WSDL Import Rules

On this Page:

Related Pages:

- SOAP Adapter
- Importing WSDL or XSD

The E2E WSDL Importer imports WSDL files and creates UML model elements corresponding to WSDL definitions.
Additionally, the importer generates classes and activity diagrams enabling the modeler to execute the imported services immediately. How these additional model elements are created is not explained in this section, because it follows directly the general Bridge modeling methodology discussed on other pages.

Find here how Web Service Definition Language (WSDL) entities are mapped to UML model elements. WSDL is a very generic description language allowing the description of Remote Procedure Call (RPC) interfaces as well as Document oriented message interfaces. Additionally, it allows for different type systems and different encoding schemes of these types. Before going into too much detail, we start by explaining the basic building blocks of WSDL and their significance for the WSDL Importer. For more information on how to import WSDL files with the WSDL Importer, see Importing WSDL or XSD. For more information on how to use imported service definitions, refer to SOAP Adapter.

SOAP Encoding Styles

SOAP uses an XML format to marshal data that is transported between software application. SOAP was planned to be used for legacy systems and modern object-oriented systems as well. Consequently, SOAP offers more than one encoding method to convert data from a software object into XML and vice versa (see the page of W3C).

There are two ways, in which it maps high level data types like arrays, integers, floats and so on to a serialized XML format: **SOAP encoding** (also called Section 5 encoding) and **Literal encoding** .
Literal encoding means that the body contents conform to a specific XML Schema. SOAP encoding uses a set of rules based on the XML Schema data types to encode the data, but the message does not conform to a particular schema.
In addition to the SOAP encoding styles, messages can be of two styles: **RPC (Remote Procedure Call) style** or **Document style**. The following encodings are commonly used:

- **SOAP Remote Procedure Call (RPC) encoding** (also known as Section 5 encoding, which is defined by the SOAP 1.1 specification).
  - SOAP encoding
  - RPC style messages
- **SOAP Remote Procedure Call Literal encoding (SOAP RPC-literal),** which uses RPC methods to make calls but uses an XML do-it-yourself method for marshalling the data.
  - literal encoding
  - RPC style messages
- **SOAP document-style encoding**, which is also known as message-style or document-literal encoding.
  - literal encoding
  - document style messages

The following table summarizes these combinations.

Ways to Map Data Types	**SOAP Encoding (also called Section 5 encoding)**		**Literal Encoding**	
Message Types	RPC style	Document style	RPC style	Document style
Name	SOAP RPC encoding	<commonly not used in SOAP applications>	SOAP RPC-literal	SOAP document-style encoding (or document-literal)
Supported by the Bridge	✓	n/a	✗	✓

Basic Building Blocks of Web Services

A WSDL document is simply a set of definitions. There is a definitions element at the root, and definitions inside. The grammar is as follows:

```
<wsdl:definitions name="nmtoken"? targetNamespace="uri"?>
   <import namespace="uri" location="uri"/>*
   <wsdl:documentation .... /> ?

   <wsdl:types> ?
      <wsdl:documentation .... />?
      <xsd:schema .... />*
      <-- extensibility element -> *
   </wsdl:types>

   <wsdl:message name="nmtoken"> *
      ...
   </wsdl:message>

   <wsdl:portType name="nmtoken">*
      ...
   </wsdl:portType>

   <wsdl:binding name="nmtoken" type="qname">*
      ...
   </wsdl:binding>

   <wsdl:service name="nmtoken"> *
      <wsdl:documentation ... />?
      <wsdl:port name="nmtoken" binding="qname"> *
        <wsdl:documentation ... /> ?
        <-- extensibility element ->
      </wsdl:port>
      <-- extensibility element ->
   </wsdl:service>

   <-- extensibility element -> *

 </wsdl:definitions>
```

Services are defined using the following elements:

Element	Description	More Information at ...
`documentation`	A container for human readable documentation. This element is ignored by the WSDL Importer.	Documentation Element
`import`	Enables the WSDL author to split the definitions into different files. The importer resolves the imports. This means, that the importer generates one XMI model file, even if the WSDL definitions are distributed over several files.	Import Element
`types`	Provides data type definitions used to describe the messages exchanged. Types are mapped to UML classes. Each schema contained in the types section is mapped to a corresponding package contained in the "Imported Types," root package of the generated UML model.	Types Element
`message`	Represents an abstract definition of the data being transmitted. A message consists of logical parts, each of that is associated with a definition within some type system. The message itself is not mapped to any UML model element but used to define UML operation parameters. The operation itself resides in UML interfaces and classes of stereotype <<PortType>>.	Message Element
`portType`	A set of abstract operations. Each operation refers to an input message and output messages. PortTypes are mapped to UML interfaces and classes of stereotype <<PortType>>.	
`binding`	Specifies concrete protocol and data format specifications for the operations and messages defined by a particular portType. Binding information is not mapped to particular UML element but found as annotations to UML <<PortType>> interfaces and <<SOAPService>> components.	Bindings

port	Specifies an address for a binding, thus defining a single communication endpoint. The port is not mapped to a particular UML model element, but its information defines the deployment diagram.	Ports
service	Is used to aggregate a set of related ports. Services are mapped to UML components of stereotype <<SOAPService>> if the containing ports have SOAP bindings. Otherwise the import ignores the ports.	Services

When importing a WSDL file the importer will generate a package structure that follows roughly the WSDL file structure. The type section corresponds to the **Imported Types** package, the services section corresponds to the **Imported Services** package. Within these packages each schema respectively each service has its own package that contains its elements.

The following topics explain the above mentioned WSDL elements and their mapping to UML model elements by the importer in more detail.