



PEtALS-SE-XSLT

This document explain how to install and configure the petals-se-xslt JBI component.

PEtALS Team

Marie SAUVAGE <>

NAUDIN Roland <roland.naudin@ebmwebsourcing.com>

LOUIS Adrien <adrien.louis@ebmwebsourcing.com>

- Mars 2009 -



(CC) EBM WebSourcing - This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>



Table of Contents

PEtALS-SE-XSLT	5
1. Component Configuration	6
2. Service Configuration	8
2.1. Transform XML messages	8
2.1.1. Service Unit descriptor	8
2.1.2. Service Unit content	9
2.1.3. Provider restrictions	10
2.1.4. Provider Usage	10

List of Figures

2.1. The XSLT Service Engine	8
------------------------------------	---

List of Tables

1.1. Configuration of the component (CDK)	6
2.1. Configuration of a Service Unit to provide a service (JBI)	9
2.2. Configuration of a Service Unit to provide a service (CDK)	9
2.3. Configuration of a Service Unit to provide a service (XSLT)	9

PEtALS-SE-XSLT

This component allows to process xml transformations based on xsl style sheet. It creates an XML output from a given XML source content and a XSL stylesheet defined in the JBI description of a Service Unit.

It is based on the PEtALS CDK.

This component provides only services and doesn't act as a consumer of service.

For more details about XSLT usage, consult url : <http://www.w3.org/TR/xslt>

Chapter 1. Component Configuration

no specific configuration for this component

Table 1.1. Configuration of the component (CDK)

Parameter	Description	Default	Required	Scope
acceptor-pool-size	The size of the thread pool used to accept Message Exchange from the NMR. Once a message is accepted, its processing is delegated to the processor pool thread.	5	Yes	Runtime
processor-pool-size	The size of the thread pool used to process Message Exchanges. Once a message is accepted, its processing is delegated to one of the thread of this pool.	10	Yes	Runtime
performance-notifications	Enable the performance notifications in the component. The CDK proposes to a performance notification feature to the component implementor. If you enable this feature, you must use the related method accessible in the <code>AbstractComponent</code> class.	-	No	Runtime
performance-step	When the performance notification feature is enabled, it is possible to define a step on the notifications. When there is an heavy message traffic, it is recommended to increase this step to avoid performance disturbance.	-	No	Runtime
properties-file	Name of the file containing properties used as reference by other parameters. Parameters reference the property name in the following pattern <code>\${myPropertyName}</code> . At runtime, the expression is replaced by the value of the property. The value of this parameter is : <ul style="list-style-type: none">• an URL• a file relative to the PEtALS installation path• an empty value to stipulate a non-using file	-	No	Installation
ignored-status	When the component receives an acknowledgement message exchange, it can skip the processing of these message according to the type of the acknowledgment. If you decide to not ignore some acknowledgement, the component listeners must take care of them. Accepted values : <code>DONE_AND_ERROR_IGNORED</code> , <code>DONE_IGNORED</code> , <code>ERROR_IGNORED</code> OR <code>NOTHING_IGNORED</code>	<code>DONE_AND_ERROR_IGNORED</code>	Yes	Component
jbi-listener-class-name	Qualified name of the class extending AbstractJBIListener	-	Yes	Component
external-listener-class-name	Qualified name of the class extending AbstractExternalListener	-	No	Component

Definition of CDK parameter scope :

- *Component* : The parameter has been defined during the development of the component. A user of the component can not change its value.
- *Installation*: The parameter can be set during the installation of the component, by using the installation MBean (see JBI specifications for details about the installation sequence). If the parameter is optional and has not been defined during the development of the component, it is not available at installation time.

- *Runtime* : The parameter can be set during the installation of the component and during runtime. The runtime configuration can be changed using the CDK custom MBean named `RuntimeConfiguration`. If the parameter is optional and has not been defined during the development of the component, it is not available at installation and runtime times.

Chapter 2. Service Configuration

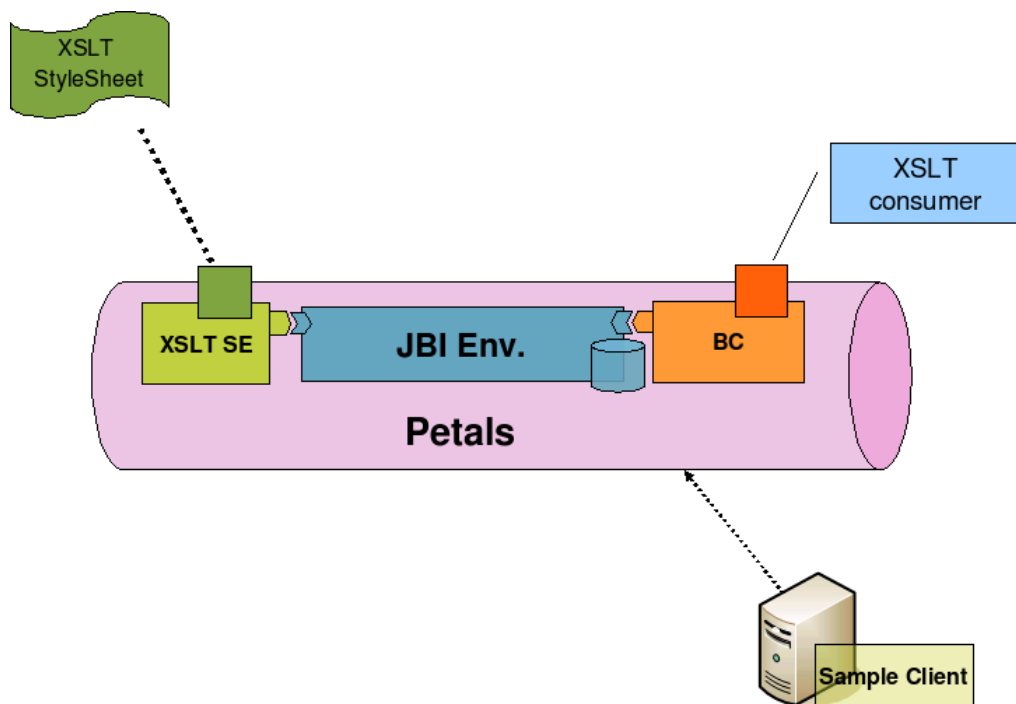
2.1. Transform XML messages

PROVIDE SERVICE : Expose an external service in the JBI environment

2.1.1. Service Unit descriptor

The XSLT component proposes services to transform an incoming document to a specific form. The transformation occurs only on the source of the incoming message. The resulted message can be returned as a source or as an attachment.

Figure 2.1. The XSLT Service Engine



To expose your transformation into the JBI bus, you must deploy a XSLT Service Unit with a JBI descriptor containing a `provides` section and your XSL stylesheet. An new endpoint would be activated in the bus at the deployment of the Service Unit.

You can define several `provides` section in the XSLT SU JBI descriptor with their bound XSL stylesheets to activate several services in a unique XSLT Service Unit.

Here is a sample of a SU JBI descriptor activating a new Endpoint named `TransformationEndpoint` bound to a `transform.xml` stylesheet.

```
<?xml version="1.0" encoding="UTF-8"?>
<jbi:jbi version="1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:jbi="http://java.sun.com/xml/ns/jbi"
  xmlns:petalsCDK="http://petals.ow2.org/components/extensions/version-4.0"
  xmlns:xslt="http://petals.ow2.org/components/xslt/version-2.2"
  xmlns:serviceNs="http://petals.ow2.org/simpletransformation">

  <jbi:services binding-component="false">

  <jbi:provides
    interface-name="serviceNs:TransformationInterface"
    service-name="serviceNs:TransformationService"
    endpoint-name="TransformationEndpoint">
```



```

<!-- WSDL file -->
<petalsCDK:wSDL xsi:nil="true" />

<!-- XSLT specific fields -->
<xslt:stylesheet>transform.xsl</xslt:stylesheet>
</jbi:provides>
</jbi:services>
</jbi:jbi>

```

Table 2.1. Configuration of a Service Unit to provide a service (JBI)

Parameter	Description	Default	Required
provides	Describe the JBI service that will be exposed into the JBI bus. <code>Interface</code> (qname), <code>service</code> (qname) and <code>endpoint</code> (string) attributes are required.	-	Yes

Table 2.2. Configuration of a Service Unit to provide a service (CDK)

Parameter	Description	Default	Required
wSDL	Path to the WSDL document describing services and operations exposed by the provided JBI endpoints defined in the SU. The value of this parameter is : <ul style="list-style-type: none"> • an URL • a file relative to the root of the SU package If not specified, a basic WSDL description is automatically provided by the CDK.	-	No
timeout	Timeout in milliseconds of a synchronous send. this parameter can be used in conjunction with the <code>sendSync(Exchange exchange)</code> method of the Listeners. Set 0 for an infinite timeout.	-	No
org.ow2.petals.messaging.provider.checkPetalsContainerDocument	Check PetALS container document for further details. This property activates the bypass of acknowledgment messages destined to this SU.	-	No

Table 2.3. Configuration of a Service Unit to provide a service (XSLT)

Parameter	Description	Default	Required
stylesheet	Location of the XSL stylesheet. This path must be a relative path from the root of the SU package.	-	Yes
output-attachment-name	If defined, the result of the transformation is set as an attachment, with the specified name.	-	No

2.1.2. Service Unit content

The Service Unit has to contain the following elements, packaged in an archive:

- The META-INF/jbi.xml descriptor file, has described above
- An optional wSDL file describing the related service
- The XSL stylesheet in the root of the structure
- A Jar containing the custom functions referenced in the XSL stylesheet, if any

```
service-unit.zip
```

```
+ META-INF
  - jbi.xml (as defined above)
  - service.wsdl (optional)
- myXSLfile.xsl (required)
- customFunctions.jar(optional)
```

2.1.3. Provider restrictions

The XSLT component supports only the `InOut` message exchange pattern.

The XSLT component doesn't support the synchronous sendings.

2.1.4. Provider Usage

When a request arrives, the content of the incoming normalized message (the XML source) is processed against the configured XSL style sheet (`stylesheet`). The result file is returned in the outgoing normalized message content or in an attachment with the configured name (`output-attachment-name`).

If the result is put in an attachment, the content of the outgoing normalized message takes the form:

```
<attached-files>
  <file-name>myOutputAttachmentName</file-name>
</attached-files>
```