

Bindings

Binding Types

SOAP Binding

The WSDL Importer supports SOAP bindings only. Each extension element of the SOAP binding is covered in subsequent sections.

HTTP GET & POST Binding

Not supported by the WSDL Importer yet. You will get an error message that ports having this binding are ignored. However, accessing such services can easily be done using the URL Adapter. Importing a WSDL file that contains services having both SOAP and HTTP GET & POST bindings will result in the import of the ports associated with the SOAP binding neglecting ports having different bindings.

MIME Binding

Not supported by the WSDL Importer yet. You will get an error message that ports having this binding are ignored. However, accessing such services can easily be done using the URL Adapter. Importing a WSDL file that contains services having both SOAP and MIME bindings will result in the import of the ports associated with the SOAP binding neglecting ports having different bindings.

List of all Binding Elements

soap:binding

The Importer supports two types of `soap:bindings`: document and RPC style. Examples of these two bindings are found in the following table:

Figure: Valid soap:binding examples and their UML counterparts

On this Page:

- [Binding Types](#)
 - [SOAP Binding](#)
 - [HTTP GET & POST Binding](#)
 - [MIME Binding](#)
- [List of all Binding Elements](#)
 - [soap:binding](#)
 - [soap:operation](#)
 - [soap:body](#)
 - [soap:fault](#)
 - [soap:header and soap:headerfault](#)
 - [soap:address](#)

<pre> <binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType"> <soap:binding style=" document" transport="http://schemas. xmlsoap.org/ soap/http"/> <operation name=" GetLastTradePrice"> <soap:operation/> <input> <soap:body use=" literal"/> </input> <output> <soap:body use=" literal"/> </output> </operation> </binding> </pre>	<pre> <binding name="GoogleSearchBinding" type="tns:GoogleSearchPortType"> <soap:binding style="rpc" transport="http://schemas.xmlsoap. org/soap/http"/> <operation name=" doGoogleSearch"> <soap:operation soapAction= "urn:GoogleSearchAction"/> <input> <soap:body use=" encoded" namespace="urn: GoogleSearch" encodingStyle= "http://schemas. xmlsoap. org/soap/encoding/" /> </input> <output> <soap:body use=" encoded" namespace="urn: GoogleSearch" encodingStyle= "http://schemas. xmlsoap. org/soap/encoding/" /> </output> </operation> </binding> </pre>

The encoding of the operation is given by the stereotype **<<DocumentEncoding>>** if the WSDL binding style is document. If we have an RPC style operation no stereotype is given. The Importer will report an error in all other cases.

The SOAP action value is put into the soapAction tagged value.

soap:operation

The `soap:operation` element contains the `soapAction` attribute (see [figure Valid soap:binding examples and their UML counterparts](#)). This attribute gets imported as `soapAction` tagged value.

soap:body

The `soap:body` element specifies how the message parts appear inside the SOAP Body element. The `soap:body` binding element provides information on how to assemble the different message parts inside the Body element of the SOAP message. The `soap:body` element is used in both RPC-oriented and document-oriented messages, but the style of the enclosing operation has important effects how the WSDL Importer handles the `soap:body` element:

- If the operation style is RPC each part of the associated message is a parameter or a return. These parts must have a type attribute referring to a SOAP encoded type schema. Thus, the Importer will report an error if the `soap:body` element does not contain the following attributes: `use="encoded"` `encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"`.

- If the operation style is Document each part of the associated message is an element of a XML Schema. These parts must have an element attribute referring to a literal XML Schema. Thus, the Importer will report an error if the soap:body element does not contain the following attributes: use="literal".

soap:fault

The `soap:fault` element specifies the contents of the SOAP Fault Details element. This element is not supported by the Importer. If a SOAP fault occurs, the Bridge generates internally an exception containing the fault code. This exception can then be handled by the modeler. The contents of the SOAP Fault Details element is put into the message part of the Bridge exception as plain string.

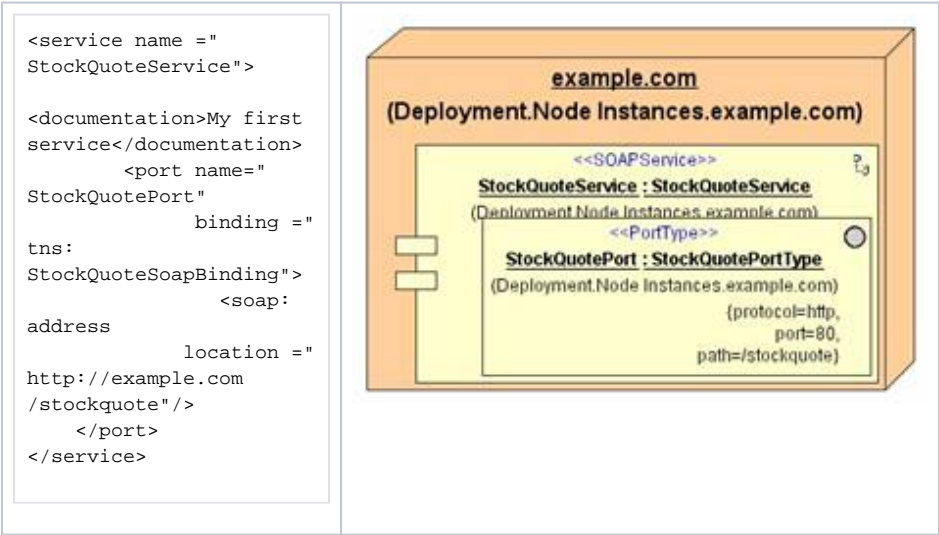
soap:header and soap:headerfault

Not supported by the WSDL Importer yet.

soap:address

The `soap:address` element defines the physical location of a port contained in a service. In UML, this information is expressed in deployment diagrams. The following table compares a WSDL fragment to the corresponding UML diagram:

Figure: `soap:address` and its Meaning for UML Deployment Diagrams



The above table shows that the `soap:address` location value is split into its parts. The protocol part becomes the protocol tagged value at port instance `StockQuotePort`. The network port becomes the port tagged value and the path information is put into yet another tagged value, namely `path`. The physical location encoded in the location attribute is mapped to an UML node instance having the name of a physical address, in this example: `example.com`.