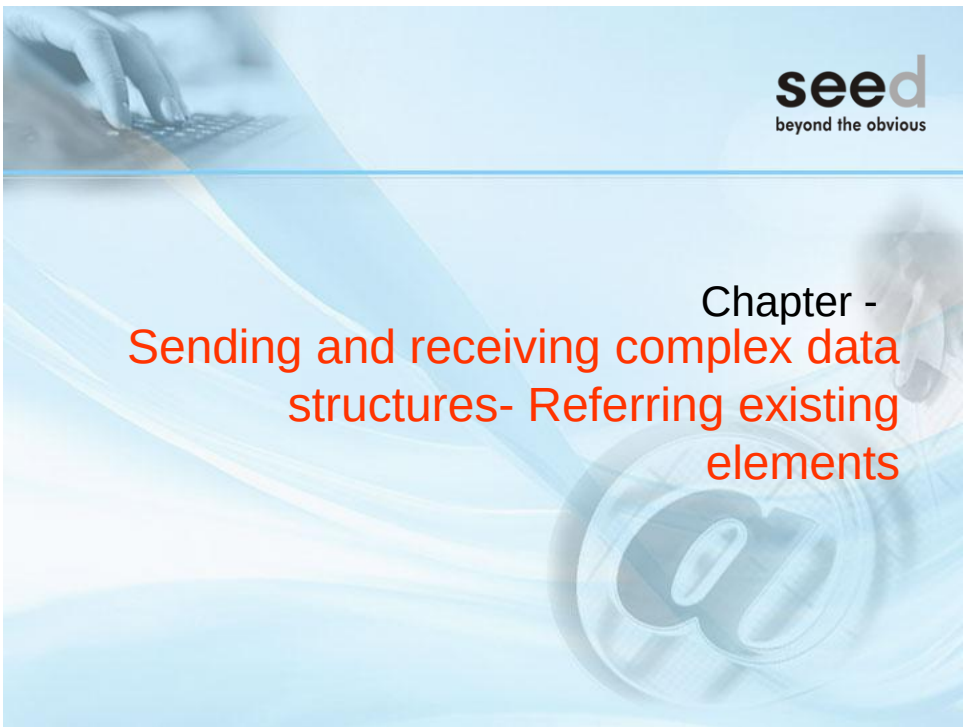


Objectives

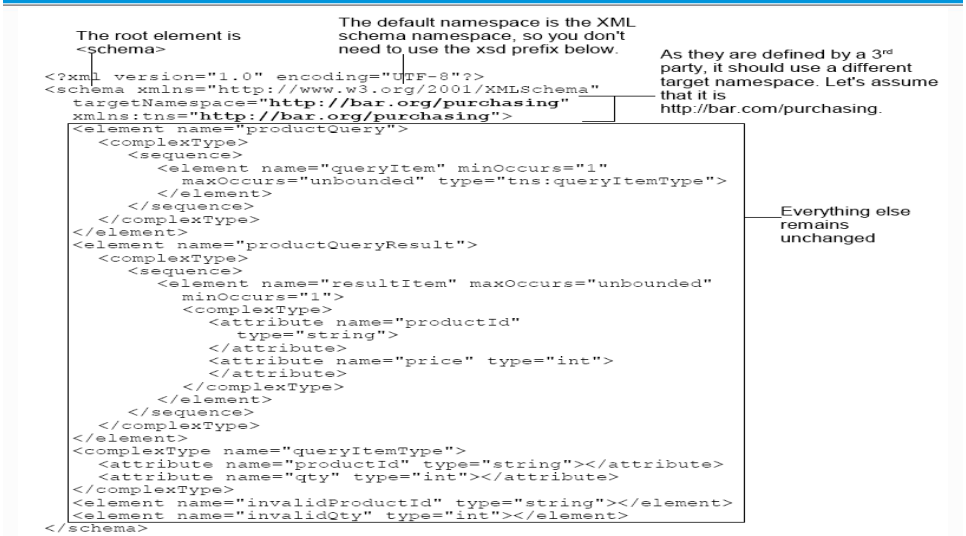
- At the end of this chapter you will be able to understand
  - How to send and receive complex data to and from a web service, which has been generated by consuming third party schemas

Referring to existing XML elements

- For the moment we're defining XML elements such as <*productQuery*> directly in the WSDL file.
- However, in practice, most likely such elements are defined by a 3rd party such as an industrial consortium or neutral association.
- Suppose that they are provided in a file *purchasing.xsd* such as this:

seed
beyond the obvious

Referring to existing XML elements




```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
   targetNamespace="http://bar.org/purchasing"
   xmlns:tns="http://bar.org/purchasing">
   <element name="productQuery">
      <complexType>
         <sequence>
            <element name="queryItem" minOccurs="1"
               maxOccurs="unbounded" type="tns:queryItemType">
            </element>
         </sequence>
      </complexType>
   </element>
   <element name="productQueryResult">
      <complexType>
         <sequence>
            <element name="resultItem" maxOccurs="unbounded"
               minOccurs="1">
               <complexType>
                  <attribute name="productId"
                     type="string">
                  </attribute>
                  <attribute name="price" type="int">
                  </attribute>
               </complexType>
            </element>
         </sequence>
      </complexType>
   </element>
   <complexType name="queryItemType">
      <attribute name="productId" type="string"></attribute>
      <attribute name="qty" type="int"></attribute>
   </complexType>
   <element name="invalidProductId" type="string"></element>
   <element name="invalidQty" type="int"></element>
</schema>
```


seed
beyond the obvious

Referring to existing XML elements

- How to refer to those XML elements in our WSDL file?

Referring to existing XML elements

- First, put the *purchasing.xsd* file into the same folder as the WSDL file (i.e., the project root).
- Then modify the WSDL file:

Referring to existing XML elements- wsdl

seed
beyond the obvious

Referring to existing XML elements

- Modify build.xml:

seed
beyond the obvious

Referring to existing XML elements- build.xml

```
<project ...>
  ...
  <target name="generate-service">
    <wsdl2code
      wsdlfilename="${name}.wsdl"
      serverside="true"
      generateservicexml="true"
      skipbuildxml="true"
      serversideinterface="true"
      namespacetopackages=
        "http://foo.com=com.ttdev.biz,http://bar.org/purchasing=com.ttdev.biz.purchasing"
      targetsourcefolderlocation="src"
      targetresourcesfolderlocation="src/META-INF"
      overwrite="true"/>
    <replaceregexp
      file="src/META-INF/services.xml"
      match="${name}Skeleton"
      replace="${name}Impl"/>
  </target>
  <target name="generate-client">
    <wsdl2code
      wsdlfilename="${name}.wsdl"
      skipbuildxml="true"
      namespacetopackages="http://foo.com=com.ttdev.biz.client"
      targetsourcefolderlocation="src"
      overwrite="true"/>
  </target>
</project>
```

As the XML elements are in the http://bar.org/purchasing namespace, you may want to map it to a Java package.

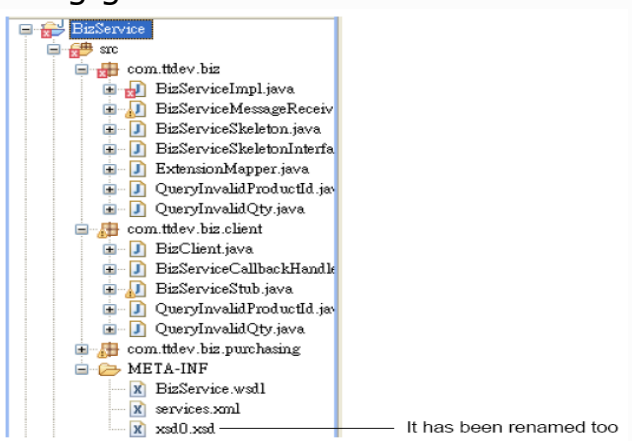
Separate them by a comma

You could do the same thing for the client, but by default the XML elements will be mapped to inner classes of the client stub. So you don't need to specify a package for them.

seed
beyond the obvious

Referring to existing XML elements

- Run the build and we will be able to see following generated files.



seed
beyond the obvious

Quick Recap . . .

- We can generate the web services based on a wsdl which import the elements from third party