Leading Open Source Middleware

Java EE Application Assembler's Guide

JOnAS Team (Guillaume SAUTHIER)

- Fev 2008 -

Table of Contents

The target audience for this guide is the Application Provider and Assembler, i.e. the person in charge of combining one or more components (ejb-jars and/or wars) to create a Java EE application. It describes how the Java EE components should be packaged to create a Java EE application.

Chapter 1. Defining the Ear Deployment Descriptor

1.1. Principles

The application programmer is responsible for providing the deployment descriptor associated with the developed application (Enterprise ARchive). The Application Assembler's responsibilities is to provide a XML deployment descriptor that conforms to the deployment descriptor's XML schema as defined in the Java EE specification version 5. (Refer to http://java.sun.com/xml/ns/javaee/application_5.xsd).

To deploy Java EE applications on the application server, all information is contained in one XML deployment descriptor. The file name for the application XML deployment descriptor is `application.xml` and it must be located in the top level META-INF directory.

Some Java EE application examples are provided in the JOnAS distribution:

- $JONAS_ROOT/examples/cluster-javaee5 for the Java EE 5 cluster demo
- $JONAS_ROOT/examples/javaee5-earsample for the Java EE 5 library example

The standard deployment descriptor should contain structural information that includes the following:

- EJB components,
- Web components,
- Client components,
- Alternate Deployment Descriptor for theses components,
- Security role.

There is no JOnAS-specific deployment descriptor for the Enterprise ARchive.

1.2. Sample Application Deployment Descriptor

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

<application
  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_5.xsd"
  version="5">


  <description>Java EE 5.0 EAR Sample</description>
  <display-name>Java EE 5.0 EAR Sample</display-name>

  <!-- EJB Modules -->
  <module>
    <ejb>ejb3.jar</ejb>
  </module>

  <!-- Web Modules -->
  <module>
    <web>
      <web-uri>javaee5-earsample.war</web-uri>
      <context-root>javaee5-earsample</context-root>
```

```
    </web>
  </module>

  <!-- Application Client Modules -->
  <module>
    <!-- Application Client using JMS to interact with the application -->
    <java>jms-application-client.jar</java>
  </module>
  <module>
    <!-- Not secured Application Client (only access read-only beans) -->
    <java>not-secured-application-client.jar</java>
  </module>
  <module>
    <!-- Secured Application Client (can access all bean under a SecurityContext) -->
    <java>jaas-secured-application-client.jar</java>
  </module>

</application>
```

Chapter 2. EAR Packaging

2.1. Ear Components

Java EE applications are packaged for deployment in a standard Java programming language Archive file called an ear file (Enterprise ARchive). This file can contain the following:

The web components (war)	One or more wars which contain the web components of the Java EE application. Due to the class loader hierarchy, when the wars are packaged in a Java EE application, it is not necessary to package classes of EJBs accessed by the web components in the WEB-INF/lib directory. Details about this class loader hierarchy are described in JOnAS class loader hierarchy [j2eeprogrammerguide#j2ee.pg.classloader] .
The EJB components (ejb-jar)	One or more ejb-jars, which contain the beans of the Java EE application.
The RAR components (resource adapters)	One or more rars, which contain the resource adapters of the Java EE application.
The libraries (jar)	One or more jars which contain the libraries (tag libraries and any utility libraries) used for the Java EE application.
The Java EE deployment descriptor	The standard xml deployment descriptor in the format defined in the Java EE 5 specification: http://java.sun.com/xml/ns/javaee/application_5.xsd . This deployment descriptor must be stored with the name `META-INF/application.xml` in the ear file.

2.2. Ear MANIFEST.MF

An EAR being a standard Java archive, it also has a MANIFEST file.

An interesting attribute to add to this file is the EAR's implementation version: indeed, if this information is present and the versioning [services.versioning.config.about] service is active, you can do smooth application version migration:

```
Manifest-Version: 1.0
Implementation-Version: 2.1.0
```

2.3. Example

Before building an ear file for a Java EE application, the ejb-jars and the wars that will be packaged in the Java EE application must be built and the XML deployment descriptor (application.xml) must be written.

Then, the ear file (`<java-ee-application>.ear`) can be built using the jar command:

```
cd <java-ee_application_directory>
jar cvfm <java-ee-application>.ear META-INF/MANIFEST.MF *
```

Note that in order to include your custom MANIFEST file, you need to specify its path to the `jar` command:

```
cd <java-ee_application_directory>
jar cvfm <java-ee-application>.ear META-INF/MANIFEST.MF *
```

Chapter 3. EAR Packaging using Maven

If the Java EE application's components are available as Apache Maven [http://maven.apache.org/] dependencies, you can also use the maven-ear-plugin [http://maven.apache.org/plugins/maven-ear-plugin/] for the EAR generation. In this case, the generation and inclusion of all files is done automatically by Maven.

Here's an example pom.xml file generating a EAR with the Implementation Version:

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

<project
  xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                      http://maven.apache.org/maven-v4_0_0.xsd">

  <parent>
    <groupId>org.ow2.jonas.samples</groupId>
    <artifactId>ear-sample</artifactId>
    <version>1.0.0</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>
  <artifactId>${parent.artifactId}-ear</artifactId>
  <packaging>ear</packaging>
  <name>JOnAS Java EE 5 sample</name>
  <description>This is a sample Java EE 5 application.</description>

  <dependencies>
    <!-- WARs and EJB-JARs -->
    <dependency>
      <groupId>${project.groupId}</groupId>
      <artifactId>${parent.artifactId}-ejb</artifactId>
      <version>${project.version}</version>
      <type>ejb</type>
    </dependency>
    <dependency>
      <groupId>${project.groupId}</groupId>
      <artifactId>${parent.artifactId}-war</artifactId>
      <version>${project.version}</version>
      <type>war</type>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <artifactId>maven-ear-plugin</artifactId>
        <configuration>
          <!-- Make sure it is a Java EE 5 EAR -->
          <version>5</version>

          <!-- Here, we can rename the included files' names -->
          <modules>
            <webModule>
              <groupId>${project.groupId}</groupId>
              <artifactId>
                ${parent.artifactId}-war
              </artifactId>
              <contextRoot>
                /${parent.artifactId}
              </contextRoot>
              <bundleFileName>
                ${parent.artifactId}.war
              </bundleFileName>
            </webModule>
            <ejbModule>
              <groupId>${project.groupId}</groupId>
              <artifactId>
                ${parent.artifactId}-ejb
              </artifactId>
              <bundleFileName>
                ${parent.artifactId}.jar
              </bundleFileName>
```

```
            </ejbModule>
          </modules>

          <archive>
            <!-- Don't forget the Implementation Version -->
            <manifestEntries>
              <Implementation-Version>${project.version}</Implementation-Version>
            </manifestEntries>
          </archive>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

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:

&lt;	<	less than
&gt;	>	greater than
&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark