WSAConfiguration

WS-Addressing Configuration

In order to use WS-Addressing in a message exchange, two things are necessary:

1. The addressing interceptors (org.apache.cxf.ws.addressing.MAPAggregator and org.apache.cxf.ws.addressing.soap.MAPCodec) need to be on the inbound and outbound interceptor chains.
2. The use of WS-Addressing is indicated by one of the following:
    a. A <UsingAddressing xmlns="http://www.w3.org/2005/02/addressing/wsdl"> element is attached to the <wsdl:port>, <wsdl:service> or <wsdl:binding> element.
    b. The (chosen alternative for the) effective policy of the message contains a <Addressing xmlns="http://www.w3.org/2007/02/addressing/metadata"> assertion or a <UsingAddressing> assertion from either one of the following three namespaces: http://schemas.xmlsoap.org/ws/2004/08/addressing/policy, http://www.w3.org/2005/02/addressing/wsdl, http://www.w3.org/2006/05/addressing/wsdl.
    c. Property org.apache.cxf.ws.addressing.using in the message context is set to Boolean.TRUE.

Note that for 2.2 to take effect, CXF's policy engine must be enabled, see WS-Policy Framework Configuration.

Using the Addressing Feature

The addressing feature element is defined in namespace http://cxf.apache.org/ws/addressing. It supports two attributes:

Name	Value
allowDuplicates	A boolean that determines if duplicate MessageIDs are tolerated (default: true)
usingAddressingAdvisory	A boolean that indicates if the presence of the <UsingAddressing> element in the wsdl is purely advisory, i.e. its absence doesn't prevent the encoding of WS-A headers. This is especially useful to enable WS-Addressing in the java-first case, where you have no wsdl and hence none of the conditions in 2.1 and 2.2 will be met.

For example, to apply this feature to a JAX-WS server endpoint:

```
<beans ... xmlns:wsa="http://cxf.apache.org/ws/addressing" ...>
    <jaxws:endpoint ...>
        <jaxws:features>
            <wsa:addressing allowDuplicates="false"/>
        </jaxws:features>
    </bean>
</beans>
```

Adding the Addressing Interceptors Manually

org.apache.cxf.ws.addressing.MAPAggregator and org.apache.cxf.ws.addressing.soap.MAPCodec must be added to the interceptor chain for inbound and outbound messages and faults.
On a global level, i.e. applicable to all client and server endpoints, this can be done as in the example below (see also Bus Configuration). Note that, as allowDuplicates and usingAddressingAdvisory are actually properties of the MAPAggregator interceptor, they can also be set using Spring syntax.

```
<bean id="mapAggregator" class="org.apache.cxf.ws.addressing.MAPAggregator">
    <property name="allowDuplicates" value="false"/>
</bean>
<bean id="mapCodec" class="org.apache.cxf.ws.addressing.soap.MAPCodec"/>

<cxf:bus>
    <cxf:inInterceptors>
        <ref bean="mapAggregator"/>
        <ref bean="mapCodec"/>
    </cxf:inInterceptors>
    <cxf:inFaultInterceptors>
        <ref bean="mapAggregator"/>
        <ref bean="mapCodec"/>
    </cxf:inFaultInterceptors>
    <cxf:outInterceptors>
        <ref bean="mapAggregator"/>
        <ref bean="mapCodec"/>
    </cxf:outInterceptors>
    <cxf:outFaultInterceptors>
        <ref bean="mapAggregator"/>
        <ref bean="mapCodec"/>
    </cxf:outFaultInterceptors>
</cxf:bus>
```