PEtALS-SE-Orchestra

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

PEtALS Team
Nicolas Salatge <>

- June 2007 -

Table of Contents

List of Tables

PETALS-SE-ORCHESTRA

The Orchestra Service Engine allows to deploy BPEL 2.0 processes as Services. It can be usefull to develop business processes or workflows.

A BPEL service is deployed as a ServiceUnit on the petals-engine-orchestra component.

The ORCHESTRA serviceEndpoint is activated in the JBI environment with the information contained in the ServiceUnit descriptor file.

The ORCHESTRA service can interact with the JBI environment as a JBI component.

If you want to have more details about BPEL 2.0 specification, you can consult it at this url : http://www.oasis-open.org/committees/download.php/10347/wsbpel-specification-draft-120204.htm

Note

Now, in the PEtALS-SE-ORCHESTRA component, the Message Exchange are able to transfer the security context (login, password...).

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 configuration for this component

Table 1.1. Component installation configuration attributes

Attribute	Description	Default	Required

Table 1.2. 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

Table 1.3. Interceptors configuration in the component

Parameter	Description	Default	Required
class	Name of the interceptor class. This class must extend the abstract class org.ow2.petals.component.common.interceptor.Interceptor. This class have to be present in the classloader, in the component jar, the CDK jar or in a shared library jar.		Yes
name	Name of the interceptor. This name will be used for additional configuration in the SU.	class name	No
active	Interceptor is activated for all SUs.	true	No

Chapter 2. Service Configuration

2.1. Expose a BPEL process as service

PROVIDE SERVICE : Expose a BPEL process as a Service

2.1.1. Service Unit descriptor

The BPEL process and other required WSDL Document have to be available as XML files at the top directory of the structure of the service unit.

Take a look at Service Assembly sa-orchestra-provides-XXX.zip. It contains a Factorial BPEL Process that realize the factorial of a input integer parameter calling his own jbi endpoint several times, if the MEP allows a response.

The Service Unit descriptor file (`jbi.xml`) looks like this :

```
<jbi:jbi xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns:petals="http://petals.ow2.org/extensions"
         xmlns:wsdl="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="wsdl:factorialPT"
                  service-name="wsdl:factorialService"
                  endpoint-name="factorialEndpoint">
         <petals:wsdl>factorial.wsdl</petals:wsdl>
         <petals:params>
              <petals:param name="bpel">factorial.bpel</petals:param>
         </petals:params>
    </jbi:provides>
  </jbi:services>
</jbi:jbi>
```

Orchestra attributes :

Table 2.1. Service Unit attributes to provide services

Attribute	Description	Default	Required
provides	Name of the JBI service that will be activated to expose the JMS Destination into the JBI environment. interface (qname), service (qname) and endpoint (string) name are required.		
bpel	the name of the BPEL file to use.		Yes

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	

Table 2.3. Interceptors configuration in the Service Unit

Parameter	Description	Default	Required
name	Name of the interceptor to use. That's the name defined in the component.		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,
- The BPEL process file,
- The wsdl file describing the related service and associated to the description of the BPEL process,
- The external wsdl file describing imported in the description of the BPEL process

```
service-unit.zip
    + META-INF
          - jbi.xml (as defined above)
    - process.bpel (required)
    - service.wsdl (required)
    - externalservice(s).wsdl (optional)
```

Caution

For the moment, the name of the BPEL process have to be the same that the service name defined in the WSDL document and the JBI descriptor

2.1.3. RPC / Document style

The PEtALS-SE-ORCHESTRA component manages the RPC style and the Document style. The style is defined into the WSDL document (via the <binding> tag).

Please note that in PEtALS, the reference for the style of the document (e.g. RPC or Document) is given by the binding part of the WSDL document of the partners. For example, if the partner document is defined as RPC into the Service Unit (where are all the WSDL files), the sending of the message will be done as RPC style in PEtALS.

If you have to use several services on the same interface, it is strongly recommanded to unify the style (all in RPC style, or all in Document style).

Chapter 3. Samples

Coming soon !