PEtALS-BC-FTP

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

PEtALS Team

Marie Sauvage <marie.sauvage@ebmwebsourcing.com>

- June 2007 -

Table of Contents

List of Figures

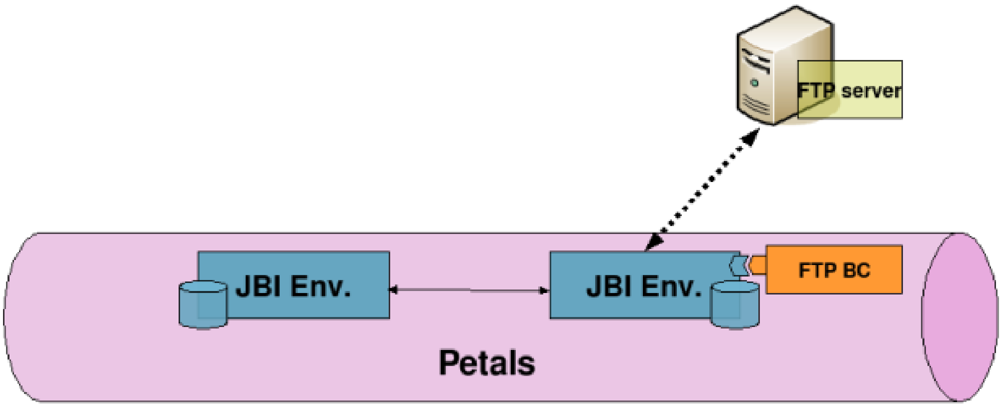
List of Tables

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. 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.objectweb.petals.component.common.interceptor.Interceptor. This class have to be present in the classloader, in component or CF or in a shared library.		Yes
name	Name of the interceptor. This name will be used for additional configuration in the SU.	class name	No
active	Interceptor is active for all SU.	true	No

Chapter 2. Service Configuration

2.1. Send commands to the FTP server

PROVIDE SERVICE : Expose an external service in the JBI environment

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:jbi xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:petals="http://petals.ow2.org/extensions"
    xmlns:jbi="http://java.sun.com/xml/ns/jbi"
    xmlns:cogitis="http://cogitis.fr/marche"
    version="1.0">
  <jbi:services binding-component="true">
   <jbi:provides interface-name="petals:interfaceftp"
     service-name="petals:serviceftp"
     endpoint-name="endpointftp">
     <petals:params>
      <petals:param name="ftp.server">myserver</petals:param>
      <petals:param name="ftp.login">login</petals:param>
      <petals:param name="ftp.password">password</petals:param>
      <petals:param name="ftp.port">21</petals:param>
      <petals:param name="ftp.folder">myfolder</petals:param>
      <petals:param name="ftp.default.operation">get</petals:param>
      <petals:param name="ftp.file-name">source.xml</petals:param>
      <petals:param name="ftp.put.exchange-data">both</petals:param>
      <petals:param name="ftp.get.exchange-data">attachments</petals:param>
    </petals:params>
  </jbi:provides>
 </jbi:services>
</jbi:jbi>
```

Table 2.1. Service Unit attributes to provide services

Attribute	**Description**	**Default Value**	**Required**
ftp.server	IP or DNS name of the server		Yes
ftp.port	the port number of the ftp server, if no port is specified, the default port will be used	21	No
ftp.login	the user login name used to connect to the server		Yes
ftp.password	the user password		Yes
ftp.folder	the folder on the FTP server	""	No
ftp.default.operation	operation to use if no one is specified in the incoming message	put	No
ftp.file-name	file name for a PUT operation writing the XML IN content on the FTP server (not for writing attachments), or for a GET operation with 'default' specified in the XML IN content.Not used in other cases		Depends
ftp.put.exchange-data	used for PUT operation. Define which elements of an incoming message have to be written on the FTP server. Values are 'source' for writing only the XML IN content, 'attachments' for writing only attachments, or 'both' for writing source and attachments	both	No
ftp.get.exchange-data	used for GET operation. Define if the get file from the FTP server has to be written in the XML OUT as a content of the message exchange or as an attachment.Not used in other cases	source	No

Table 2.2. Advanced configuration of Service Unit (provides elements)

Parameter	Description	Default	Required
wsdl	path to a wsdl file describing services and operations offered by an endpoint activated by the SU. **This extension is only usable with provides fields.** The path can be a url "http" or "file" or relative to the r oot 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 CF.		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 commands

The `petals-bc-ftp` component exposes 4 operations :

- **dir** : return the files list on the FTP server
- **put** : copy incoming messages on the FTP server
- **get** : get one specified file from the FTP server and return it
- **mget** : get all files from the FTP server and return them

2.1.3.1. DIR operation

When the `dir` operation is set on the incoming IN message, the component retrieves the filenames list from the FTP server (with the informations set in the service unit :URL, user, folder...).

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

```
<list folder="[folder set in the SU]">
    <file-name>[filename 1]</file-name>
    <file-name>[filename 2]</file-name>
    ...
</list>
```

2.1.3.2. PUT operation

When the `put` operation is set on the incoming IN message, the component write the elements contained in the message to the FTP server (with the informations set in the service unit :URL, user, folder...).

Elements to write are defined with the `put.exchange-data`, set in the service-unit :

- source : the XML content of the message is written. The name of the written file is the one set in the service-unit with the `file-name` attribute.
- attachments : all the attachments of the message are written.
- source-and-attachments : the two previous cases.

2.1.3.3. GET operation

When the `get` operation is set on the incoming IN message, the component retrieve ONE file from the FTP server (with the informations set in the service unit :URL, user, folder...).

The name of the file to retrieve is set in the XML IN content of the message, as follow:

```
<file-name>[filename]</file-name>
```

If the specified file-name value is `default`, the file name used is the `file-name` attribute set in the service-unit.

The way to store the file in the OUT message is defined with the `get.exchange-data`, set in the service-unit :

- source : the retrieved file is set in the XML OUTcontent of the message.
- attachment : the retrieved file is set as an attachment.

2.1.3.4. MGET operation

When the `mget` operation is set on the incoming IN message, the component retrieves all the files from the FTP server (with the information sets in the service unit : URL, user, folder...).

There is no recursivity, sub folders are ignored.

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

The XML OUT content references the file-name list of retrieved files, as follow :

```
<attached-files>
    <file-name>[filename 1]</file-name>
    <file-name>[filename 2]</file-name>
    ...
<attached-files>
```

2.2. Listen FTP events (not implemented)

CONSUME SERVICE : Expose an internal service outside the JBI environment

Table 2.3. Advanced configuration of Service Unit (consumes elements)

Parameter	Description	Default	Required
mep	Message exchange pattern abbreviation. This parameter can be user in conjunction with a method of the Listeners : **createMessageExchange(Extensions extensions)**. This merthod returns a MessageExchange corresponding to the type of the specified pattern. Admitted values are : **InOnly**, **RobustInOnly**, **InOptionalOut** et **InOut**		Yes
operation	Operation to call on a service. This parameter can be used in conjuntcion with the sendXXX methods of the Listeners. If no operation is specified in the MessageExchange to send, this parameter will be used.		Yes
timeout	Timeout in milliseconds in a synchroneous send. this parameter can be used in conjunction with the **sendSync(MessageExchange exchange)** method of the Listeners. With this, a synchroneous send is done with this timeout value. 0 for no timeout int number >= 0 for a timeout	0	No
org.objectweb.petals.routing.strategy	This property defines the routing strategy. Two kind of strategy can be defines: highest or random. The others parameters represents respectively the local ponderation, the ponderation of the remote active endpoint and the ponderation of the remote inactive endpoint. The 'random' strategy chooses an endpoint in function of defined ponderations. The endpoints that have the strongest ponderation can be more easely choose in comparison with the others. The 'highest' strategy chooses the first endpoint in the list that have the strongest ponderation.		No
org.objectweb.petals.transport.compress	The payload of a MessageExchange is an XML file. It can be interesting to compress it before messages are exchanged between two PEtALS nodes. Values are `true` or `false`. True activated the compression of the content of the message.		No
org.objectweb.petals.messaging.noack	All JBI exchanges ended by a message containing a `DONE` or `ERROR` status. The consumer must accept those messages, otherwise they are accumulated in the NMR. Moreover, thoses messages cause useless trafic. Values are `true` or `false`. True make `DONE` or `ERROR` messages not sent.		No
org.objectweb.petals.transport.qos	This property to set up the policy of the Quality of Service supported by Petals Transporter. Possible values are : reliable, fast. If not specified, the reliable policy is selected by default.		No

Table 2.4. 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

Chapter 3. Samples

See the following Service Assembly sample that illustrates the configuration of this component :

http://wiki.petals.objectweb.org/xwiki/bin/download/Components.Binding/FTP/FTPClientServiceSA.zip