1.4: Create and edit an XSL style sheet	7
Create the style sheet	7
Edit the style sheet	7
Transform the XML file to HTML	8
Lesson checkpoint	9
Summary	9

Generate and transform an XML file

This tutorial will help you to generate and transform an XML file using the XML editor to utilize DTD and XSL files.

Learning objectives

This tutorial is divided into several exercises you must complete in sequence. This tutorial teaches you to create an XML file based on a DTD and transform it using an XSL file. While completing the exercises, you will perform these tasks:

- Create a DTD
- Add and modify elements and attributes in a DTD
- Generate XML files from DTD files
- Explore the concept of grammar constraints
- Create and modify an XSL file
- Transform XML files using XSL files

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.

When you are ready, begin “Lesson 1.1: Create your DTD”

Related information

[View the PDF version](#)

Lesson 1.1: Create your DTD

DTD files provide a basic definition for an XML document by placing certain restrictions on the format of the file.

Before you can create your DTD file and begin adding components to it, you must first create a project to contain your DTD.

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

1. Click **File** → **New** → **Project**.
2. Expand the **General** folder and select **Project**. Click **Next**.
3. In the **Project name** field, type MyProject.
4. Click **Finish**.

Create a DTD

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

2. Expand the XML folder and select **DTD file**. 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 MyProject and in the **File name** field, type CDTitle.dtd.
5. Click **Finish**.

The CDTitle.dtd file is created and opened in the DTD editor.

Lesson checkpoint

The DTD editor allows you to create and edit a DTD file.

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

- Switch perspectives.
- Create a new project.
- Create a DTD file.

Lesson 1.2: Customize your DTD

This exercise teaches you how to create and modify elements and attributes in a DTD file.

Before you begin, you must complete Exercise 1.1: Creating your DTD.

In this lesson, you will define the following elements in your DTD:

Element	Description
CDLib	The CD library root element that contains all the other elements.
CD	A container element that contains information about a CD.
artist	An element that contains the name of a musician on the CD.
title	An attribute that describes the title of the CD.
format	An element that describes the number of CDs for this title.
description	An element that provides a description of the CD.
price	An element that provides the price of the CD.

Create and name elements

Follow these steps to add the elements listed in the table above to your DTD:

1. In the Outline view, right-click the DTD and click **Add Element**.
2. In the Properties view, change the name of the element to CDLib.
3. Right-click the DTD again, and click **Add Element**. In the Properties view, change the name of the element to CD.
4. Repeat step 3 to create the following elements: artist, format, description, and price.

Edit an element's content model

An element's content model is the representation of any data that might be contained inside an element. When you create an element, a node automatically appears in the Outline view to represent the content model type of the element, which by default is **EMPTY** (does not contain any other elements or text). You need to change the value of each element:

1. In the Outline view, under CDLib, select **EMPTY**. In the Properties view, click the **Other** tab. In the content type lists, select **CD**.
2. In the Outline view, under the parent element **CD** (not the child element under CDLib), select **EMPTY**.
3. In the Properties view, select **Children Content** from the **Content type** list. This adds a group node under the **CD** element.
4. Expand the group node and select **newChild**. Then, in the Properties view, select **artist** from the **Content Model** list.
5. Right-click the **group** node. From its pop-up menu, select **Add Element to Content Model**. This adds another child element.
6. Select the new child element. In the Properties view for it, select **format** from the **Content Model** list.
7. Repeat steps 5 and 6 to create **description** and **price** child elements.
8. Click the parent element named **artist** (not the child element under CD) and expand it.
9. Click **EMPTY**. In the Properties view, click the **Other** tab. From the content type list, select **(#PCDATA)**.
10. Repeat steps 8 and 9 for the **format**, **description**, and **price** elements.

You have now specified the content models for each of the elements. A CDLib element can only contain CD elements, whereas a CD element can contain the artist, format, description, and price elements. The artist, format, description, and price elements cannot contain any other elements, they can only contain text.

Since you wanted the CD element to be able to contain more than one element, you defined its **Content type** to be **Children Content**. This caused a group node to be automatically added to the **CD** element. The group node enables you to add as many elements to an element's content model as you like. It also determines whether the elements are sorted in sequence or are available as a choice (you can only select one option or the other) and how often the group of elements can occur.

Edit an element's attributes

A CD library typically contains more than one CD. To indicate this in your DTD, you need to change the content model for the **CDLib** element:

1. In the Outline view, select the **CD** child element that is under the **CDLib** element. In the Properties view, click the **Attributes** tab. The Properties view shows the occurrence of its content model to be **Just once** (that is, CDLib can only contain one CD element).
2. Click **One or more** from the **Occurrence** drop down list to change its content model to contain one or more CDs.
3. Review the contents of the Source view. The CDLib element should look like this: `<!ELEMENT CDLib (CD+)>`

The CD library can now contain as many CDs as you want to add to it.

Edit a child element's content model

You also need to make some changes to the **artist** and the **format** content models:

1. For a CD, there can be more than one contributing artist. Indicate this by selecting the **artist** child element (that is, the **artist** element that is listed under **CD**) and changing its occurrence to be **One or more**.
2. Most titles fit on one CD. For such titles, the **format** element is optional. Indicate this by selecting the **format** child element and change its occurrence to be **Optional**.

The resulting CD element should look like this (in the Source view):

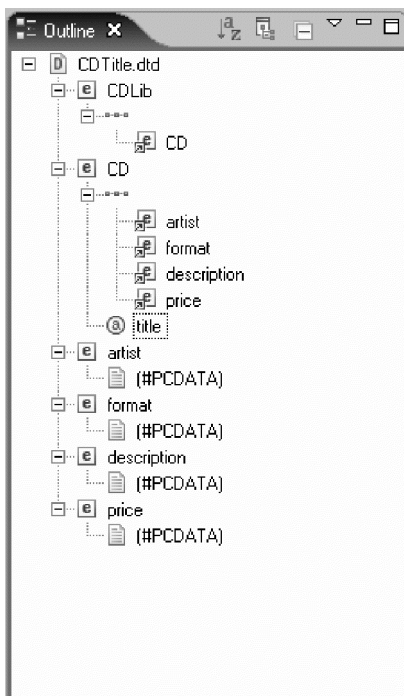
```
<!ELEMENT CD (artist+,format?,description,price)>
```

Add attributes

You now need to add the attribute **title** to the **CD** element (the parent element, not the child element under CDLib). To do so, follow these steps:

1. In the Outline view, right-click the CD parent element, and select **Add Attribute**. **Note:** All elements must have their attributes listed in an attribute list, therefore, if you add an attribute to an element that does not already have an attribute list, an attribute list will automatically be created and the new attribute added to it.
2. Expand the attribute list and select **NewAttribute**. In the Properties view, change the name of the attribute to **title**. The attribute type value defaults to Character Data (CDATA), and the attribute default value should already be set to #IMPLIED.
3. To save the DTD, select **File** → **Save**.

You have now completed your CDLib definition. It should look similar to the following:



Lesson checkpoint

The DTD editor allows you to efficiently create, and edit, elements and attributes.

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

- Create and name elements.
- Edit element's content models.
- Add attributes to elements.

Lesson 1.3: Create and edit your XML file

This section of the tutorial shows you how to generate an XML file based on your DTD (and associated with it) and then edit the XML file.

Before you begin, you must complete “Lesson 1.2: Customize your DTD” on page 2.

Generate an XML file based on your DTD

To generate an XML file from your DTD file follow these steps:

1. In the Navigator view, right-click the CDTitle.dtd that you created earlier, and from its pop-up menu click **Generate** → **XML File**.
2. Select your project and accept the default name. Click **Next**.
3. Accept the default values in the Select Root Element page. Your XML file will be generated with a root element of CDLib and any elements generated will contain sample data.
4. Click **Finish**.

The XML file appears in the Navigator view and automatically opens in the XML source page editor.

The XML file only contains the selected root element and any elements or attributes contained in the root element. You can now add elements, attributes, entities, and notations to the XML file, however, they must follow the rules established in the DTD that you used to create the XML file.

The format element is not included in the generated XML file because you did not select to create optional elements in the steps above.

Edit in the Design view

In the Design view, when you edit an XML file that has a set of constraints (or, a set of rules) defined by a DTD or an XML schema, you can turn those constraints on and off.

When constraints are turned on, you cannot make any changes that violate the rules of the DTD, such as adding an invalid element or removing necessary tags. Also, in this mode, you can use guided editing options.

If you turn constraints off, you can insert or delete any element or attribute regardless of the DTD rules. You may find it faster to work in this mode, but you may also introduce errors, so you should explicitly validate your XML file from time to time.

The CDTitle.xml file is constrained by the rules in the CDTitle.dtd file.

You are now going to edit the CDTitle.xml file in the Design view, at first with constraints on, and then constraints off:

1. If necessary, open CDTitle.xml in the XML editor (to do so, right-click it in the Navigator view and click **Open With** → **XML Editor**).
2. In the Design view, select the **CDLib** element and expand it.
3. Edit the contents of the **CD** element. Change the **artist** value to John Brown, the **description** to Opera and the **price** to 15.00.
4. CDTitle.xml currently only contains one CD. You are going to add another:
 - a. Right-click **CDLib** and select **Add Child** → **CD**.
 - b. A new **CD** element is automatically created, and it contains three elements: **artist**, **description**, and **price**.
 - c. Change the **artist** value to Jane Smith, the **description** to Classical and the **price** to 20.00.

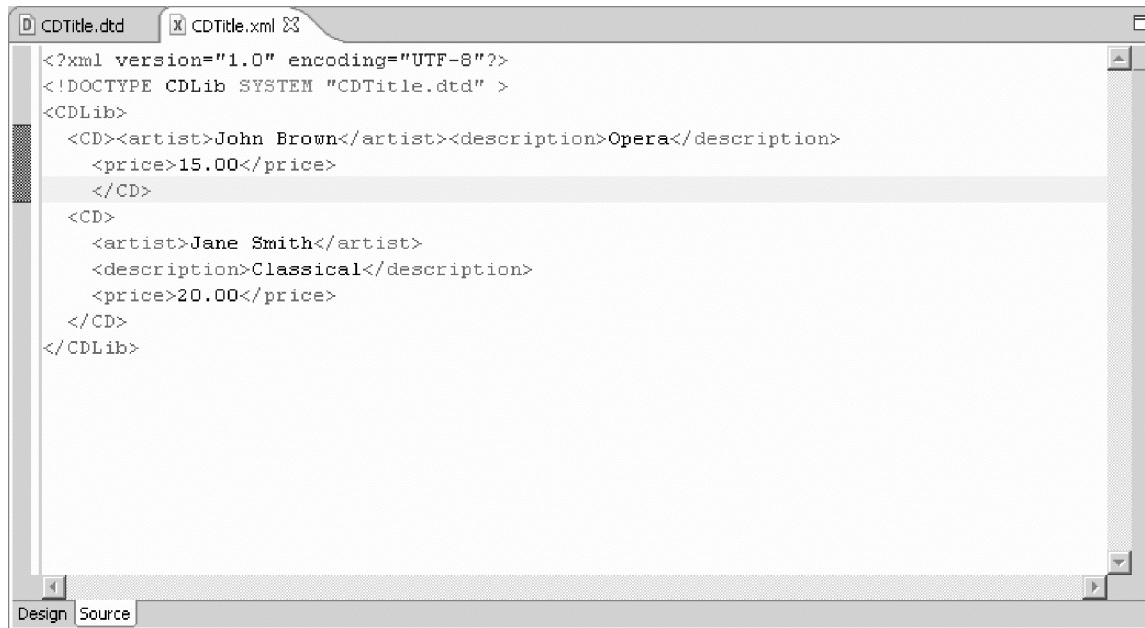
- d. Click **File** → **Save** to save your changes.
5. Try to add and remove certain elements and attributes from the first **CD** element:
 - a. Right-click **artist**. The menu option **Remove** is not available because **artist** is a required element. (You specified this in your DTD file.)
 - b. Right-click **description** and **price**. The menu option **Remove** is not available because they are also required items.
 - c. Right-click **CD** and click **Add Child**. You have the option to add another artist, but not another description or price. The option to add another description or price is not available, because in CDTitle.dtd, you specified that these items could only occur once inside a **CD** element. However, you specified that you could have one or more artist elements, which is why you can select to add another artist.
6. You are now going to turn the grammar constraints off, and continue to edit CDTitle.xml. Click **XML** → **Turn Grammar Constraints Off**.
7. Right-click **artist** in the first **CD** element and click **Remove**. You can remove this required element because the grammar constraints are off.
8. Right-click **description** in the first **CD** element and click **Remove**.
9. Right-click the first **CD** element and select **Add Child**. You now have the option to add another price. Click **price**.
10. Save your file. Note that you are allowed to save it even with all the syntactical errors. This is because you turned the constraints off.
11. In the Navigator view, right-click the XML file and click **Validate XML File**. The errors you have introduced into the file are listed in the Problems view.

Edit in the Source view

You are now going to edit the CDTitle.xml file in the Source view and fix the errors that you introduced into the file:

1. Select your XML file, and click **XML** → **Turn Grammar Constraints On** to turn the grammar constraints back on.
2. Click the **Source** tab to switch to the Source view.
3. Place your cursor after the opening tag for the first CD element: `<cd>`
4. Press Ctrl + Space to activate code assist. A pop-up list of available choices, which is based on the context, is displayed.
5. Double-click **artist**.
6. Type John Brown between the `<artist>` tags.
7. Now place your cursor after the artist tags and active code assist. Double-click **description**.
8. Type Opera between the description tags.
9. Select the code for the price element you have not modified and delete it.
10. Save your file.
11. In the Navigator view, right-click the XML file and select **Validate XML File**. The file is now valid and no errors appear in the Problems view for it.

The source code for your file should appear similar to the following:



Lesson checkpoint

The Design and Source views within the XML editor allow you to easily make changes to your XML file based on the constraints of the associated DTD. The grammar constraints option allows you to choose whether to apply the rules to the file at any given time.

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

- Generate an XML file based on your DTD.
- Edit the file in the Design view with grammar constraints on and off.
- Edit the file in the Source view.

Lesson 1.4: Create and edit an XSL style sheet

Style sheets are files that can be used to transform XML documents into other document types and to format the output. In this exercise, you will create a simple XSL style sheet to format your XML file data into a table in an HTML file.

Before you begin, you must complete Exercise 1.3: Creating and editing your XML file

Create the style sheet

Follow these steps to create a new, empty style sheet:

1. Select **File** → **New** → **Other**. In the New window, select **XML** → **XSL**. Click **Next**.
2. Select the MyProject directory.
3. In the File name field, type **CDTitle.xsl** and click **Next**.
4. In the Select XML file page, select the **CDTitle.xml** file. This associates the **CDTitle.xml** file with the **CDTitle.xsl** file.
5. Click **Finish**.

The **CDTitle.xsl** file is created and opens automatically in the XSL editor.

Edit the style sheet

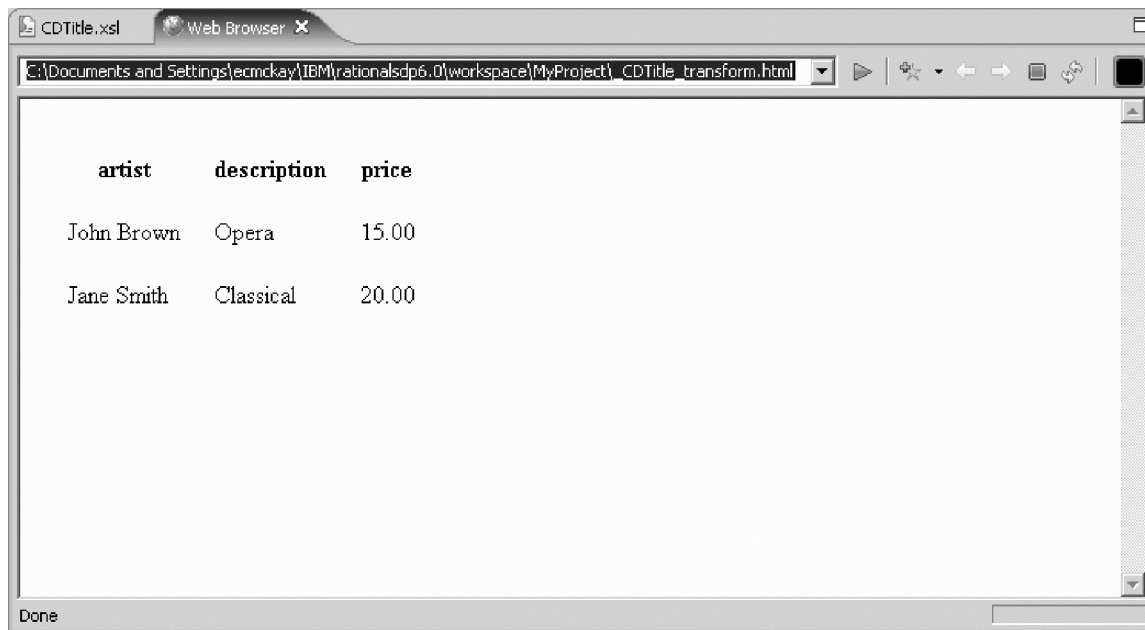
The XSL editor provides a number of wizards to help you create content in your style sheet.

1. To create an HTML document, you will add a template to generate HTML header information and define the output method for the document.
 - a. In the XSL editor, position the cursor on an empty line after the `<xsl:stylesheet>` element in the CDTitle.xsl file.
 - b. Click the **Snippets** tab and open the XSL drawer.
 - c. Double-click **Default HTML header**. This creates an `<xsl:output>` element that will output the results of the transformation (that is, applying CDTitle.xsl to CDTitle.xml) in HTML, and a template that will produce an HTML header with an `<xsl:apply templates>` rule to process all of the immediate children in the CDTitle.xml file.
2. Create a template that will produce an HTML table:
 - a. Position the cursor on an empty line after the `<xsl:template>` element that was generated in Step 1 above.
 - b. In the XSL drawer, double-click **HTML table in XSL**.
 - c. Select the first CD element as the context node for building the HTML table.
 - d. Click the **Wrap table in a template** check box. This indicates that you want to wrap this table in a new template.
 - e. Click the **Include header** check box to indicate that you want to include a header in this table. Click **Next**.
 - f. In the **Cell spacing** field, type 20. This will help the table look better by increasing the amount of space between the cells in your table.
 - g. Click **Finish**. This creates a new template that will produce an HTML table in your CDTitle.xsl file.
3. Click **File** → **Save** to save your file.

Transform the XML file to HTML

1. In the Navigator view, hold down the Ctrl key and select CDTitle.xml and CDTitle.xsl.
2. Right-click and select **Run** → **XSL Transformation**.
3. The result file name is _CDTitle_transform.html. It automatically opens in the Page Designer, and contains the contents of your XML file in a table.
4. To see the table as it will look in a Web browser, close the _CDTitle_transform.html file, right-click it in the Navigator view and select **Open With** → **Web Browser**.

It should appear as shown below:



Lesson checkpoint

The XSL editor provides a number of wizards to help you create a content filled style sheet. It also provides a method to create an HTML table based on that style sheet.

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

- Create an XSL style sheet.
- Fill the style sheet with contents which will generate an HTML table.
- Transform an XML file to HTML.

Summary

Congratulations! You have successfully completed these tasks:

- Created and customized a DTD.
- Generated an XML file from your DTD.
- Worked with grammar constraints.
- Created and edited an XSL file.
- Transformed your XML file using your XSL file.

Completed learning objectives

If you completed all of the exercises, you should now understand these tasks:

- Create a DTD.
- Add and modify elements and attributes in a DTD.
- Generate XML files from DTD files.
- Understand the concept of grammar constraints.
- Validate an XML file.
- Create and modify an XSL file.
- Transform XML files using XSL files.

More information

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

- The W3 Web site: www.w3.org/Style/XSL/ contains information on DTDs, XML files, and XSL files and transformation.
- The online help (**Help** → **Help Contents**) includes documentation for the DTD editor, the XML editor, and the XSL editor.