PEtALS-BC-MAIL Component User's Guide

This document explain how to install and configure the petals-bc-mail JBI component.

PEtALS Team
Adrien LOUIS <adrien.louis@ebmwebsourcing.com>
Marie Sauvage <marie.sauvage@ebmwebsourcing.com>
Nicolas Salatge <nicolas.salatge@ebmwebsourcing.com>

- October 2008 -

Table of Contents

List of Figures

List of Tables

PEtALS-BC-MAIL

The Petals Mail binding component is a bidirectional binding component, it allows to :

- retrieve mails from an email account and send them to a JBI Service

- send JBI messages to an email account

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. 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	empty value	Yes	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

Table 1.3. 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 : `createMessageExchange(Extensions extensions)`. This method returns a CDK Exchange corresponding to the type of the specified pattern. Admitted values are : `InOnly`, `RobustInOnly`, `InOptionalOut` et `InOut`	-	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 `sendSync(Exchange exchange)` method of the Listeners. Set 0 for an infinite timeout.	-	No
org.ow2.petals.messaging.consumer.ack	Check PEtALS container document for further details. This propety activates the bypass of acknowledgment messages destinated 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.compress	**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.qos	**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

Chapter 2. Service Configuration

2.1. Send mails

PROVIDE SERVICE : Import, into JBI environment, an email account as a target service, or use a generic SendMail service

Petals Mail binding component allows JBI consumers to send mails to an email account. A JBI endpoint is registered into the JBI environment, and is linked to an smtp server, with an email address defined. When MailBC receives a message exchange from Petals platform, the content of the message is sent to the defined email address.

The component can also provide a generic SendMail service. This service allows the consumer to send a specific XML message to the component, which defines all the information needed to send an email. To use this generic service, the consumer has to call explicitly the `{service namespace}send` operation.

The message send by the consumer has to respect the following definition :

```xml
<ns0:mail xmlns:ns0="http://petals.ow2.org/components/mail/version-3.0">
  <ns0:from>from@from.com</ns0:from>
  <ns0:reply>reply@reply.com</ns0:reply>
  <ns0:to>to@to.com</ns0:to>
  <ns0:subject>subject</ns0:subject>
  <ns0:body>Hello, here is an email</ns0:body>
</ns0:mail>
```

Figure 2.1. Sending mails
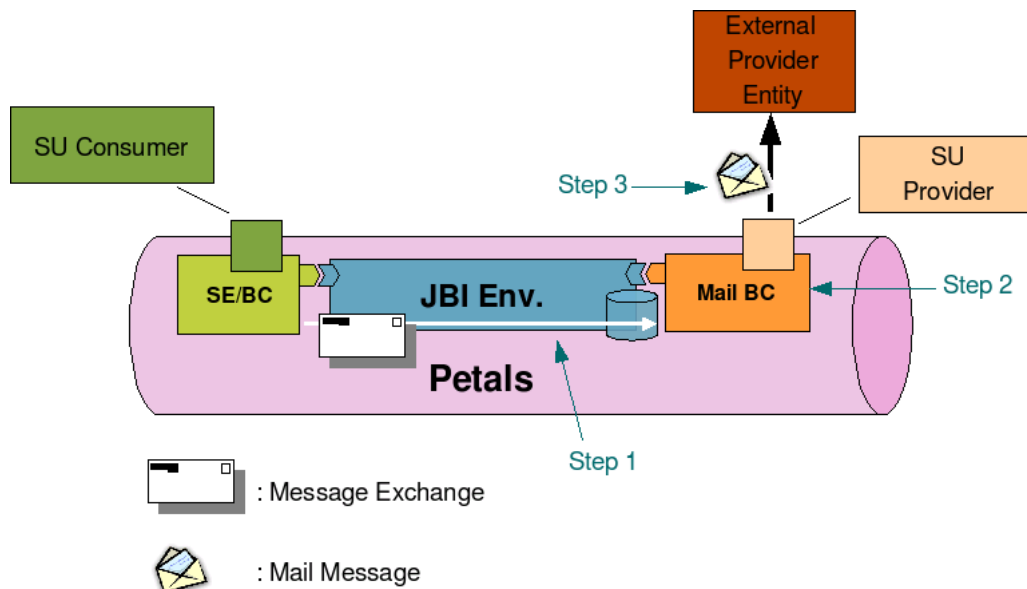
- Step 1 : A JBI Consumer sends a Message Exchange to the Mail Binding Component.

- Step 2 : Mail Binding Component processes the Message Exchange : transforms it into a mail message and retrieve targeted External Provider Service (email address) linked to the endpoint set in the Message Exchange.

- Step 3 : Mail Binding Component sends this new mail to the targeted External Provider Service (Business Service, simple email account...).

2.1.1. Service Unit descriptor

Petals Mail binding component can be configured by deploying a new service unit to it. The jbi descriptor (`jbi.xml` file) of this service unit must contain a provides node describing the link between an internal jbi endpoint and an external email address. Here is an exemple of jbi descriptor activating a new "provided service" :

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

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

  <!-- Import a Service into PEtALS => provides a Service. -->
  <jbi:provides
    interface-name="generatedNs:SendMail"
    service-name="generatedNs:SendMailService"
    endpoint-name="SendMailServiceEndpoint">

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

    <!-- Component specific elements -->
    <mail:scheme>smtp</mail:scheme>
    <mail:host>smtp.host.com</mail:host>
    <mail:port>25</mail:port>
    <mail:user>user</mail:user>
    <mail:password>password</mail:password>
    <mail:from>from email address</mail:from>
    <mail:reply>reply email address</mail:reply>
    <mail:to>recipient address</mail:to>
    <mail:subject>mail subject</mail:subject>
  </jbi:provides>
</jbi:services>
</jbi:jbi>
```

Mail communication attributes :

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 Mail Destination into the JBI environment. interface (qname), service (qname) and endpoint (string) name are required.		Yes
scheme	the connection protocol (smtp)		Yes
username	the username used for authentication		No
password	the password used for authentication.		No
host	the host address used for connection		Yes
port	the port used for connection		Yes
to	email address of the recipient **DYNAMIC ADDRESS** : You can specify the recipient address dynamicaly by setting a {http://www.w3.org/2005/08/addressing}To property in the incoming message exchange, or by using the SendMail service. **WARNING** : the recipient address must be specified at least once in the message exchange or in the SU descriptor		Yes (except when using SendMailSer
from	email address of the sender **DYNAMIC ADDRESS** : You can specify the sender email address dynamicaly by setting a {http://www.w3.org/2005/08/addressing}From property in the incoming message exchange, or by using the SendMail service. **WARNING** : the From address must be specified at least once in the message exchange or in the SU descriptor		Yes (except when using SendMailSer
reply	email address for the reply **DYNAMIC ADDRESS** : You can specify the sender email address dynamicaly by setting a {http://www.w3.org/2005/08/addressing}Reply property in the incoming message exchange, or by using the SendMail service. **WARNING** : the Reply address must be specified at least once in the message exchange or in the SU descriptor		Yes (except when using SendMailSer
subject	the subject of the mail **DYNAMIC SUBJECT** : You can specify the subject dynamicaly by setting a {http://www.w3.org/2005/08/addressing}Action property in the incoming message exchange, or by using the SendMail service.	petals-bc-mail	No

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.providermode	Check PEtALS container document for further details. This propety activates the bypass of acknowledgment messages destinated to this SU.	-	No

Table 2.3. Interceptors configuration for component (CDK)

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 component classloader, in one of the component jars or in a shared library jar.	-	Yes
name	Name of the interceptor. This name will be used for activation or additional configuration in the SU.	-	Yes
active	Interceptor is activated for all deployed SUs.	-	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

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

2.1.3. Usage

Once a *provides* node is configured, you can start to send email via the mail binding component. You just have to send message exchange to endpoints activated by service unit deployments (containing jbi.xml with provides node).
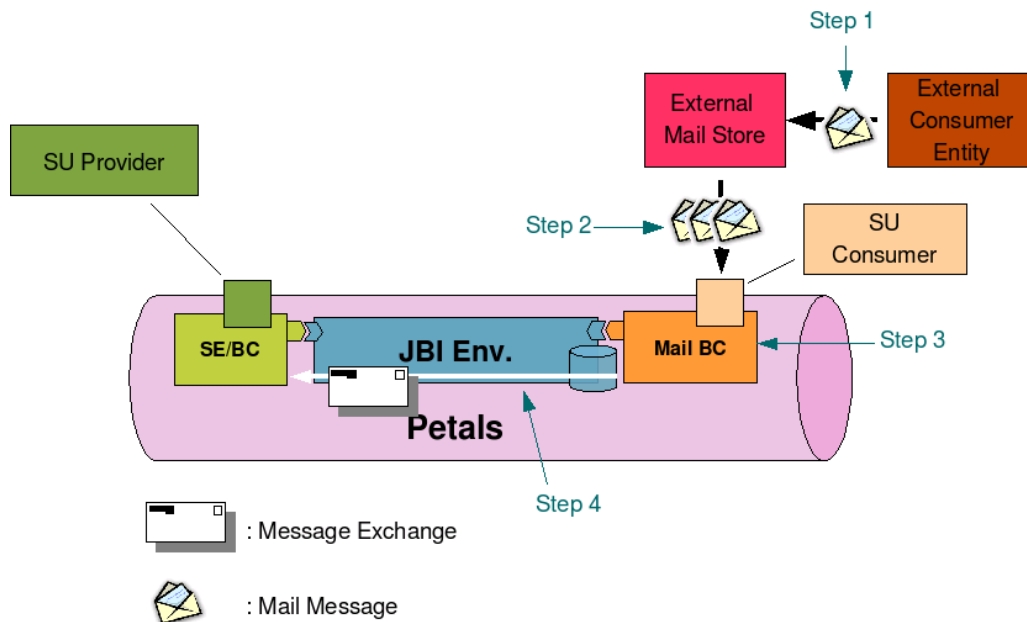
⚠ **Caution**

InOnly or RobustInOnly message exchange patterns are allowed.

2.2. Receive mails

CONSUME SERVICE : Expose an internal JBI service outside the JBI environment that can be accessed by sending mails to an email account

Figure 2.2. Receiving mails

Petals Mail binding component (MailBC) allows to receive mails from external consumer and to bind them to message exchanges intinded to internal jbi components. To receive new mails, MailBC can be linked to specific mail stores. It will check these stores periodicaly to retrieve new mails. If it finds a new mail in a store, it will process it (map this mail to a message exchange) and send it to the targeted jbi endpoint. Then the mail is removed from the store. So, all mails (read or unread) in a store are considered as new mail.

- Step 1 : An External Consumer Entity (Business Service or simple mail client) sends an email to the registered Mail Store (a classical email account).

- Step 2 : Mail Binding Component periodicaly checks for new mails and imports them.

- Step 3 and 4 : Mail Binding Component processes this new mails : transforms them into Message Exchanges, sends them to targeted jbi components (step 4) and finally delete them from the mail Store.

2.2.1. Service Unit descriptor

Petals Mail binding component can be configured by deploying a new service unit to it. The jbi descriptor (jbi.xml file) of this service unit must contains a consumes node describing the link between an external mail store and an internal jbi endpoint. Here is an exemple of jbi descriptor activating a new "consumed service" :

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

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

  <!-- Expose a PEtALS Service => consumes a Service. -->
  <jbi:consumes
    interface-name="generatedNs:Interface"
    service-name="generatedNs:Service"
    endpoint-name="Endpoint">

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

    <!-- Component specific elements -->
    <mail:scheme>pop3</mail:scheme>
    <mail:host>pop.host.com</mail:host>
    <mail:port>110</mail:port>
    <mail:user>user</mail:user>
    <mail:password>password</mail:password>
    <mail:folder>INBOX</mail:folder>
    <mail:period>60000</mail:period>
  </jbi:consumes>
</jbi:services>
</jbi:jbi>
```

Mail communication attributes :

Table 2.4. service-unit attributes to consume services

Attribute	Description	Default Value	Required
consumes	Name of the JBI service that will be called into the JBI environment. When a mail message is received. Only the interface (qname) name can be provided (the container will choose a ServiceEndpoint for this interface), or you can only set service (qname) and endpoint (string) names, without the interface name.		Yes
scheme	the connection protocol (imap or pop3)		Yes
username	the username used for authentication		No
password	the password used for authentication. Can be null or empty		No
host	the host used for connection		Yes
port	the port used for connection	imap : 143 pop3 : 110	No
folder	the folder to check for new mails	INBOX	No
period	the checking period time	60 000 ms	No
expunge	Expunge deleted messages (read messages are marked as DELETED, default is TRUE)	true	No

Note

To be able to download WSDL imports in a long futur, these imports are cached by the CDK when installing services. This feature can be disabled with disableWsdlImportsDownload.

Table 2.5. Interceptors configuration for component (CDK)

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 component classloader, in one of the component jars or in a shared library jar.	-	Yes
name	Name of the interceptor. This name will be used for activation or additional configuration in the SU.	-	Yes
active	Interceptor is activated for all deployed SUs.	-	Yes

2.2.2. 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)
```

2.2.3. Usage

When a new email is in the INBOX folder of the configured email account, the content of the mail is forwarded to the JBI Service defined in the Consumes section of the Service Unit

⚠️ **Caution**

The component sends exchange with the InOnly pattern only.

Chapter 3. Samples

Coming soon !