PEtALS-SE-XSLT

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

PEtALS Team
Marie Sauvage <>

- June 2007 -

Table of Contents

List of Figures

List of Tables

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 an XSLT Stylesheet defined in the JBI description of a Service Unit.

It is based on the PEtALS CDK 3.0.

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

If you want to have more details about XSLT specification, you can consult it at this url : http://www.w3.org/TR/xslt

Chapter 1. Component Configuration

The following attributes can be set during the installation phase to configure the component, using the params element of the `jbi-install-component` ANT task:

no specific configuration for this component

Table 1.1. Advanced configuration of the component

Parameter	Description	Default	Required
pool-size	Number of threads listening to messages coming from the JBI container (JBIListeners). Int number >= 1	0	No
ignored-status	Status of messages exchanges that component must ignore. Accepted values : **DONE_AND_ERROR_IGNORED**, **DONE_IGNORED**, **ERROR_IGNORED** or **NOTHING_IGNORED**	DONE_AND_ERROR_IGNORED	No
jbi-listener-class-name	Fully qualified name of the class extending **AbstractJBIListener**		Yes
external-listener-class-name	Fully qualified name of the class extending **AbstractExternalListener**		No
properties-file	Name of the file containing values of keys used as reference by other parameters. To be able to configure a service-unit, you will use a key that has its value hosted by the component (ie. CDK documentation). The value of this parameter is : • whether an URL, • or a file relative to the directory defined by the environment variable PETALS_HOME.		No

Chapter 2. Service Configuration

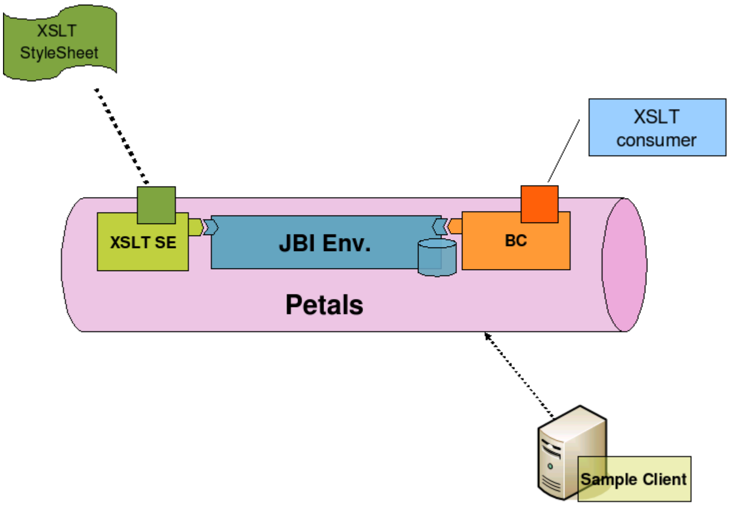
2.1. Transform XML messages

PROVIDE SERVICE : Expose an external service in the JBI environment

2.1.1. Service Unit descriptor

For each xsl style sheet one endpoint is activated. Then, a client can use this service sending a request to this endpoint.

Figure 2.1. The XSLT Service Engine



To activate a new endpoint, you must deploy a service unit that contains an endpoint definition in a provides node and an xsl stylesheet. Here is a sample of a su descriptor that activates a new Endpoint (XsltEndpoint) linked to a test.xsl stylesheet.

```
<?xml version="1.0" encoding="UTF-8"?>
<jbi:jbi xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:petals="http://petals.ow2.org/extensions"
 xmlns:tns="http://petals.ow2.org/"
 xmlns:jbi="http://java.sun.com/xml/ns/jbi" version="1.0">
 <jbi:services binding-component="false">
  <jbi:provides interface-name="tns:BottleInformationAsHTMLInterface"
   service-name="tns:BottleInformationAsHTMLService"
 endpoint-name="BottleInformationAsHTMLEndpoint">

   <!-- Specific parameters for this service unit -->
   <petals:params>
    <petals:param name="xsl.output-attachment-name">output.html</petals:param>
    <petals:param name="xsl.file">test.xsl</petals:param>
   </petals:params>
  </jbi:provides>
 </jbi:services>
</jbi:jbi>
```

Table 2.1. Service Unit attributes to provide services

Attribute	Description	Default Value	Required
provides	Name of the JBI service that will be activated to expose the XSLT Stylesheet endpoint into the JBI environment. interface (qname), service (qname) and endpoint (string) name are required.		Yes
xsl.file	Location of the xsl stylesheet. This path can be a relative path, starting at the SU root or a classpath resource from the SU root package (the xsl stylesheet can be embedded in a JAR file inside the SU).		Yes
xsl.output-attachment-name	If defined, the result of the transformation is set as an attachment, with the specified name.		No

Table 2.2. Advanced configuration of Service Unit (provides elements)

Parameter	Description	Default	Required
wsdl	Path to the wsdl file describing services and operations offered by the provided JBI endpoints defined in the SU. The path can be a url "http" or "file" or relative to the root directory of the SU archive. Ex : "file:///user/ofabre/test.wsdl" or "/WSDL/test.wsdl" If no wsdl path is specified, a simplified description will automaticaly be written by the CDK.		No
org.ow2.petals.messaging.provider.noack	All JBI exchanges end by a message containing a `DONE` or `ERROR` status. The consumer must accept those messages, otherwise they are accumulated in the NMR. With this parameter , they can be ignored to reduce the PEtALS bus traffic. Possible values are `true` or `false`. Setting a true value makes the PEtALS container ignoring acknowlegment messages addressed to the deployed SU. **This feature is unactivated when using synhronous sends.**	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

```
service-unit.zip
  + META-INF
       - jbi.xml (as defined above)
       - service.wsdl (optional)
  - test.xsl (required)
```

2.1.3. Provider restrictions

The XSLT component supports only the InOut message exchange pattern.

The XSLT componet doesn't support the synchronous sendings.

2.1.4. Provider Usage

When the activated endpoint is reached, the content of the incoming normalized message of the message exchange (the XML source) is processed against the configured xsl style sheet (`xsl.file`). The result file is returned in the outgoing normalized message content or in a attachment with the configured name (`xsl.output-attachment-name`).

If the result is put in an attachment, the content of the outgoing normalized message is like following:

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