



PEtALS-SE-QUARTZ

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

PEtALS Team
Adrien LOUIS <>
- Mars 2009 -



Table of Contents

PEtALS-SE-QUARTZ	4
1. Component Configuration	5
2. Service Configuration	7
2.1. Job scheduling	7
2.1.1. Service Unit descriptor	7
2.1.2. Service Unit content	8
2.1.3. Consumer restrictions	9
2.1.4. Consumer Usage	9
2.1.5. Cron Expression Meaning	9

List of Tables

1.1. Configuration of the component (CDK)	5
2.1. Configuration of a Service Unit to consume a service (JBI)	7
2.2. Configuration of a Service Unit to consume a service (CDK)	8
2.3. Configuration of a Service Unit to consume a service (Quartz)	8

PEtALS-SE-QUARTZ

This component schedules jobs to trigger services invocations within the JBI bus. The jobs are based on time policies.

It is relying on the *Quartz* open source project.

This component only acts as a service consumer and doesn't act as a service provider. You use it to start another treatment managed by another service.

For more details about Quartz usage, see url : <http://www.opensymphony.com/quartz>

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. Job scheduling

CONSUME SERVICE : Call a JBI service according to a time expression

2.1.1. Service Unit descriptor

The Quartz component is configured with a cron expression, a service to call and an XML message request to send to the service.

You have to deploy a Service Unit with a JBI descriptor containing a `consumes` section and the parameters.

Here is a sample of a SU JBI descriptor to call a Hello service with `<text>hello</text>` message, at 10:15am every day during the year 2005.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- JBI descriptor for PETALS' "petals-se-quartz" (QUARTZ), version 1.0 -->
<jbi:jbi version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:jbi="http://java.sun.com/xml/ns/jbi"
xmlns:quartz="http://petals.ow2.org/components/quartz/version-1.0"
xmlns:petalsCDK="http://petals.ow2.org/components/extensions/version-4.0"
xmlns:generatedNs="http://petals.ow2.org/components/hello/version-3.0">

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

<!-- Expose a PETALS Service => consumes a Service. -->
<jbi:consumes
  interface-name="generatedNs:Hello"
  service-name="generatedNs:HelloSrv"
  endpoint-name="HelloSrvEndpoint">

  <!-- CDK specific elements -->
  <petalsCDK:operation>sayHelloOperation</petalsCDK:operation>
  <petalsCDK:mep>InOnly</petalsCDK:mep>

  <!-- Component specific elements -->
  <quartz:cron-expression>0 15 10 * * ? 2005</quartz:cron-expression>
  <quartz:content><![CDATA[ <text>hello</text> ]]> </quartz:content>
</jbi:consumes>
</jbi:services>
</jbi:jbi>
```

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

Parameter	Description	Default	Required
consumes	Name of the JBI service to invoke into the JBI bus. You can define only the interface (qname) to let the NMR choose a matching service, or the pair service(qname) and endpoint (string) to consume the localized service.	-	Yes

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

Parameter	Description	Default	Required
mep	Message exchange pattern abbreviation. This parameter can be user in conjunction with the method of the CDK Listeners : <code>createMessageExchange(Extensions extensions)</code> . This method returns a CDK Exchange corresponding to the type of the specified pattern. Admitted values are : <code>InOnly</code> , <code>RobustInOnly</code> , <code>InOptionalOut</code> et <code>InOut</code>	-	No
operation	Operation to call on a service. This parameter can be used in conjunction with the sending methods of the Listeners. If no operation is specified in the Message Exchange to send, this parameter will be used.	-	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.consumeCheck	Check PETALS container document for further details. This property activates the bypass of acknowledgment messages destined to this SU.	-	No
org.ow2.petals.routing.strategy	To be used only in platform (distributed) PETALS distribution. Check PETALS platform documentation for further details. Override the default routing strategy for Message Exchanges sent by this SU	-	No
org.ow2.petals.transport.compression	To be used only in platform (distributed) PETALS distribution. Check PETALS platform documentation for further details. This property activates the compression of the messages payload when set to true.	-	No
org.ow2.petals.transport.quality	To be used only in platform (distributed) PETALS distribution. Check PETALS platform documentation for further details. This property overrides the default policy of the Quality of Service supported by PETALS Transporter for Message Exchange sent by this SU.	-	No

Table 2.3. Configuration of a Service Unit to consume a service (Quartz)

Parameter	Description	Default	Required
cron-expression	a cron expression, for ex <code>0 30 10-13 ? * WED,FRI</code> : trigger that fires at 10:30, 11:30, 12:30, and 13:30, on every Wednesday and Friday	-	Yes
content	the XML message to send to the scheduled service. Must be CDATA format : <code><![CDATA[<text>hello</text>]]></code>	-	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

```
service-unit.zip
+ META-INF
- jbi.xml (as defined above)
```


2.1.3. Consumer restrictions

The Quartz component call scheduled service with In-Only message exchange pattern.

2.1.4. Consumer Usage

Each time that the cron expression is triggered, the Quartz component call the configured service, and send to it the defined XML message content.

2.1.5. Cron Expression Meaning

- 0 0 12 * * ? Fire at 12pm (noon) every day
- 0 15 10 ? * * Fire at 10:15am every day
- 0 15 10 * * ? Fire at 10:15am every day
- 0 15 10 * * ? * Fire at 10:15am every day
- 0 15 10 * * ? 2005 Fire at 10:15am every day during the year 2005
- 0 * 14 * * ? Fire every minute starting at 2pm and ending at 2:59pm, every day
- 0 0/5 14 * * ? Fire every 5 minutes starting at 2pm and ending at 2:55pm, every day
- 0 0/5 14,18 * * ? Fire every 5 minutes starting at 2pm and ending at 2:55pm, AND fire every 5 minutes starting at 6pm and ending at 6:55pm, every day
- 0 0-5 14 * * ? Fire every minute starting at 2pm and ending at 2:05pm, every day
- 0 10,44 14 ? 3 WED Fire at 2:10pm and at 2:44pm every Wednesday in the month of March.
- 0 15 10 ? * MON-FRI Fire at 10:15am every Monday, Tuesday, Wednesday, Thursday and Friday
- 0 15 10 15 * ? Fire at 10:15am on the 15th day of every month
- 0 15 10 L * ? Fire at 10:15am on the last day of every month
- 0 15 10 ? * 6L Fire at 10:15am on the last Friday of every month
- 0 15 10 ? * 6L Fire at 10:15am on the last Friday of every month
- 0 15 10 ? * 6L 2002-2005 Fire at 10:15am on every last friday of every month during the years 2002, 2003, 2004 and 2005
- 0 15 10 ? * 6#3 Fire at 10:15am on the third Friday of every month
- 0 0 12 1/5 * ? Fire at 12pm (noon) every 5 days every month, starting on the first day of the month.