



<a href="#">Home</a>	<a href="#">About</a>	<a href="#">Search</a>	<a href="#">Changes</a>	<a href="#">Unused Pages</a>	<a href="#">Undefined Pages</a>	<a href="#">Index</a>	<a href="#">License</a>
----------------------	-----------------------	------------------------	-------------------------	------------------------------	---------------------------------	-----------------------	-------------------------

[Edit this page](#)

# SOAPProcessor

Your trail: [WebservicesSupport](#)

JBoss Webservices SOAP Processor.

This action supports invocation of a [JBossWS](#) hosted webservice endpoint through any [JBossESB](#) hosted listener. This means the ESB can be used to expose Webservice endpoints for Services that don't already expose a Webservice endpoint. You can do this by writing a thin Service Wrapper Webservice (e.g. a JSR 181 implementation) that wraps calls to the target Service (that doesn't have a Webservice endpoint), exposing that Service via endpoints (listeners) running on the ESB. This also means that these Services are invocable over any transport channel supported by the ESB (http, ftp, jms etc).

## "ESB Message Aware" Webservice Endpoints

Note that Webservice endpoints exposed via this action have direct access to the current [JBossESB](#) Message instance used to invoke this action's `process(org.jboss.soa.esb.message.Message)` method. It can access the current Message instance via the `getMessage()` method and can change the Message instance via the `setMessage(org.jboss.soa.esb.message.Message)` method. This means that Webservice endpoints exposed via this action are "ESB Message Aware".

## Webservice Endpoint Deployment

Any [JBossWS](#) Webservice endpoint can be exposed via ESB listeners using this action. That includes endpoints that are deployed from inside (i.e. the Webservice .war is bundled inside the .esb) and outside (e.g. standalone Webservice .war deployments, Webservice .war deployments bundled inside a .ear) a .esb deployment.

## JAXB Introductions

The native [JBossWS](#) SOAP stack uses JAXB to bind to and from SOAP. This typically means that an unannotated typeset could not be used to build a JSR 181 endpoint on [JBossWS](#). To overcome this we use a [JBossESB](#) and [JBossWS](#) feature called "JAXB Introductions". This feature allows you define an XML configuration to "Introduce" the JAXB Annotations.

For details on how to configure [JBossWS](#) 2.0.x to use [JAXB Introductions](#), see [JAXB Introductions On JBossWS](#). The one exception to the instructions found there is that the `jboss-jaxb-intros.jar` file is already provided as part of the [JBossESB](#) distribution, so there's no need to build this lib. It can be found in the "extras/jaxbintros" directory in the [JBossESB](#) distribution.

## Action Configuration

The `<action ... />` configuration for this action is very straightforward. The action just takes one property value, which is the name of the [JBossWS](#) endpoint it's exposing (invoking).

```
<action name="ShippingProcessor" class="org.jboss.soa.esb.actions.soap.SOAPProcessor">
  <property name="jbossws-endpoint" value="ABI_Shipping"/>
</action>
```

## Quickstarts

A number of quickstarts that demonstrate how to use this action are available in the [JBossESB](#) distribution (samples/quickstarts). See the "webservice\_jbossws\_adapter\_01" and "webservice\_bpel" quickstarts.

---

[Go to top](#) [Edit this page](#) [More info...](#) [Attach file...](#)

*This page last changed on [04-Jul-2007 10:33:59 EDT](#) by [tfennelly](#).*



© 2007 Red Hat Middleware, LLC. All rights reserved. [Privacy Policy](#)