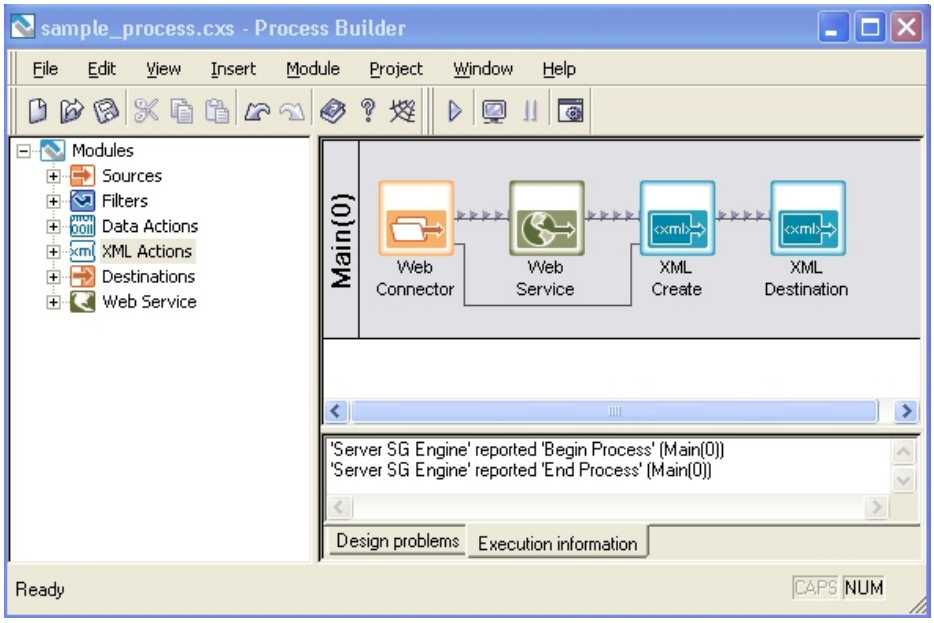
Corel® Smart Graphics Studio

Creating a sample XML file



Last update: June 26, 2003

Table of Contents

Creating a sample XML file

When you create a Corel Smart Graphics Studio project, you'll need to create a sample XML file to serve as a model for incoming data. The sample XML file is used when you map incoming data to the template.

In this lesson, you'll create a sample XML file based on Web Service data. You'll access the Web Service through the Process Builder.

You'll learn how to:

- use the Process Builder to connect to a Web service
- determine the input and output information you need
- structure your sample XML file

Introduction to the Process Builder

The Process Builder uses visual building blocks, called modules, to help you publish your Corel Smart Graphics Studio project to the Web through Corel Server SG.

Each module represents a project, action, or data. Modules let you hook up input and output streams to create a communication path.

The Process Builder workspace has three panes:

- Modules pane -- lists categories of modules
- Process pane -- where you work with the modules and hook up input and output streams
- Information pane -- displays information about the process (such as design problems or execution information).

To open the Process Builder

From the Windows® **Start** menu, click **All Programs** (Windows XP) **or Programs** (Windows 2000) > **Corel Smart Graphics > Process Builder.**

Connecting to a Web service

You'll connect to a Web service through the Process Builder and identify how the data is input and output. This information is used to build the sample XML file, but it's also used in the initial project planning stages as you decide what data you need to drive your dynamic graphics and build your template.

This lesson uses a Web service that provides information on atomic mass.

To add Web modules

1. In the Modules pane of the Process Builder, expand **Modules** > **Sources**, and double-click **Web Connector**. The **Web Connector** module appears in the **Process** pane.
2. In the Modules list, expand **Web Service**, and double-click **Web Service Consumer**.

To save your process

1. From the **File** menu, click **Save**.
2. Navigate to the folder in which you want to save the file. If you're working on a Smart Graphics project, this should be the folder you've specified as your project folder.
3. Type a **sample_process** in the **File Name** box.
4. Click **Save**.

The file is saved with the **.cxs** filename extension.

To configure the Web Connector module

1. In the **Process** pane, double-click the **Web Connector** module.
2. In the **Web Connector Setup** dialog box, click **Add**.
3. In the **New Data Property** dialog box, fill in the following boxes:
   - **Web variable name**-- type **sym** (an alias for your URL)
   - **Process property Name** -- type **sym** if it doesn't fill in automatically (this name appears as an output from the module)
   - **Data Type** -- choose **emstring** if it doesn't fill in automatically.
   - **Default Value** -- type **He** (the symbol for Helium)
4. Click **OK**, and click **OK** again to close both Web Connector dialog boxes.

To configure the Web Service Consumer module

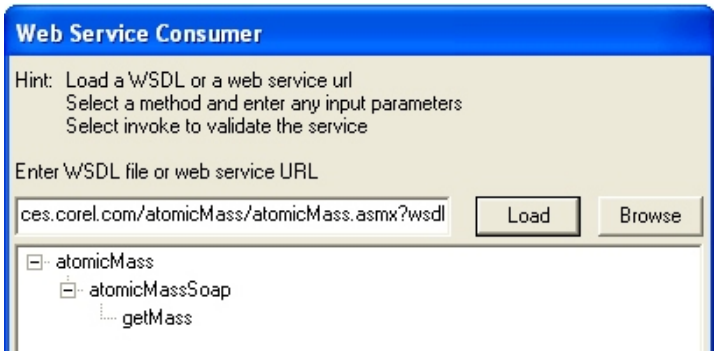
1. In the **Process** pane, double-click the **Web Service** module.
2. In the **Web Service Consumer** dialog box, enter the URL for the Web service you want to use in the **Enter WSDL file or Web service URL box**.

For this lesson, type:

`http://services.corel.com/atomicMass/atomicMass.asmx?wsdl`

3. Click **Load**.

   If the URL is valid, a structure appears in the window under the URL.



4. Select the **getMass** statement .

   In the **I/O Parameters** window, input and output information appears under **Parameter**, **Direction**, and **Type**.

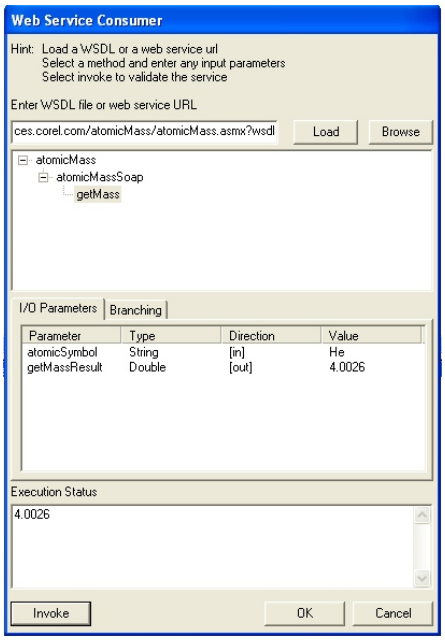
5. Click the **Value** column beside **{in}**.

   A box appears.

6. Type **He** in this box.

7. Click the **Invoke** button.

   If the **{in}** value entry is valid, the Web service returns a value beside **{out}** -- in this case, the atomic mass of Helium.



8. Click **OK**.

You now have the basic input and output data parameters you need.

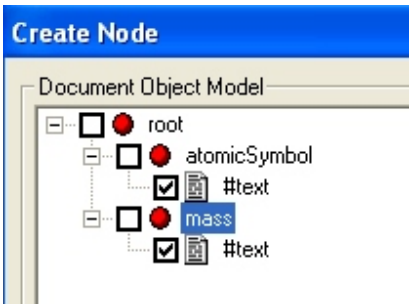
Defining the structure of a sample XML file

Using the input and output information you received from the Web service, you can now create a structure for your sample XML file.

To add an XML module

1. In the **Modules** pane, expand **XML Actions**, and double-click **XML Create**.
2. In the **Process** pane, double-click the **XML Create** module.
3. In the **Create Node** dialog box, click **Add Element**.
4. Click **Rename**, and type **root**.
5. Click **Add Element** to add a child element to the **root** element.
6. Select the child element and rename it **atomicSymbol**.
7. Select the **root** element, and click **Add Element** to add another child element.
8. Select the child element and rename it **mass** to reflect the data request (in this case, the **getMass** statement).
9. Select **atomicSymbol**, and click **Add Text Node** to enable text string input from the Web service.
10. Select **mass**, and click **Add Text Node**.
11. Enable the checkbox for each text node.

12. Click **OK**. Your structure should look like the following picture:



Note

- At this point, you could use the structure shown above to create an XML sample XML file in a text editor. Generating a sample XML file using the Process Builder is particularly effective when you are working with more complex projects and processes (for example, projects with more than one data source).

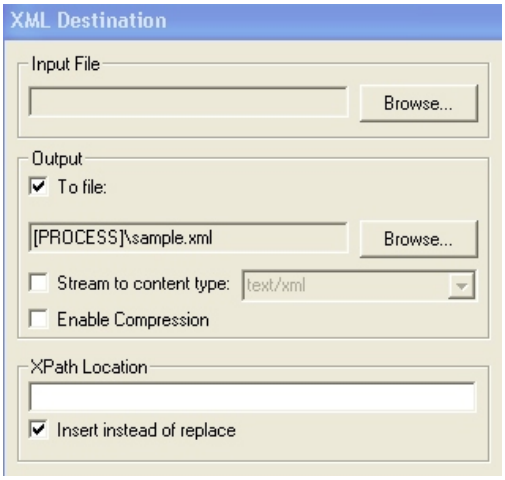
To create an XML file in the Process Builder

1. In the **Modules** pane, expand **XML Actions**, and double-click **XML Destination.**
2. In the **Process** pane, double-click the **XML Destination** module.
3. In the **XML Destination** dialog box, enable the **Output to File** checkbox and disable the **Stream to content type** check box.

   Outputting to file creates the sample XML file.
4. In the Output section, click **Browse** and in the **Virtual Path File Browser** dialog box, select a virtual path from the **Virtual Paths** list.

   If you have not created a virtual path for your project, click **Edit Paths**. In the **Virtual Paths** dialog box, click **New**, and type **PROCESS** (use uppercase letters) for the new path. Click the **Browse** button and navigate to your project folder (if you don't have a project folder yet, save the file to the location of your choice). Select the folder and click **OK**. Your new virtual path appears in the **Virtual Paths** dialog box. Select the path and click **OK** to return to the **Virtual Path File Browser** dialog box. Select your new virtual path from the **Virtual Paths** list.
5. Click **OK** to return to the **Virtual Path File Browser** dialog box.

6. Ensure that the Virtual Paths **PROCESS** is selected and in the **Filename** box, type **sample**. Your **XML Destination** dialog box should now include the information as indicated in the following picture:



7. Click **OK** to close the **XML Destination** dialog box.

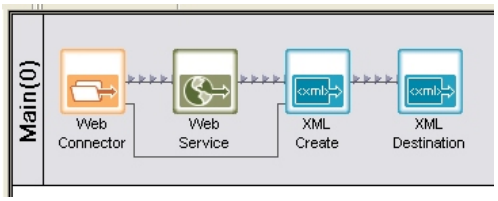
To link modules

1. Right-click the **Web Connector** module, and choose **Link Outputs** from the menu.

   The output options for the module appear in the **Outputs** window.

2. Click **sym.**
3. Point to the **Web Service** module to view the **Inputs** window.
4. Click **atomicSymbol**.
5. Continue to link the module outputs and inputs listed below:
   - **Web Service** module (**GetMassResult** ouput) to **XML Create** (**/root/mass/text()**)
   - **XML Create** module (**XML Node** output) to **XML Destination** module (**Master XML DOM** input)
   - **Web Connector** module (**Sym** output) to **XML Create** module (**/root/atomicSymbol/text()** input).

   Your **Process** pane should appear as in the following picture:



6. From the menu, click **File** and **Save**.

To run the process

1. In the Process Builder, click **Project** menu > **Execute**.
2. In the **Project Execute** dialog box, click **OK**.

   If the modules are configured and linked correctly, the following XML sample file (**sample**) is generated in your project folder:

```
- <root>
    <atomicSymbol>He</atomicSymbol>
    <mass>4.0026</mass>
  </root>
```

You can now use the sample XML file when you map your Corel Smart Graphics Studio project in the Data Mapper. Corel Smart Graphics Studio tutorials provide examples of how to use sample XML files.

You can save your process, to edit and build on when you create a project.