

WSDL Format Type Mapping Extension

1 Details

The format type mapping WSDL extension maps qnames to Java classes. It appears in the Java, EJB and Native JMS bindings.

The format type mapping extends WSDL with the following extensibility elements:

```
<format:typeMapping style="uri" encoding="..." />?
  <format:typeMap typeName="qname" | elementName="qname "
formatType="nmtoken" />*
</format:typeMapping>
```

Each element is described in detail below.

- **format:typeMapping** This element allows the specification of a mapping from abstract types used in WSDL message parts (within the abstract service description) to java types that can represent the same information. The style attribute is used to say something about the target type system (i.e. the native type system being used to represent the abstract information); in the case of the java type system the value here must be "Java". This use of this attribute allows this extension to be reused for other kinds of bindings. The encoding attribute must be a URI which is used to denote the specific way in which the native type corresponds to the abstract type. We define a special encoding, the "Java" encoding, with the rules that tell us how to create a java class corresponding to an abstract schema type that can be used through the WSDL java binding. Details on the "Java" encoding follow. Having the encoding attribute allows us to map abstract types to java types in other convenient ways by using a customized encoding.

Java encoding

The java encoding used by WSIF is unspecified. The reason it does not need to be specified is that encoding information is useful only when the information contained within the java object is transformed in some way - serialized to a SOAP message, or converted to a representation in another type system for example. If all we do with a WSIF message containing parts belonging to the java type system is invoke the corresponding java service, we don't need to do anything more than verify that each message part is represented using a java object of the correct type (as specified by the typemapping element in the java binding).

Of course, we envisage the need for transforming messages from one representation to

another, but we leave the complete specification of the java encoding for a later release.

- **format:typemap** Each typemap element maps an abstract type to a type in some more convenient type system; in the case of the Java, EJB and Native JMS bindings this is the java type system. The typeName attribute is a qualified name for the abstract type being mapped (this must be one of the predefined schema types recognised in WSDL, or a type defined under the <types> section in the WSDL). The elementName attribute may be used to specify an element instead of a type (since WSDL message parts can be described either way). The formatType attribute is the java type corresponding to that abstract type or element. Its values must be one of the primitive java types (char, byte, short, int, long, float, double) or the fully qualified name of a java class.