Java EE Client
Programmer's Guide

JOnAS Team (Philippe COQ)

- Feb 2008 -

Copyright © OW2 Consortium 2008-2009

Table of Contents

The target audience for this guide is the Client component provider, i.e. the person in charge of developing the Client components on the client side. It describes how the Client component provider should build the deployment descriptors of its Client components and how the client components should be packaged.

Chapter 1. Launching Java EE Client Applications

1.1. Launching clients

The Java EE client application can be

- a standalone client in a .jar file,

- a client bundled in an .ear file. An ear can contain many java clients.

- a class name which must be found in the CLASSPATH.

All the files required to launch the client container are in the `JONAS_ROOT/lib/client.jar` file. This jar includes a manifest file with the name of the class to launch. To launch the client container on a computer where a JONAS_TROOT is present , simply type:

`jclient -?` . This will launch the client container and display usage information about this client container.

To launch the client container on a remote computer, copy the client.jar and invoke the client container by typing `java -jar path_to_your/client.jar`

The client that must be launched by the client container is given as an argument of the client container. example: `java -jar client.jar myApplication.ear` or `java -jar client.jar myClient.jar` .

1.2. Configuring client container

1.2.1. JNDI access

Defining the JNDI access and the protocol to use is an important part of configuration. The JOnAS server, as well as the ClientContainer, uses the values specified in the `carol.properties` file. This file can be used at different levels. The `carol.properties` is searched with the following priority (high to low):

- the `carol.properties` specified by the `-carolFile` argument to the client container

- the `carol.properties` packaged into the client application (the jar client)

- if not located previously, it will use the `carol.properties` contained in the `JONAS_ROOT/lib/client.jar` .

A convenient way is to update the `carol.properties` of a `client.jar` with a customized `carol.properties` file. That is, use the `jar -uf client.jar carol.properties` command.

1.2.2. Trace configuration

The client container `client.jar` includes a `traceclient.properties` file. This is the same file as the one in `JONAS_ROOT/conf` directory. A different configuration file can be used for the traces by specifying the parameter `-traceFile` when invoking the client container. The file in the `client.jar` can be replaced with the `jar -uf client.jar traceclient.properties` command.

1.2.3. Classpath configuration

Some jars/classes can be added to the client container. For example if a class requires some extra libraries/classes, the option `-cp path/to/classes` can be used.

The classloader of the client container will use the libraries/classes provided by the `-cp` flag.

1.2.4. Specifying the client to use (EAR case)

An ear can contain many java clients, which are described in the `application.xml` file inside the `<module><java>` elements. To invoke the client container with an ear, such as `java -jar client.jar my.ear`, specify the java client to use if there are many clients. Otherwise, it will take the first client. To specify the jar client to use from an ear, use the argument `-jarClient` and supply the name of the client to use. The `earsample` example in the JOnAS examples has two java clients in its ear.

1.2.5. Specifying the directory for unpacking the ear (EAR case)

By default, the client container will use the system property `java.io.tmpdir` To use another temporary directory, specify the path by giving the argument `-tmpDir` to the client container.

1.2.6. Disable Automated WsGen

By default, the client container will apply WsGen (generation of web services artifacts) on all given archives. To disable that feature (because WsGen has already been applied on the application, or because the client contains no web services), add the `-nowsgen` option to the client container.

1.3. Examples

The Java EE 5 EAR Sample located under $JONAS_ROOT/examples/javaee5-earsample provides three application clients showing how to interact with the application in different ways, and under different security levels.

These application clients are described in details in Getting started with JOnAS 5 [1]

---

[1] getting_started_guide.html#getting_application_clients

Chapter 2. Defining the Client Deployment Descriptor

2.1. Principles

The Client component programmer is responsible for providing the java class of the Client annotated accordingly to the Java EE 5 Specification.

If the Client component programmer wants to provide a XML deployment descriptor, this one mus be compliant to the XML Schema for the application client 5 deployment descriptor http://java.sun.com/xml/ns/javaee/application-client_5.xsd

To customize the Client components, information not defined in the standard XML deployment descriptor may be needed.

The JOnAS-specific deployment descriptor's XML schema is located in `http://jonas.ow2.org/ns/jonas-client_5_0.xsd`

2.2. Examples of Client Deployment Descriptors

configuration_guide.html

• Example of a standard Client Deployment Descriptor (application-client.xml):

```xml
<?xml version="1.0" encoding="UTF-8"?>

<application-client
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
                      http://java.sun.com/xml/ns/javaee/application-client_5.xsd"
  version="5">

  <display-name>OW2 JOnAS :: EAR Sample :: JMS Application Client</display-name>

  <!-- The JMS ConnectionFactory to use -->
  <resource-ref>
    <res-ref-name>jms/QueueConnectionFactory</res-ref-name>
    <res-type>javax.jms.QueueConnectionFactory</res-type>
    <res-auth>Container</res-auth>
  </resource-ref>

  <!-- The JMS Queue where Messages will be send -->
  <resource-env-ref>
    <resource-env-ref-name>jms/SampleQueue</resource-env-ref-name>
    <resource-env-ref-type>javax.jms.Queue</resource-env-ref-type>
  </resource-env-ref>

</application-client>
```

• Example of a specific Client Deployment Descriptor (jonas-client.xml):

```xml
<jonas-client xmlns="http://www.objectweb.org/jonas/ns"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.objectweb.org/jonas/ns
        http://jonas.ow2.org/ns/jonas-client_5_0.xsd" >
  <jonas-resource>
    <res-ref-name>jms/QueueConnectionFactory</res-ref-name>
    <jndi-name>JQCF</jndi-name>
```

```
    </jonas-resource>

    <jonas-resource-env>
      <resource-env-ref-name>jms/SampleQueue</resource-env-ref-name>
      <jndi-name>SampleQueue</jndi-name>
    </jonas-resource-env>
</jonas-client>
```

For advices about xml file writing, refer to Section A.1, "xml Tips" .

Chapter 3. Client Packaging

3.1. Principles

Client components are packaged for deployment in a standard Java programming language Archive file called a jar file (Java ARchive). The document root contains a subdirectory called META-INF , which contains the following files and directories:

- application-client.xml : The standard xml deployment descriptor in the format defined in the Java EE 5 Specification. Refer to http://java.sun.com/xml/ns/javaee/application-client_5.xsd .

- jonas-client.xml : The optional JOnAS specific xml deployment descriptor in the format defined in http://jonas.ow2.org/ns/jonas-client_5_0.xsd .

The manifest of this client jar must contain the name of the class to launch (containing the main method). This is defined by the value of the Main-Class attribute of the manifest file. For a standalone client (not bundled in an Ear), all the Ejb classes (except the skeleton) on which lookups will be performed must be included.

3.2. Example

Two examples of building a java client are provided.

- The first is the `build.xml` of the `earsample` example with a java client inside the ear. Refer to the client1jar and client2jar targets.

- The second is the `build.xml` of the `jaasclient` example with a java standalone client which performs a lookup on an EJB. Refer to the clientjars target.

Appendix A. Appendix

A.1. xml Tips

Although some characters, such as ">", are legal, it is good practice to replace them with XML entity references.

The following is a list of the predefined entity references for XML:

<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark