PEtALS-BC-FTP

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

PEtALS Team
Adrien Louis <adrien.louis@ebmwebsourcing.com>
Marie Sauvage <marie.sauvage@ebmwebsourcing.com>

- November 2008 -

Table of Contents

PEtALS-BC-FTP 5
1. Component Configuration 6
2. Service Configuration 8
2.1. Put XML content or send FTP commands to the FTP server 8
2.1.1. Service Unit descriptor 8
2.1.2. Service Unit content 9
2.1.3. FTP Service 10
3. Samples 13

List of Figures

1. ftp component mechanism 5

List of Tables

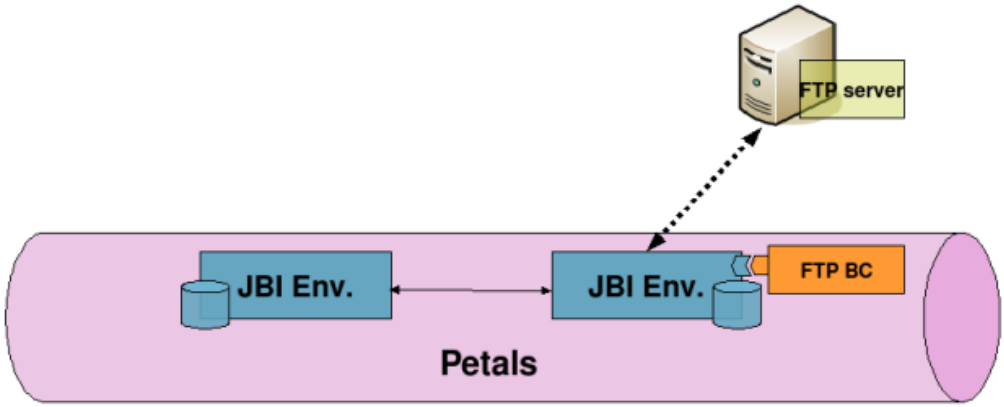
1.1. Component installation configuration attributes 6
1.2. Configuration of the component (CDK) 6
1.3. Configuration of a Service Unit to consume a service (CDK) 7
2.1. Service Unit attributes to provide services 9
2.2. Configuration of a Service Unit to provide a service (CDK) 9

PEtALS-BC-FTP

The `petals-bc-ftp` component allows to put, get or list files on an FTP server.

FTP protocol (RFC959) is described at : *http://www.w3.org/Protocols/rfc959/*

Figure 1. ftp component mechanism



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.ackD	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. Put XML content or send FTP commands to the FTP server

PROVIDE SERVICE : Expose an external service in the JBI environment to put XML messages or send FTP commands on an FTP server

Petals FTP binding component allows JBI consumers to send XML messages to an FTP server. A JBI endpoint is registered into the JBI environment, and is linked to an FTP server, with a configured user/password. When the FTP component receives a message exchange from Petals platform, the XML content of the message is put on the FTP server.

The component can also provide a generic **Ftp service** . This service allows the consumer to call FTP commands. This service can either connects to an FTP server configured in the ServiceUnit, or retrieves the FTP server information in the XML message request.

To use this generic service, the consumer has to call explicitely the **`{service namespace}get/mget/put/mput/dir`** operations. Otherwise, the standard PUT operation is called.

2.1.1. Service Unit descriptor

The service unit is configurable via its extensions in the jbi.xml file :

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- JBI descriptor for PEtALS' "petals-bc-ftp" (FTP), 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:ftp="http://petals.ow2.org/components/ftp/version-3.0"
xmlns:petalsCDK="http://petals.ow2.org/components/extensions/version-4.0"
xmlns:interfaceNs="http://petals.ow2.org/components/ftp/version-3.0"
xmlns:serviceNs="http://myservice">

<!-- 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="interfaceNs:Ftp"
  service-name="serviceNs:FtpSrv"
  endpoint-name="FtpSrvEndpoint">

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

    <!-- Component specific elements -->
    <ftp:server>server</ftp:server>
    <ftp:port>21</ftp:port>
    <ftp:user>user</ftp:user>
    <ftp:password>password</ftp:password>
    <ftp:folder>folder</ftp:folder>
    <ftp:filename>file.xml</ftp:filename>
  </jbi:provides>
</jbi:services>
</jbi:jbi>
```

Table 2.1. Service Unit attributes to provide services

Attribute	Description	Default Value	Required
server	IP or DNS name of the server		Yes (except when using FtpService)
port	the port number of the ftp server	21	No
user	the user login name used to connect to the server		Yes (except when using FtpService)
password	the user password		Yes (except when using FtpService)
folder	the folder on the FTP server		No
filename	file name for the standard PUT operation (write the XML message content on the FTP server)	content.xml	No
connection-mode	In Active mode, the connection port is set by the client . In Passive mode, it is set by the server. (default is Active)	active	No
transfer-type	transfer mode used for put or get files ascii/binary/auto (default is AUTO, but not safe)	auto	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.provider.nack	Check PEtALS container document for further details. This propety activates the bypass of acknowledgment messages destinated to this SU.	-	No

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. FTP Service

The `petals-bc-ftp` component provides 5 operations, when using the Ftp service:

- **dir** : return the files list on the FTP server
- **put** : copy incoming XML message on the FTP server
- **mput** : copy incoming message attachments on the FTP server
- **get** : get one specified file from the FTP server and return it as an XML message
- **mget** : get files from the FTP server and return them as attachments

When using the FTP service, you can optionnaly define the FTP connection information in your XML request. If you don't, the service unit parameters are used.

To configure the FTP connection in your XML message request, you have to provide a `connection` element.

```
...
         <ver:connection>
            <ver:server>server</ver:server>
            <ver:port>port</ver:port>
            <ver:user>user</ver:user>
            <ver:password>password</ver:password>
            <ver:folder>folder (optional)</ver:folder>
            <ver:connection-mode>active|passive (optional, default is active)</ver:connection-mode>
            <ver:transfer-type>ascii|binary|auto (optional, default is auto)</ver:connection-mode>
         </ver:connection>
```

2.1.3.1. DIR operation

When the `dir` operation is set on the incoming IN message, the component returns the file names listed from the FTP server.

The IN message looks like :

```
      <ver:dir>
         <!--Optional:-->
         <ver:connection>
           ...
         </ver:connection>
      </ver:dir>
```

The OUT message returned to the consumer is defined as follow :

```
<tns:dirResponse xmlns:tns="http://petals.ow2.org/components/ftp/version-3.0">
         <tns:filename>source.xml</tns:filename>
         <tns:filename>test.xml</tns:filename>
</tns:dirResponse>
```

The service might return a Fault when an element in the request is missing or if the FTP connection failed

The available exchange patterns are : **InOptionalOut**, **InOnly**, **RobustInOnly**.

2.1.3.2. PUT operation

When the `put` operation is set on the incoming IN message, the component write the XML message to the FTP server.

The IN message looks like :

```
       <ver:put xmlns:ver="http://petals.ow2.org/components/ftp/version-3.0">

         <ver:body>xml body</ver:body>
         <ver:filename>file name on the FTP server</ver:filename>

        <!--Optional:-->
        <ver:connection>
           ...
        </ver:connection>
     </ver:put>
```

The service does not return a message.

The service might return a Fault when an element in the request is missing or if the FTP connection failed

The available exchange patterns are : **InOptionalOut**, **InOnly**, **RobustInOnly**.

2.1.3.3. MPUT operation

When the `mput` operation is set on the incoming IN message, the component write the attachments on the FTP server.

The IN message looks like :

```
     <ver:mput xmlns:ver="http://petals.ow2.org/components/ftp/version-3.0">
          <!--Optional:-->
        <ver:connection>
          ...
        </ver:connection>
     </ver:mput>
```

The service does not return a message.

The service might return a Fault when an element in the request is missing or if the FTP connection failed

The available exchange patterns are : **InOptionalOut**, **InOnly**, **RobustInOnly**.

2.1.3.4. GET operation

When the `get` operation is set on the incoming IN message, the component retrieve ONE file from the FTP server and return it as an XML message content.

The name of the file to retrieve is set in the XML IN content of the message

The IN message looks like :

```
    <ver:get xmlns:ver="http://petals.ow2.org/components/ftp/version-3.0">

         <ver:filename>file name to retrieve</ver:filename>

        <!--Optional:-->
        <ver:connection>
          ...
        </ver:connection>
     </ver:get>
```

The OUT message returned to the consumer is the content of the XML file

The service might return a Fault when an element in the request is missing or if the FTP connection failed

The available exchange patterns are : **InOptionalOut** and **InOut**.

2.1.3.5. MGET operation

When the `mget` operation is set on the incoming IN message, the component retrieves files from the FTP server, according to the file names (or filters) set in the XML request.

There is no recursivity, sub folders are ignored.

Each file is set in the OUT message as an attachment.

The IN message looks like :

```
      <ver:mget xmlns:ver="http://petals.ow2.org/components/ftp/version-3.0">

          <!--1 or more repetitions:-->
          <ver:filename>*.xml</ver:filename>
          <ver:filename>myFile.txt</ver:filename>

         <!--Optional:-->
         <ver:connection>
           ...
         </ver:connection>
      </ver:mget>
```

The OUT message returned to the consumer contains files, as attachments, and an XML message report :

```
<tns:mgetResponse xmlns:tns="http://petals.ow2.org/components/ftp/version-3.0">
         <tns:filename>source.xml</tns:filename>
         <tns:filename>source2.xml</tns:filename>
         <tns:filename>myFile.txt</tns:filename>
</tns:mgetResponse>
```

The service might return a Fault when an element in the request is missing or if the FTP connection failed

The available exchange patterns are : **InOptionalOut**, **InOut**.

Chapter 3. Samples

Coming soon !