

WSPConfiguration

WS-Policy Framework Configuration

CXF's WS-Policy Framework is described in more detail in the [WS-Policy Framework User's Guide](#). Like many other features in CXF, the implementation is interceptor based, and thus the computation of the effective policy for a specific message (as well as the verification that one of the alternatives of that effective policy is supported) happens in interceptors. As there are quite a number of interceptors involved, these are not normally on the interceptor chains, i.e. the WS-Policy Framework is disabled. To enable it, use any of the following options:

Using the Policies Feature

The policies feature element is defined in namespace <http://cxf.apache.org/policy>. It supports one attribute:

Name	Value
ignoreUnknownAssertions	Indicates an exception should be thrown when encountering assertions for which no AssertionBuilders are registered (default: true). When set to false, a warning will be logged instead.

The element also support the the following child elements:

Name	Value
alternativeSelector	A bean or reference to a bean that implements the <code>org.apache.cxf.ws.policy.selector.AlternativeSelector</code> interface. The default selector chooses the minimal alternative, i.e. the one with the least number of assertions.

In addition, the element can have any number of `Policy` or `PolicyReference` child elements. This has the same effect as if the `Policy` or `PolicyReference` elements were attached to the `wsdl:port` element of the `wsdl` contract of the client or server endpoint to which the feature is applied (or to all endpoints if the feature is applied to the bus).

For example, to apply this feature to the bus and prevent exceptions being thrown when encountering unknown assertions:

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:cxf="http://cxf.apache.org/core"
       xmlns:p="http://cxf.apache.org/policy"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
http://cxf.apache.org/core http://cxf.apache.org/schemas/core.xsd
http://cxf.apache.org/policy http://cxf.apache.org/schemas/policy.xsd
http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
  <cxf:bus>
    <cxf:features>
      <p:policies ignoreUnknownAssertions="true"/>
    </cxf:features>
  </cxf:bus>
</beans>
```

Configuring the Policy Engine

The policy engine is the central instance of the WS-Policy framework implementation. You can enable and configure it directly, using the `<engine>` element in the <http://cxf.apache.org/policy> namespace. This element supports the same attributes and child elements as the `<policies>` element above, except:

- It supports the additional attribute 'enabled' of type boolean (default: false) to determine if the engine, and hence the policy framework, is enabled.
- It ignores `Policy` and `PolicyReference` child elements.

The following configuration achieves the same as in the example above:

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:cxf="http://cxf.apache.org/core"
       xmlns:p="http://cxf.apache.org/policy"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
http://cxf.apache.org/core http://cxf.apache.org/schemas/core.xsd
http://cxf.apache.org/policy http://cxf.apache.org/schemas/policy.xsd
http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">

    <p:engine enabled="true" ignoreUnknownAssertions="true"/>
</beans>
```

Note: Starting with CXF 2.2, the Policy Engine defaults to `enabled=true` and `ignoreUnknownAssertions=true`. Thus, the above configuration is no-longer needed as that is the default.

Specifying the Location of External Attachments

To specify the location of an external attachment that the policy framework should take into consideration when aggregating the policies applying to a specific message, you can use the `<externalAttachment>` element in the same namespace. It supports the following attributes:

Name	Value
location	Location of the external attachment document. This takes the form of a Spring Resource type property, e.g. 'classpath:etc/policies.xml' or 'file:/x1/resources/policies.xml'.

Example:

```
<p:externalAttachment location="classpath:org/apache/cxf/systest/ws/policy/addr-external.xml"/>
```

You can have any number of `<externalAttachment>` elements in your configuration file.