PEtALS-SE-VALIDATION

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

PEtALS Team

Mathias BELDAME <mathias.beldame@ebmwebsourcing.com>

- Mars 2009 -

Table of Contents

List of Figures

List of Tables

PEtALS-SE-VALIDATION

This component allows to process xml validation based on xsd schema. It creates an XML output from a given XML source content and a XSD schema defined in the JBI description of a Service Unit depending on the fact the xml input validates or not against the given schema.

It is based on the PEtALS CDK.

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

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 `AbstractComponent` 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 recommanded 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 `${myPropertyName}`. At runtime, the expression is replaced by the value of the property. The value of this parameter is : • 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 in an 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 : `DONE_AND_ERROR_IGNORED`, `DONE_IGNORED`, `ERROR_IGNORED` or `NOTHING_IGNORED`	`DONE_AND_ERROR_IGNORED`	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 paramater 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.
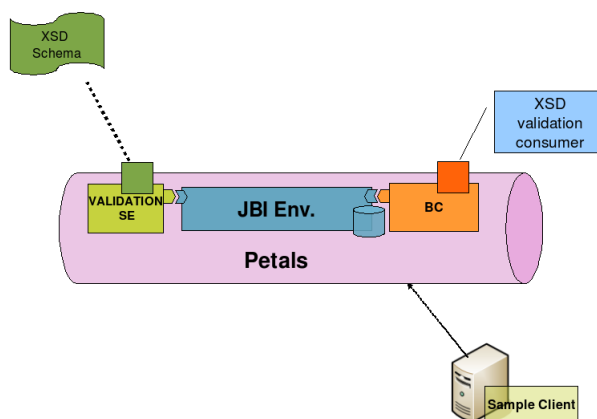
Chapter 2. Service Configuration

2.1. Valid XML messages

PROVIDE SERVICE : Expose an external service in the JBI environment

2.1.1. Service Unit descriptor

The validation component proposes services to validate xml inputs against a pre defined xsd schema. The validation occurs only on the source of the incoming message. The resulted message is returned as a source.

Figure 2.1. The Validation Service Engine

To expose your validation service into the JBI bus, you must deploy a VALIDATION Service Unit with a JBI descriptor containing a `provides` section and your XSD schema. An new endpoint would be activated in the bus at the deployment of the Service Unit.

Here is a sample of a SU JBI descriptor activating a new Endpoint named `ValidationEndpoint` bound to a schema.xsd schema.

```
<?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:validation="http://petals.ow2.org/components/validation/version-1.0"
 xmlns:serviceNs="http://petals.ow2.org/simpletransformation">

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

  <jbi:provides
   interface-name="serviceNs:ValidationInterface"
   service-name="serviceNs:ValidationService"
   endpoint-name="ValidationEndpoint">

   <!-- WSDL file -->
   <petalsCDK:wsdl>your optional description wsdl file.wsdl</petalsCDK:wsdl>

   <!-- Validation specific fields -->
   <validation:schema>schema.xsd</validation:schema>
  </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. `Interface` (qname), `service` (qname) and `endpoint` (string) attributes are required.	-	Yes

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

Parameter	Description	Default	Required
wsdl-imports-download	If false, the external imports declared in the service WSDL won't be downloaded, so they won't be replaced by their content.	True	No
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 : • an URL • a file relative to the root of the SU package If not specified, a basic WSDL description is automaticaly provided by the CDK.	-	No
timeout	Timeout in milliseconds of a synchronous send. this parameter can be used in conjunction with the `sendSync(Exchange exchange)` method of the Listeners. Set 0 for an infinite timeout.	-	No
org.ow2.petals.messaging.provider.acl	Check PEtALS container document for further details. This propety activates the bypass of acknowledgment messages destinated to this SU.	-	No

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

Parameter	Description	Default	Required
schema	Location of the XSD schema. This path must be a relative path from the root of the SU package.	-	Yes

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 XSD schema in the root of the structure, and the imported XSD files if needed.

```
service-unit.zip
  + META-INF
      - jbi.xml (as defined above)
      - file.wsdl (your optional description wsdl file.wsdl)
  - myfile.xsd (required)
  - myfile2.xsd (required if myfile2.xsd is imported in myfile.xsd)
```

2.1.3. Provider restrictions

The Validation component supports only the InOut message exchange pattern.

2.1.4. Provider Usage

When a request arrives, the content of the incoming normalized message (the XML source) is validated against the defined XSD schema).

Depending on the operation used, two behavior are possibles:

• filter mode: it is the default mode. It sends back the incoming normalized message if it validates, and sends back a fault in the other case.

• validate mode: only used when the operation is "validate". It sends back the following message when the incoming normalized message validates:

```
<tns:validateResponse>
    <tns:valid>
        true
    </tns:valid>
</tns:validateResponse>
```

In the other case, the following message is the service response (with $reason replaced by the reason why the xml incoming message does not validate):

```
<tns:validateResponse>
    <tns:valid>
        false
    </tns:valid>
    <tns:comment>
        $reason
    </tns:comment>
</tns:validateResponse>
```