PEtALS-SE-SCA

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

PEtALS Team
Mohammed EL JAI - mohammed.eljai@ebmwebsourcing.com <>
Vincent ZURCZAK - vincent.zurczak@ebmwebsourcing.com <>

- February 2009 -

Table of Contents

List of Tables

PETALS-SE-SCA

The SCA Service Engine allows you to deploy an SCA application into PEtALS. This application should be designed to compose services in the PEtALS environment.

An SCA application is materialized through an SCA composite file. To make this composite runnable into PEtALS, it must be packaged in a Service Unit and deployed through a Service Assembly for the petal-se-sca component.

When an SCA service unit is deployed, a new SCA composite is instantiated. When a service of this composite is invoked, the composite interacts with the JBI environment through the SCA service engine to execute and complete the request.

There are two distinct kinds of interaction between an SCA composite and the PEtALS environment. The first one is related to the services exposed by the composite. They are visible as end-points in the bus. The second one is related to the services used by the composite, i.e. composite references. These used (or required) services target JBI end-points, and are called by the composite implementation during runtime.

This reference mechanism is the way you can compose PEtALS services using SCA. The main advantage of SCA over other approaches, like BPEL or EIP, is that you can define the composition with Java instead of XML. The SCA composite defines services, which are exposed in the bus. It also defines references, which point to services that might be called or used by the composite at runtime. These references define possible dependencies. And eventually, your composite embeds a Java implementation which allows you to manipulate the references as simple Java objects. This makes the user job easier, in particular to call a service operation or test conditions on a service call result.

For more details, you can check further in this documentation and take a look at the links below.

- SCA is a specification defined by the Open SOA consortium.
- SCA is in standardization process by the OASIS Consortium.
- OW2 FraSCAti is the SCA platform the SCA service engine is based on.
- Eclipse SCA Tools exist and are hosted by the SOA Tools Platform project. They can be used with the SCA service engine.
- PEtALS Eclipse tools complete the STP SCA tools for PEtALS specifics (PEtALS Maven plug-in support, packaging for PEtALS...).

Chapter 1. Component Configuration

no 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 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. Expose a composite service as JBI end-point

PROVIDE SERVICE : expose a composite service as JBI end-point

2.1.1. Service Unit descriptor

The Service Unit descriptor file (`jbi.xml`) contains at least one PROVIDE mode. Every SCA service is defined in its own PROVIDE block. Here is a sample jbi.xml with one PROVIDE block:

```
<?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:sca="http://petals.ow2.org/components/sca/version-1.0"
  xmlns:petalsCDK="http://petals.ow2.org/components/extensions/version-4.0"
  xmlns:generatedNs="http://sca">

  <!-- Import a Service into PEtALS or Expose a PEtALS Service => use a BC. -->
  <jbi:services binding-component="false">

    <!--
     The following 'provide' section must be repeated as many times as
     there are services in the SCA composite.
     As an example, this jbi.xml would mean the referenced composite only exposes one service.
     -->
    <jbi:provides
      interface-name="generatedNs:sca"
      service-name="generatedNs:scaService"
      endpoint-name="scaServiceEndpoint">

      <!-- CDK specific elements -->
      <petalsCDK:wsdl>compositeService.wsdl</petalsCDK:wsdl>

      <!-- Component specific elements -->
      <sca:composite-file>%theCompositeRelativeFilePath</sca:composite-file>
    </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.noack	Check PEtALS container document for further details. This propety activates the bypass of acknowledgment messages destinated to this SU.	-	No

Table 2.3. Service Unit attributes to provide services

Attribute	Description	Default	Required
composite-file	The value of this parameter is : • The relative path of the SCA composite into the service unit archive.	-	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, as described above,
- The WSDL files of the composite services. Every service of the SCA composite (i.e. promoted component services) must provide a WSDL. These WSDLs must be in the service unit archive.

  The WSDL files of the composite references are optional. If they are not provided, they will be retrieved from the bus.

  If the WSDL is available at a fixed location (e.g. an URL), it does not have to be in the archive. The location can be referenced instead.
- The SCA artifacts, that is to say, the composite file. For the moment, this component was only tested with one composite. Using composites as component implementations or using composite inclusions has not been tested yet.
- The Java implementation jars and their dependencies.

```
service-unit.zip
    + META-INF
          - jbi.xml (as defined above)
    - compositeService(s).wsdl (1..*)
    - scaComposite(s).composite (1)
    - compositeImplementation.jar (1..*)
    - compositeReference(s).wsdl (optional)
```

Caution

The WSDLs of the composite services and references must be in the document/literal style.

Caution

The SCA composites to deploy into PEtALS must only define JBI bindings on services and references.

Caution

There is currently a strong limitation about redeployment with this version of the component. If you install the same service unit twice, you may encounter a InvocationTargetException. To solve this, you should uninstall and reinstall the SCA service engine before redeploying this service unit. This constraint will be solved in the next maintenance release of the component.

Caution

To bypass a class-loading problem, the SCA component must be installed in PEtALS with an isolated class loader. Please, use a PEtALS server version higher or equal to 2.2 and set the `isolatedClassLoader` option in the server.properties file.

2.1.3. USE CASE

See here