

Schemas and Namespaces

Use of Schemas and Namespaces

CXF uses XML schemas extensively in the runtime: Its core includes the JAXB mappings of standard schemas such as <http://schemas.xmlsoap.org/wsdl/> and <http://www.w3.org/2006/03/addressing/ws-addr.xsd>. CXF also defines its own schemas for WSDL extensions such as the `jaxws-address` element in namespace <http://cxf.apache.org/transports/jms>, and schemas for both Spring and Blueprint (OSGi) configuration, for example the <http://cxf.apache.org/jaxws> and <http://cxf.apache.org/blueprint/jaxws> namespaces.

To avoid accessing resources over the network during a build, CXF maintains local copies of these schemas. Logically however, third party as well as CXF schemas are referred to by their public URIs in `schemaLocation` attributes and similar places. The CXF code generators use catalog files to map them to their actual location in the trunk, see the `pom.xml` file in the `api` module for an example.

Note if you're using Spring it will validate your Spring configuration files, using information in each JAR's `META-INF/spring.schemas` file (an example [here](#)). Validation is extremely helpful in diagnosing errors in CXF configurations, but it is also expensive at runtime. Once you have got your application working, you can disable validation by setting a system property: `-Dorg.apache.spring.validation.mode=VALIDATION_NONE`

The table below includes the URIs for the CXF configuration schemas that you need to specify in your Spring configuration file's `schemaLocation` attribute so that Spring's validating parser can validate the file. In theory you need not be concerned with the actual location of the files: Spring can normally detect their location from the classpath. And if you want to check the content of a schema file (in a binary distribution), you should be able to do so using its URI. You can also use the classpath location in the table below to find the original version of the schema in the trunk.

Configuration Schemas common to both Spring and Blueprint

URI	Target Namespace	Classpath Resource	Module
http://cxf.apache.org/schemas/configuration/http-conf.xsd	http://cxf.apache.org/transports/http/configuration	/schemas/configuration/http-conf.xsd	cxfrt-transports-http
http://cxf.apache.org/schemas/configuration/http-jetty.xsd	http://cxf.apache.org/transports/http-jetty/configuration	/schemas/configuration/http-jetty.xsd	cxfrt-transports-http-jetty
http://cxf.apache.org/schemas/configuration/jms.xsd	http://cxf.apache.org/transports/jms	/schemas/configuration/jms.xsd	cxfrt-transports-jms
http://cxf.apache.org/schemas/configuration/security.xsd	http://cxf.apache.org/configuration/security	/schemas/configuration/security.xsd	cxfrt-core
http://cxf.apache.org/schemas/configuration/soap.xsd	http://cxf.apache.org/bindings/soap	/schemas/configuration/soap.xsd	cxfrt-bindings-soap
http://cxf.apache.org/schemas/ws-addr-conf.xsd	http://cxf.apache.org/ws/addressing	/schemas/ws-addr-conf.xsd	cxfrt-ws-addr
http://cxf.apache.org/schemas/configuration/wsrmananager-types.xsd	http://cxf.apache.org/ws/rm/manager	/schemas/configuration/wsrmananager-types.xsd	cxfrt-ws-rm
http://cxf.apache.org/schemas/configuration/wsrmananager.xsd	http://cxf.apache.org/ws/rm/manager	/schemas/configuration/wsrmananager.xsd	cxfrt-ws-rm
http://cxf.apache.org/schemas/clustering.xsd	http://cxf.apache.org/clustering	/schemas/clustering.xsd	cxfrt-features-clustering

* `wsrm-manager.xsd` does an "xsd:include" on `wsrm-manager-types.xsd` as they are in the same namespace. As such, the user just needs to reference `wsrm-manager.xsd` to use both.

Spring-only Schemas

URI	Target Namespace	Classpath Resource	Module
http://cxf.apache.org/schemas/core.xsd	http://cxf.apache.org/core	/schemas/core.xsd	cxfrt-core
http://cxf.apache.org/schemas/jaxrs.xsd	http://cxf.apache.org/jaxrs	/schemas/jaxrs.xsd	cxfrt-frontend-jaxrs
http://cxf.apache.org/schemas/jaxws.xsd	http://cxf.apache.org/jaxws	/schemas/jaxws.xsd	cxfrt-frontend-jaxws
http://cxf.apache.org/schemas/policy.xsd	http://cxf.apache.org/policy	/schemas/policy.xsd	cxfrt-ws-policy
http://cxf.apache.org/schemas/simple.xsd	http://cxf.apache.org/simple	/schemas/simple.xsd	cxfrt-frontend-simple

Blueprint-only Schemas

URI	Target Namespace	Classpath Resource	Module
http://cxf.apache.org/schemas/blueprint/core.xsd	http://cxf.apache.org/blueprint/core	/schemas/blueprint/core.xsd	cxfrt-core
http://cxf.apache.org/schemas/blueprint/jaxrs.xsd	http://cxf.apache.org/blueprint/jaxrs	/schemas/blueprint/jaxrs.xsd	cxfrt-frontend-jaxrs
http://cxf.apache.org/schemas/blueprint/jaxws.xsd	http://cxf.apache.org/blueprint/jaxws	/schemas/blueprint/jaxws.xsd	cxfrt-frontend-jaxws
http://cxf.apache.org/schemas/blueprint/policy.xsd	http://cxf.apache.org/policy	/schemas/blueprint/policy.xsd	cxfrt-ws-policy
http://cxf.apache.org/schemas/blueprint/simple.xsd	http://cxf.apache.org/blueprint/simple	/schemas/blueprint/simple.xsd	cxfrt-frontend-simple

Example

The following is an example of a valid Spring configuration file. Using the table above, there is no magic involved in setting the correct value for the schemaLocation attribute!

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:http-conf="http://cxf.apache.org/transport/http/configuration"
  xmlns:jaxws="http://cxf.apache.org/jaxws"
  xsi:schemaLocation="
http://cxf.apache.org/transport/http/configuration http://cxf.apache.org/schemas/configuration/http-conf.xsd
http://cxf.apache.org/jaxws http://cxf.apache.org/schemas/jaxws.xsd
http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">

  <http-conf:conduit name="{http://cxf.apache.org/hello_world_soap_http}SoapPort.http-conduit">
    <http-conf:client DecoupledEndpoint="http://localhost:9999/decoupled_endpoint"/>
  </http-conf:conduit>

  <jaxws:client name="{http://cxf.apache.org/hello_world_soap_http}SoapPort" createdFromAPI="true">
    <jaxws:conduitSelector>
      <bean class="org.apache.cxf.endpoint.DeferredConduitSelector"/>
    </jaxws:conduitSelector>
  </jaxws:client>
</beans>
```

Other Schemas

The following schemas are imported by one or more of the schemas above, directly or indirectly. You may also find their classpath locations useful when you import or include any of the schemas below in your own schema, and want to know how you can access them locally.

URI	Target Namespace	Classpath Resource	Module
http://cxf.apache.org/schemas/configuration/cxf-beans.xsd	http://cxf.apache.org/configuration/beans	/schemas/configuration/cxf-beans.xsd	cxf-core
http://cxf.apache.org/schemas/wsd/http-conf.xsd	http://cxf.apache.org/transport/http/configuration	/schemas/wsd/http-conf.xsd	cxf-rt-transport-http
http://cxf.apache.org/schemas/wsd/jms.xsd	http://cxf.apache.org/transport/jms	/schemas/wsd/jms.xsd	cxf-rt-transport-jms
http://www.w3.org/2010/soapjms	http://www.w3.org/2010/soapjms	/schemas/wsd/spec/jms-spec-wsdl.xsd	cxf-rt-transport-jms
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd	/schemas/oasis-200401-wss-wssecurity-secext-1.0.xsd	cxf-rt-ws-policy
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	/schemas/oasis-200401-wss-wssecurity-utility-1.0.xsd	cxf-rt-ws-policy
http://schemas.xmlsoap.org/ws/2004/08/addressing	http://schemas.xmlsoap.org/ws/2004/08/addressing	/schemas/wsd/addressing.xsd	cxf-core
http://schemas.xmlsoap.org/ws/2004/09/policy/ws-policy.xsd	http://schemas.xmlsoap.org/ws/2004/09/policy	/schemas/ws-policy-200409.xsd	cxf-rt-ws-rm
http://schemas.xmlsoap.org/ws/2005/02/rm/wsr-policy.xsd	http://schemas.xmlsoap.org/ws/2005/02/rm/policy	/schemas/configuration/wsr-policy.xsd	cxf-rt-ws-rm
http://schemas.xmlsoap.org/wsd/	http://schemas.xmlsoap.org/wsd/	/schemas/wsd/wsd.xsd	cxf-core
http://schemas.xmlsoap.org/wsd/http/	http://schemas.xmlsoap.org/wsd/http/	/schemas/wsd/http.xsd	cxf-core
http://www.w3.org/2001/xml.xsd	http://www.w3.org/XML/1998/namespace	/schemas/xml.xsd	cxf-rt-ws-policy
http://www.w3.org/2006/07/ws-policy.xsd	http://www.w3.org/2006/07/ws-policy	/schemas/ws-policy-200607.xsd	cxf-rt-ws-policy

Note for Developers: If you define your own configuration schema, place it in the schemas subdirectory of the resources directory, then combine <http://cxf.apache.org/> and the path of the schema relative to the resources directory to form the system ID, and make the latter known to Spring by adding a line similar to the following to the **spring.schemas** file in your module's META-INF directory (note the escaped : character):

```
http\://cxf.apache.org/schemas/wsd/http-conf.xsd=schemas/wsd/http-conf.xsd
```