**Working with WSDLs**

WSDL files are central to testing SOAP-based services; they define the actual contract a service exposes and are required by SoapUI to generate tests, messages, validations and MockServices.

SoapUI supports the most widely used 1.1 version of WSDL and corresponding bindings for SOAP 1.1 and 1.2. The now-obsolete SOAP-Encoding standard (used for specifying XML contracts before XML Schema was formalized) is supported partially.

Let's dig in to WSDL related functionality in SoapUI by creating a new project and importing a simple publically available WSDL (http://www.webservicex.com/CurrencyConvertor.asmx?wsdl), create a new project in your SoapUI Workspace as below:
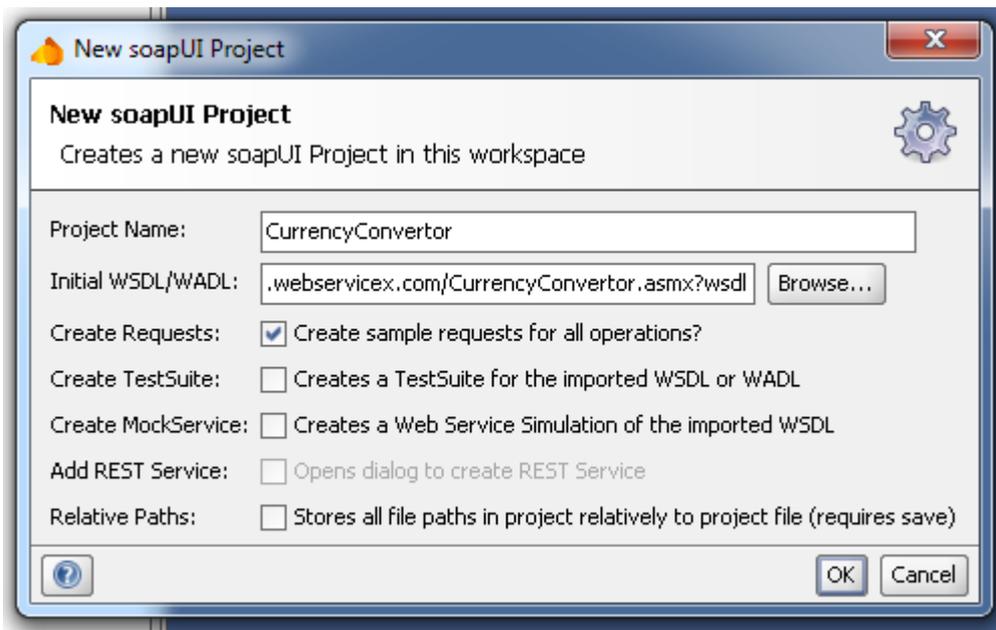
After pressing OK, SoapUI will load the specified WSDL and parse its contents into the following object model:
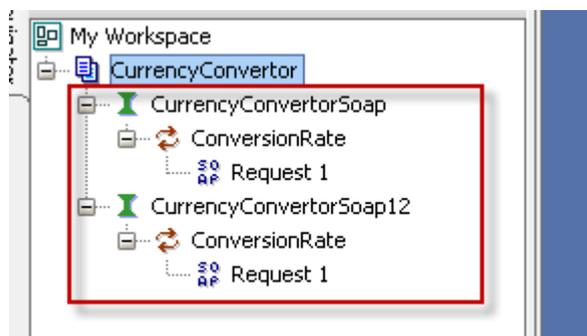

A WSDL can expose any number of services (bindings in WSDL-speak), which expose a certain contract ("portType") for a specified protocol; above we can see the same contract being exposed with two bindings, one for SOAP 1.1 ("CurrencyConverterSoap") and one for SOAP 1.2 ("CurrencyConverterSoap12").
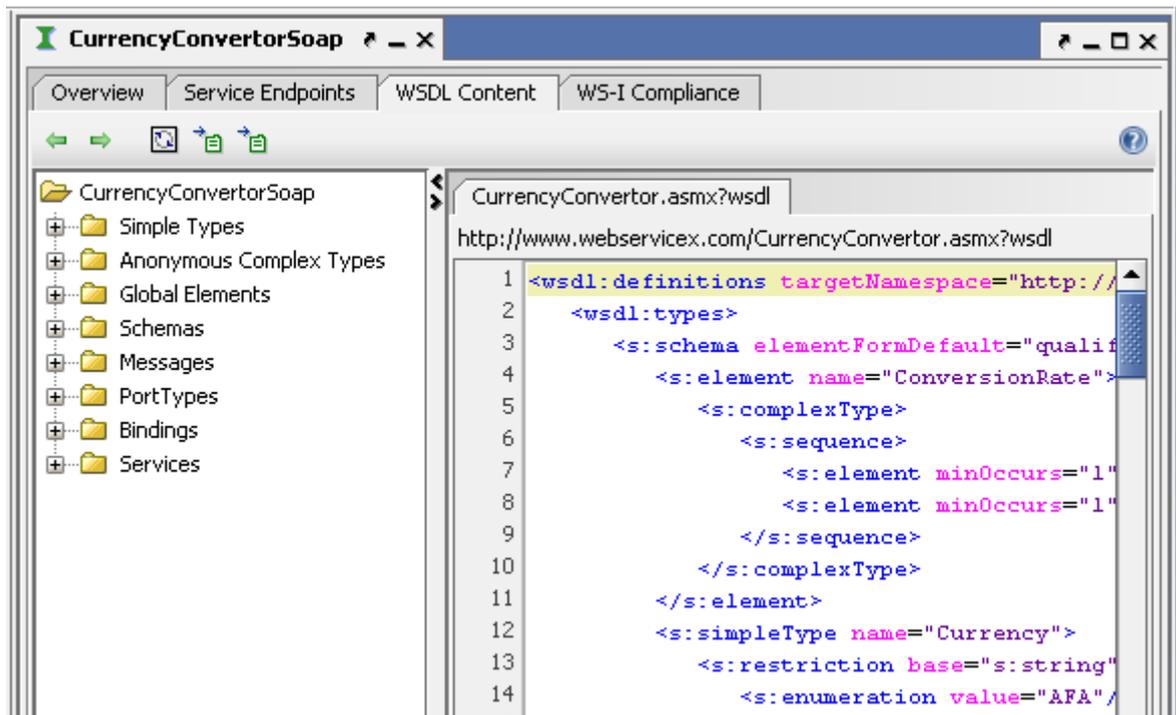
Tip: SoapUI Caches WSDLs in the project file to avoid unnecessary network access when opening and working with a project. If you want to disable this and force SoapUI to always use the remote WSDL for validations, etc then change the "Cache Definitions" bottom-left property for the containing project to false.

1. Working with WSDLs

Let's dig into this WSDL some more; double-click the first service icon in the navigator above which will open the following window:
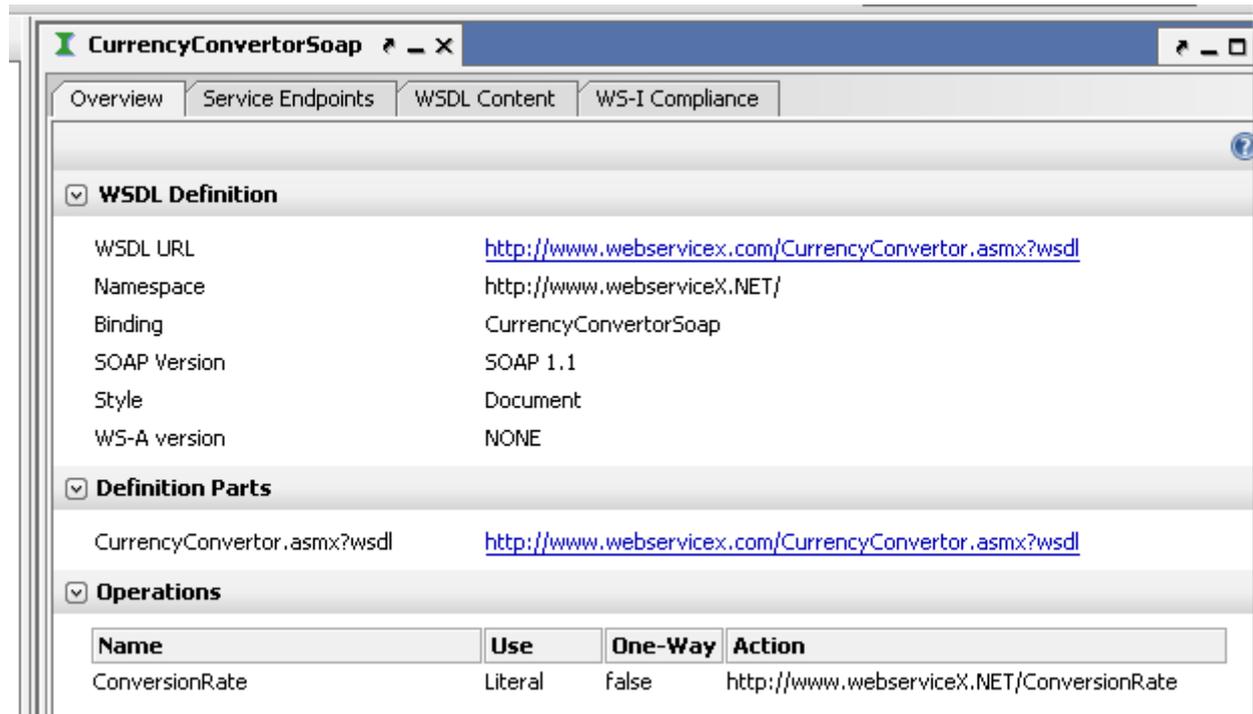
The first tab gives some general overview information on the WSDL; it's URL, target namespace, etc. Let's switch to the WSDL Content tab to have a more detailed look at the WSDL itself:

The navigation tree to the left allows us to browse through the contents of the WSDL, to the right we can see the WSDL file itself. Had there been multiple files involved (via WSDL imports or XML Schema imports and includes), these would all be shown as tabs to the right to allow browsing of the entire contract (with corresponding entries in the tree to the left).

The toolbar buttons include a "Create Documentation" option (second from the right), opening this and specifying a desired folder generates the following for us:

2. Validating the WSDL against the WS-I Basic Profile

Since the initial creation of WSDL and SOAP, a multitude of standards have been created and embodied in the Web Services domain, making it hard to agree on exactly how these standards should be used in a Web Service Context. To make interoperability between different Web Service vendors easier, the Web Service Interoperability Organization (WS-I; http://www.ws-i.org) has defined a set of rules mandating how the standards should be used, the "Basic Profile", which can be used both to validate WSDL contracts and SOAP messages. SoapUI bundles the 1.1 version of this profile, allowing you easily check the conformance of your WSDLs and messages from inside the tool;

- WSDL Contracts can be validated via the WSDL Service popup menu or WS-I Compliance tab in the WSDL Service window
- SOAP Messages can be validated via the right-click popup menu in the request and response message editors

Let's have a look at WSDL validation; select the "WS-I Compliance" tab in the WSDL Service window and press the green arrow on the top, a progress bar will be shown while the validation is being performed and the following report will be shown:

Scroll through the report to see its different sections, errors, etc.

While we're at it, let's try the message validation feature as well; open a request for any operation in the corresponding service and send it away (even if you'll get an error). Right click in the resulting XML response and select the "Check WS-I Compliance" option:
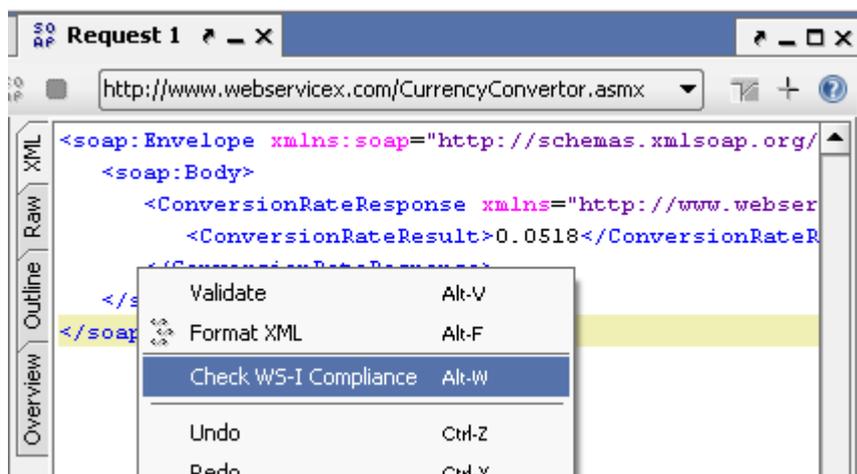
A corresponding report to the above will be generated highlighting any compliance errors for the current request/response message exchange.

# 3. Generating Code for your WSDL

Most Web-Service development frameworks allow you to generate code from a WSDL, either client code for calling the web service specified in the WSDL or server stubs for implementing the service(s) instead. To make this code generation easier and also allow for easy comparison between different frameworks, SoapUI provides a graphical front-end for most of them; right click the service for which you want to create code and select the "Generate Code" menu option:
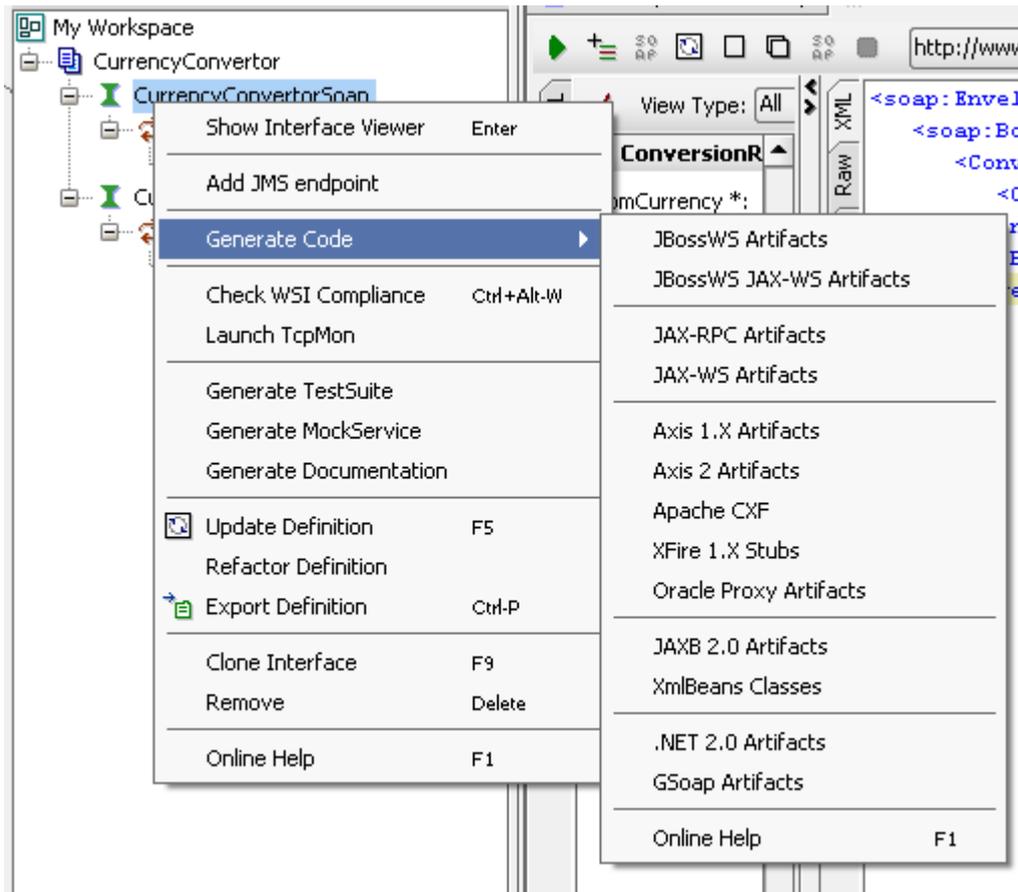

The underlying popup-menu shows all supported frameworks, selecting for example the "Apache CXF" option opens the following dialog with Apache CXF specific options:
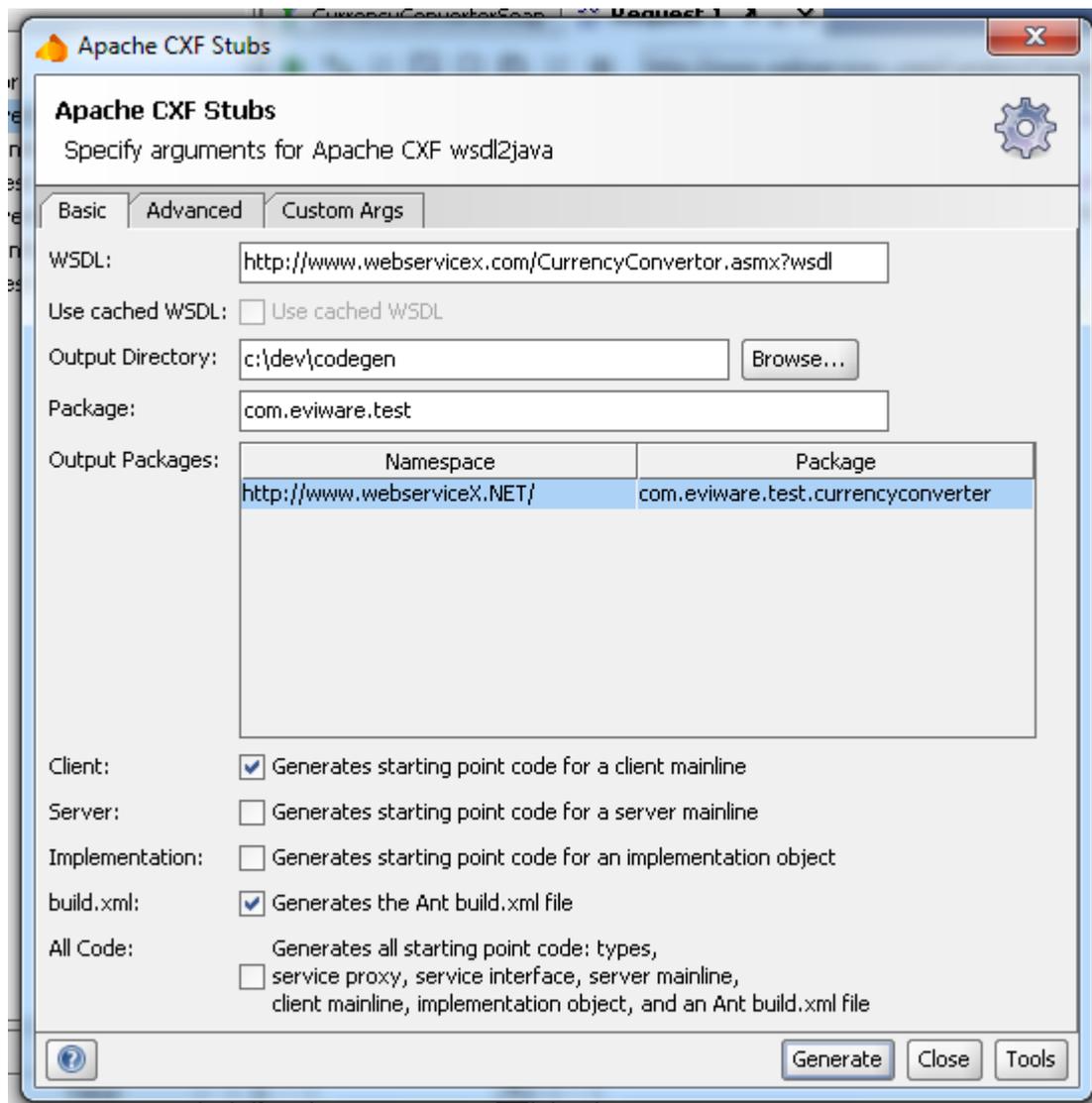
Fill in the desired settings and press the Generate button; SoapUI will launch the corresponding command-line tool (as configured in the Global Tool Integrations);
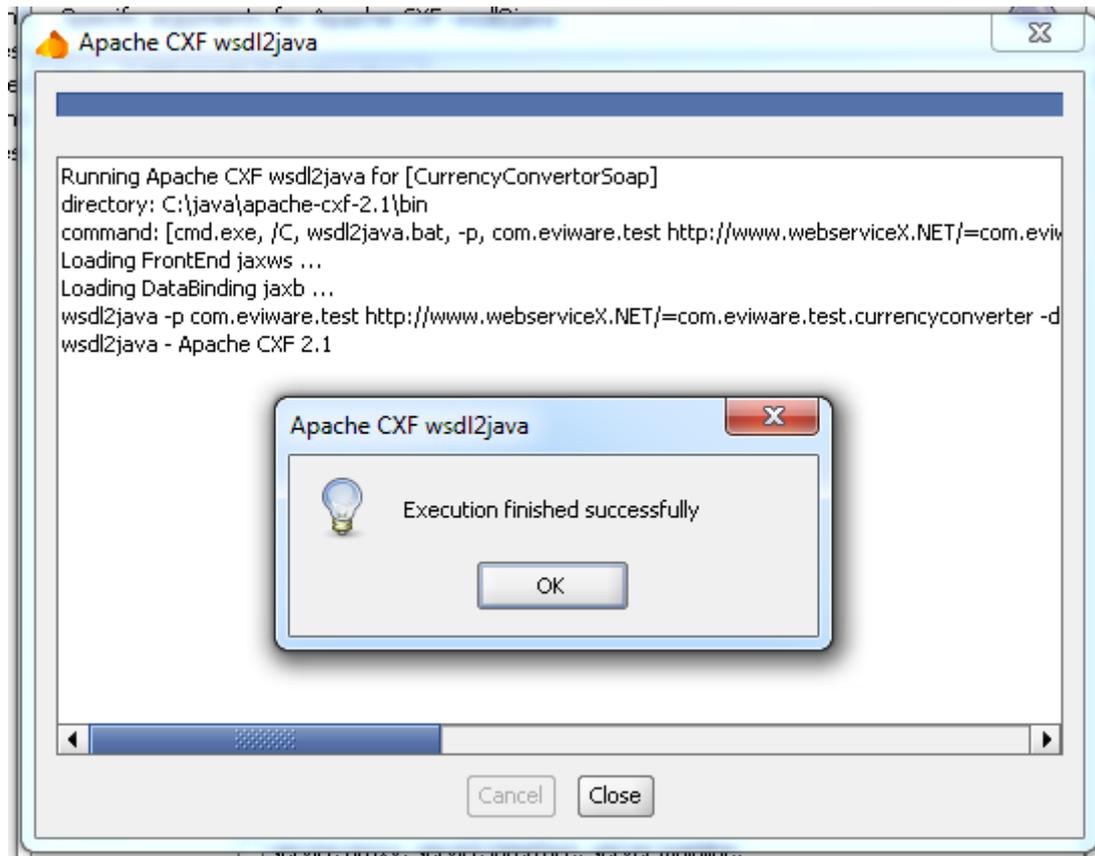
Checking in the specified folder in the file system reveals the generated files:
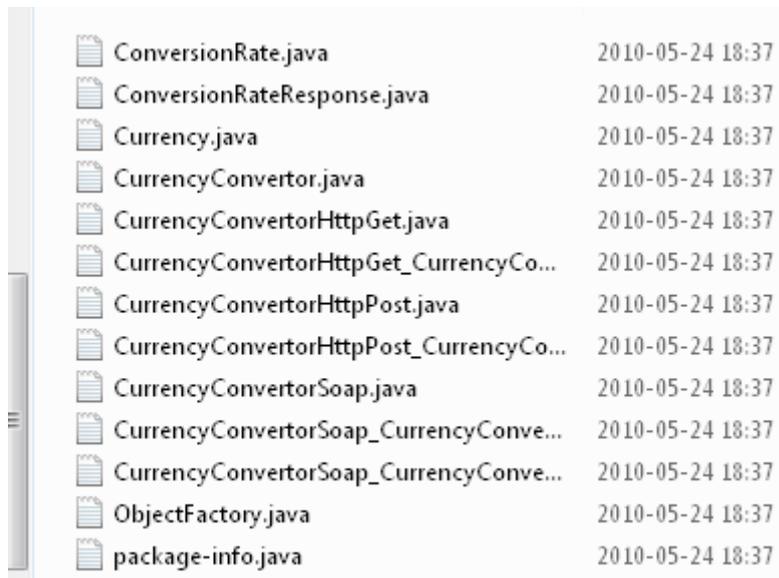
That's it for WSDL services for now, next up is how to work with requests!