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

PEtALS-BC-FTP ......................................................................................................................... 5
1. Component Configuration ................................................................................................. 6
2. Service Configuration ...................................................................................................... 7
   2.1. Send commands to the FTP server ............................................................................... 7
       2.1.1. Service Unit descriptor ....................................................................................... 7
       2.1.2. Service Unit content .......................................................................................... 8
       2.1.3. FTP commands .................................................................................................. 8
  2.2. Listen FTP events (not implemented) .......................................................................... 9
3. Samples ............................................................................................................................ 11
List of Figures

1. ftp component mechanism ........................................................................................................ 5
List of Tables

1.1. Component installation configuration attributes ................................................................. 6
1.2. Advanced configuration of the component .............................................................................. 6
1.3. Interceptors configuration in the component ............................................................................ 6
2.1. Service Unit attributes to provide services ............................................................................ 7
2.2. Advanced configuration of Service Unit (provides elements) .................................................. 8
2.3. Advanced configuration of Service Unit (consumes elements) ................................................ 10
2.4. Interceptors configuration in the Service Unit ........................................................................ 10
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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
</table>

### Table 1.2. Advanced configuration of the component

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool-size</td>
<td>Number of threads listening to messages coming from the JBI container (JBIListeners).</td>
<td>Int number &gt;= 1</td>
<td>No</td>
</tr>
<tr>
<td>ignored-status</td>
<td>Status of messages exchanges that component must ignore. Accepted values: DONE_AND_ERROR_IGNORED, DONE_IGNORED, ERROR_IGNORED or NOTHING_IGNORED</td>
<td>DONE_AND_ERROR_IGNORED</td>
<td>No</td>
</tr>
<tr>
<td>jbi-listener-class-name</td>
<td>Fully qualified name of the class extending AbstractJBIListener</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>external-listener-class-name</td>
<td>Fully qualified name of the class extending AbstractExternalListener</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>properties-file</td>
<td>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.</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

### Table 1.3. Interceptors configuration in the component

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>class</td>
<td>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.</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>name</td>
<td>Name of the interceptor. This name will be used for additional configuration in the SU.</td>
<td>class name</td>
<td>No</td>
</tr>
<tr>
<td>active</td>
<td>Interceptor is active for all SU.</td>
<td>true</td>
<td>No</td>
</tr>
</tbody>
</table>
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
<?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">attachments</petals:param>
         </petals:params>
      </jbi:provides>
   </jbi:services>
</jbi:jbi>
```

Table 2.1. Service Unit attributes to provide services

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

PetALS-BC-FTP

7
Table 2.2. Advanced configuration of Service Unit (provides elements)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>wsdl</td>
<td>path to a wsdl file describing services and operations offered by an endpoint activated by the SU. <strong>This extension is only usable with provides fields.</strong> The path can be a url &quot;http&quot; or &quot;file&quot; or relative to the root directory of the SU archive. Ex : &quot;file:///user/ofabre/test.wsdl&quot; or &quot;/WSDL/test.wsdl&quot; If no wsdl path is specified, a simplified description will automatically be written by the CF.</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

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 :
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 OUT content 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)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>mep</td>
<td>Message exchange pattern abbreviation. This parameter can be used in conjunction with a method of the Listeners: <code>createMessageExchange(Extensions extensions)</code>. This method returns a MessageExchange corresponding to the type of the specified pattern. Admitted values are: <code>InOnly</code>, <code>RobustInOnly</code>, <code>InOptionalOut</code> et <code>InOut</code></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>operation</td>
<td>Operation to call on a service. This parameter can be used in conjunction with the sendXXX methods of the Listeners. If no operation is specified in the MessageExchange to send, this parameter will be used.</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>timeout</td>
<td>Timeout in milliseconds in a synchronous send. This parameter can be used in conjunction with the <code>sendSync(MessageExchange exchange)</code> method of the Listeners. With this, a synchronous send is done with this timeout value. 0 for no timeout int number ≥ 0 for a timeout</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>
| org.objectweb.petals.routing.strategy | This property defines the routing strategy. Two kinds of strategy can be defined: 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 easily 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, those messages cause useless traffic. Values are `true` or `false`. True make `DONE` or `ERROR` messages not sent. | No |
| org.objectweb.petals.transport.qos | This property 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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the interceptor to use. That's the name defined in the component.</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 3. Samples

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