BEA WebLogic Server R
Web Services XML Catalogs

Version: 10.3 Tech Preview

Document Date: October 2007

XML Catalogs Support

WebLogic Server now supports the XML catalogs (OASIS XML catalogs 1.1) option in (WLS) JAX-WS tools. The XML catalogs feature provides a way of referencing XML targets such as WSDLs and XSDs. Any included and/or imported XML targets may also be referenced via a catalog file. Sometimes it is convenient to refer to local resources instead of referring to resources over a network. For example, a WSDL may be accessible during client generation but not accessible when the client is run, this situation is solved by using XML catalogs. The JAX-WS runtime uses the XML catalogs for resolving a Web Service document that is a part of the description of a Web Service, specifically WSDL and XML Schema documents.

JAX-WS tools now have the -catalog option which turns on the use of catalog-based entity resolver. This option allows redirects of references to local copies of resources without touching the schema files.

The -catalog option has been added to the following Ant tasks:

- WsdlGet–Downloads a WSDL as well as its included and/or imported XMLs (XSDs and WSDLs) to local directory. The WSDL and its related XMLs may or may not be defined with a catalog. The task provides an ability to catch a snapshot of WSDL artifacts from remote network.
- Wsdlc–Downloads the referencing XMLs defined in catalog file to a compiled WSDL JAR or directory.
- clientGen–Used in two ways: standalone or embedded in a Jwsc task where a Web Service (client) is referenced in a JWS. Either way a jax-ws-catalog.xml file is put into the client runtime environment if the client artifacts are generated by clientGen with the -catalog option. When a Web Service client is packed in a web appliction, the jax-ws-catalog.xml file is put under WEB-INF/. Otherwise, it is put under META-INF/.

The following table illustrates how the -catalog option is supported by the Ant tasks for different types of Web Services.

Types→ Tasks ↓	**JAX-WS**	**JAX-RPC**
WsdlGet	Supported	Supported
Wsdl**c**	Supported	Supported
clientGen (for runtime)	Supported	Not supported

To use the XML catalogs feature a catalog file must be defined. For example:

```
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog"
prefer="system">

   <system systemId="http://foo.org/hello?wsdl"

                uri="HelloService.wsdl"/>

   <system systemId="http://helloservice.org/types/HelloTypes.xsd"

                uri="HelloTypes.xsd" />
```

```
        <public publicId="http://foo.org/jaxrpc?wsdl"
             uri="wsdl/jaxrpc/NoWebWS_JAXRPCService.wsdl"/>
</catalog>
```

Use the -catalog option to point to the catalog file in an Ant task. For example:

```
<target name="wsdlc">
    <wsdlc
         debug="true"
                  verbose="${verbose}"
         type="JAXWS"
         srcWsdl="${wsdl}"
                  destJwsDir="${cowDir}"
         destImplDir="${jwsImplDir}"
         catalog="mycatalog.xml">
   </wsdlc>
</target>
```

The XML Catalogs feature also supports a shortcut that directly defines a local reference entity in build.xml files in addition to or instead of a catalog file. For example:

```
<target name="wsdlc">
    <wsdlc
         debug="true"
                  verbose="${verbose}"
         type="JAXWS"
         srcWsdl="${wsdl}"
                  destJwsDir="${cowDir}"
         destImplDir="${jwsImplDir}"
         catalog="mycatalog.xml">
                  <xmlcatalog refid="localimportcatalog"/>
   </wsdlc>
</target>
<xmlcatalog id="localimportcatalog">
        <entity publicid="http://helloservice.org/types/HelloTypes.xsd"
location="${basedir}/HelloTypes.xsd"/>
</xmlcatalog>
```

Note: The local reference entity has higher priority that the same referenceID entity in a catalog file.

The following code examples illustrate using the -catalog option with clientGen.

Standalone Code Example

```
<target name="clientgen">
    <clientgen
        type="JAXWS" (only for JAXWS)
        wsdl="${wsdl}"
        destDir="${clientclasses.dir}"
   catalog="wsdlcatalog.xml"
        packageName="xmlcatalog.jaxws.clientgen.client" />
   [<xmlcatalog refid="wsimportcatalog"/>]  (optional)
</target>
```

The client runtime needs the the `jax-ws-catalog.xml` and its related XMLs under a specific directory (as defined by `destDir/META-INF/jax-ws-catalog.xml`). The directory needs to be defined in the client runtime `CLASSPATH`.

Embedded in jwsc Code Example

```
//case1: embedded in a JWS
<jwsc
  srcdir="${jwsImplDir}"
  destdir="${output.dir}/dog"
  keepGenerated="yes"  classpath="${cowDir}/HelloService_wsdl.jar">
      <jws file="org/helloservice/wsdl/HelloServiceImpl.java"  type="JAXWS"
(JWS can be JAXWS or JAXRPC) >
     <clientgen
       type="JAXWS"  (only for JAXWS)
       wsdl="http://xxx.wsdl"
       catalog="wsdlcatalog.xml"
       serviceName="NoWebWSService"
                packageName="xmlcatalog.jaxws.normal.client" />
            [<xmlcatalog refid="wsimportcatalog"/>]  (optional)
   </jws>
</jwsc>


//case2: embedded in a JWS moudle
<jwsc
  srcdir="${jwsImplDir}"
  destdir="${output.dir}/dog"
  keepGenerated="yes"  classpath="${cowDir}/HelloService_wsdl.jar">
  <module contextPath="test" name="myJar" explode="true" >
         <jws file="org/helloservice/wsdl/HelloServiceImpl.java" />
     <clientgen
       type="JAXWS"
       wsdl="http://xxx.wsdl"

       catalog="wsdlcatalog.xml"

       serviceName="NoWebWSService"

                packageName="xmlcatalog.jaxws.normal.client" />

      </module>

</jwsc>
```

Notice the difference in `Wsdlc(WsdlGet)` and `clientGen` when downloading referenced XMLs:

- The `Wsdlc(WsdlGet)`downloads all WSDLs, its imported/included XMLs, and some contents of the XMLs which may also contain include or import elements.
- `clientGen` only downloads those XMLs which are defined in a catalog file with a relative path and their contents are never changed.

A sample XML catalog file

```
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog"
prefer="system">

   <system systemId="http://foo.org/a?wsdl"

                 uri="a.wsdl"/>

   <system systemId="http://foo.org /b.xsd"

             uri="b.xsd" />

</catalog>
```

A sample wsdl file

```
<wsdl:definitions …>

  <wsdl :types>

  <xs:schema …>

      <xs:import namespace="http://xmlcatalog/clientgen"
schemaLocation="http://foo.org /b.xsd" />

     <xs:import namespace="http://xmlcatalog/clientgen"
schemaLocation="http://www.bea.com /remote.xsd" />

  </xs:schema>

….
```

The following table summarizes how `wsdlc` and `clientgen` work with the `–catalog` option.

Tasks → Items ↓	**wsdlc**	**clientgen**
Downloaded files	wsdls/ a.wsdl b.xsd remote.xsd (remote XML also downloaded)	META-INF/ a.wsdl b.xsd

Content of downloaded a.wsdl	`<wsdl:definitions …>`
	`<wsdl:types>`
	`<xs:schema ...>`
	`<xs:import namespace="http://xmlcatalog/clientgen" schemaLocation="./b.xsd" />`
		No any change in the original content

	<xs:import namespace="http://xmlcatalog/clientgen" schemaLocation="**./remote.xsd**" /> </xs:schema> ...	