Difference Between JAR, WAR and EAR Files

In J2EE application, modules are packaged as EAR, JAR and WAR depending on their functionality. This is not any kind of structural difference between these files. All of the are archieved using jar/zip compression.

A Java Application Server has two containers - one is a **Web Container** and the other one is a **EJB Container**.

The Web container has Web applications in light of JSP or the Servlets API - designed particularly for web request handling, so to a greater degree a request/response style of distributed computing. A Web container requires the web module to be bundled as a WAR document - that is a unique JAR file with a web.xml file in the WEB-INF folder.

An EJB container has Enterprise java beans in view of the EJB API intended to give expanded business usefulness, for example, declarative method level security, multiprotocol support. EJB compartments require EJB modules to be bundled as JAR files - these have an ejb-jar.xml files in the META-INF folder.

Enterprise applications may have more than one modules that can either be Web modules (bundled as a WAR document) or EJB modules (bundled as a JAR files) or both. Enterprise applications are packaged as EAR records, these are extraordinary JAR documents containing an application.xml file in the META-INF folder.

Fundamentally EAR records are a superset containing WAR documents and JAR documents. Java Application Servers permit deployment of standalone web modules in a WAR file, however inside they make EAR files as a wrapper around WAR files. Standalone web containers, for example, Tomcat and Jetty doesn't support EAR records - these are not undeniable Application servers. Web applications in these containers are to be deployed as WAR files as it were.

1. JAR File (Java Archive) :

Basically JAR file is compressed using JDk software. There are many other tools available for zipping files. Most of them have one problem i.e. they are platform dependent. A file zipped using winzip cannot be unzipped on Linux. On the other hand JAR file can be zipped and unzipped on any platform where Java is working.

How to create JAR file?

Let's say we have 3 files as **sample1.txt, sample2.txt, sample3.jpeg** and we want to create Demo.jar for these files

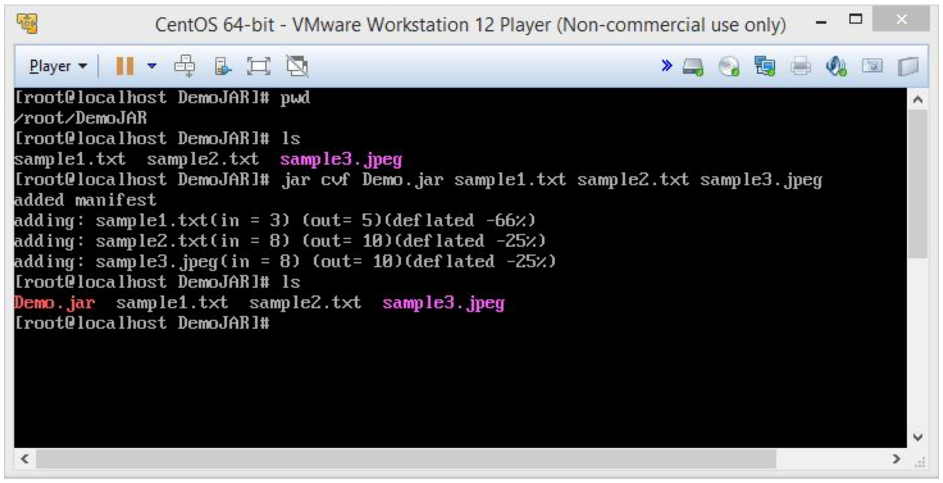
Run this command on terminal :

<p>&nbsp;[root@localhost DemoJAR]# jar cvf Demo.jar sample1.txt sample2.txt sample3.jpeg</p> <div>&nbsp;</div>

'cvf' Options from command are.

c : create a JAR file.
v : display the verbose output at standard output.
f : the jar file name is included.

How to view contents of JAR file?

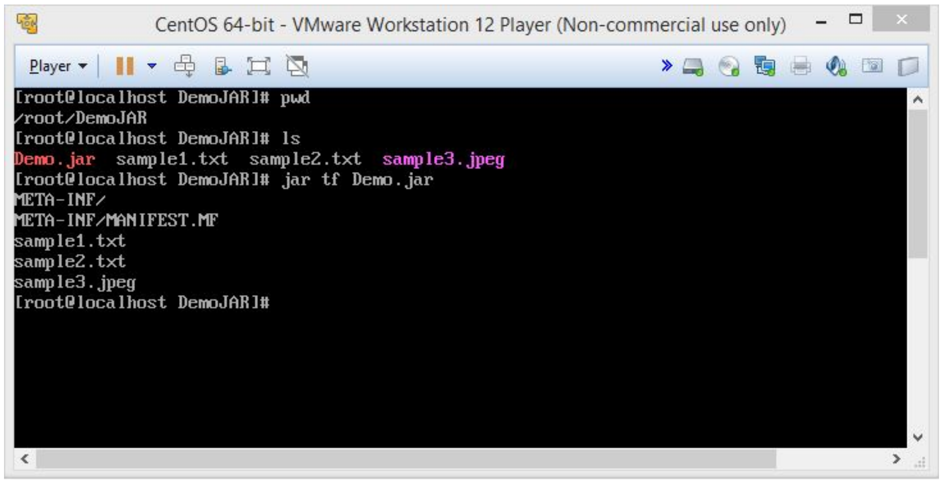

CentOS 64-bit - VMware Workstation 12 Player (Non-commercial use only)
Player

```
[root@localhost DemoJAR]# pwd
/root/DemoJAR
[root@localhost DemoJAR]# ls
sample1.txt  sample2.txt  sample3.jpeg
[root@localhost DemoJAR]# jar cvf Demo.jar sample1.txt sample2.txt sample3.jpeg
added manifest
adding: sample1.txt(in = 3) (out= 5)(deflated -66%)
adding: sample2.txt(in = 8) (out= 10)(deflated -25%)
adding: sample3.jpeg(in = 8) (out= 10)(deflated -25%)
[root@localhost DemoJAR]# ls
Demo.jar  sample1.txt  sample2.txt  sample3.jpeg
[root@localhost DemoJAR]#
```

<p>&nbsp;[root@localhost DemoJAR]# jar tf Demo.jar</p>

t : table of contents.

f : jar file name is included.


CentOS 64-bit - VMware Workstation 12 Player (Non-commercial use only)
Player

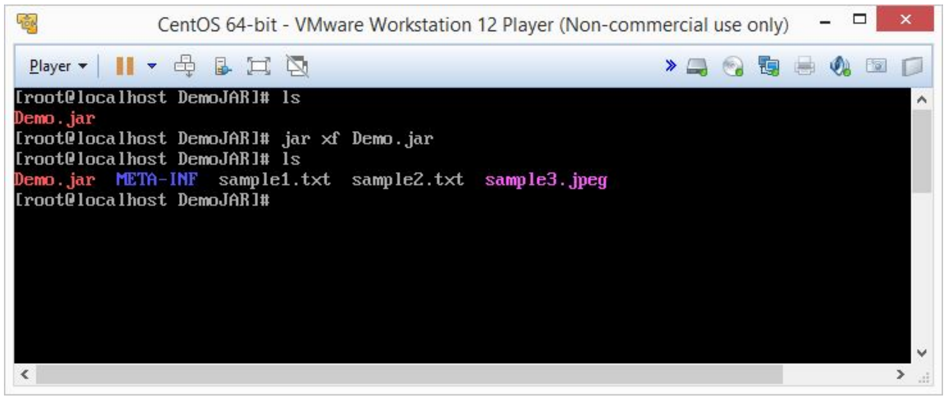
```
[root@localhost DemoJAR]# pwd
/root/DemoJAR
[root@localhost DemoJAR]# ls
Demo.jar  sample1.txt  sample2.txt  sample3.jpeg
[root@localhost DemoJAR]# jar tf Demo.jar
META-INF/
META-INF/MANIFEST.MF
sample1.txt
sample2.txt
sample3.jpeg
[root@localhost DemoJAR]#
```

How to extract JAR file ?

<div>[root@localhost DemoJAR]# jar xf Demo.jar</div> <p>&nbsp;</p>

x : extract the contents.

f : jar file name is included.


CentOS 64-bit - VMware Workstation 12 Player (Non-commercial use only)
Player

```
[root@localhost DemoJAR]# ls
Demo.jar
[root@localhost DemoJAR]# jar xf Demo.jar
[root@localhost DemoJAR]# ls
Demo.jar  META-INF  sample1.txt  sample2.txt  sample3.jpeg
[root@localhost DemoJAR]#
```

2. WAR File (Web Application Archieve) :

A WAR file is fundamentally a JAR document having just Web related Java documents for e.g. Servlets, JSP, HTML, Database Java Beans, web.xml records and so forth which are essential while creating Web applications. WAR file has upper hand that it can be deployed effectively on client machine in a Web server environment. The extension of WAR file is .war however ofcourse made with JAR command as it were.

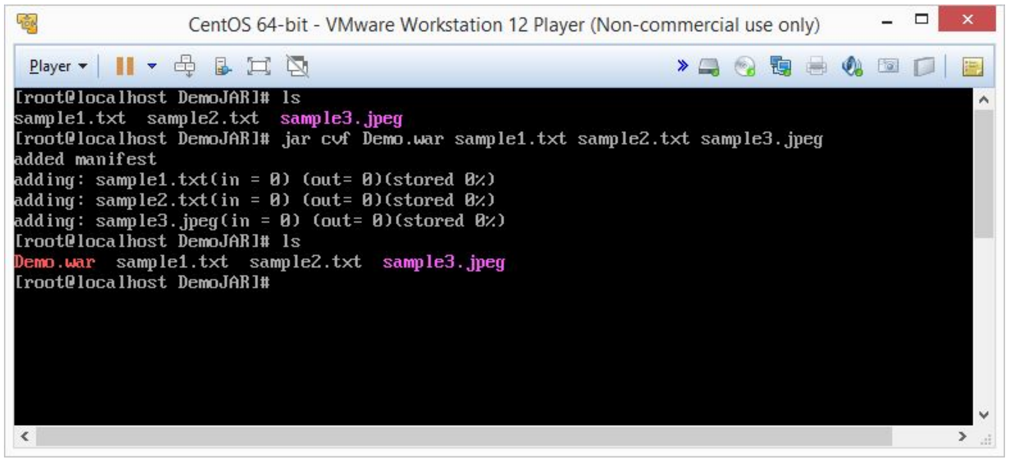
For execution of a WAR file, a Web server or Web holder is required, for instance, **Tomcat or Weblogic or Websphere**. To execute a **JAR** file, basic **JDK** is sufficient. The IDE tools like Eclipse, JBOSS and so on keep up a directory hierarchy structure for WAR documents like WEB-INF organizer and so on.

Let's say we have 3 files as **sample1.txt, sample2.txt, sample3.jpeg** and we want to create Demo.war for these files

How to create WAR file?Let's say we have 3 files as sample1.txt, sample2.txt, sample3.jpeg and we want to create Demo.war for these files

Run this command on terminal :

<p>&nbsp;[root@localhost DemoJAR]# jar cvf Demo.war sample1.txt sample2.txt sample3.jpeg</p>


CentOS 64-bit - VMware Workstation 12 Player (Non-commercial use only)
Player

```
[root@localhost DemoJAR]# ls
sample1.txt  sample2.txt  sample3.jpeg
[root@localhost DemoJAR]# jar cvf Demo.war sample1.txt sample2.txt sample3.jpeg
added manifest
adding: sample1.txt(in = 0) (out= 0)(stored 0%)
adding: sample2.txt(in = 0) (out= 0)(stored 0%)
adding: sample3.jpeg(in = 0) (out= 0)(stored 0%)
[root@localhost DemoJAR]# ls
Demo.war  sample1.txt  sample2.txt  sample3.jpeg
[root@localhost DemoJAR]#
```

'cvf' Options from command are.

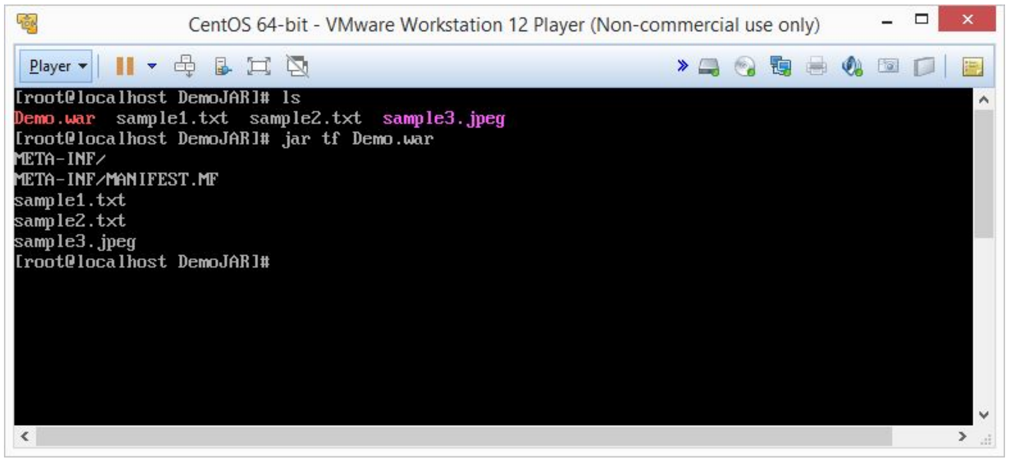
c : create a WAR file.
v : display the verbose output at standard output.
f : the WAR file name is included.

How to view contents of WAR file?

<div>[root@localhost DemoJAR]# jar tf Demo.war&nbsp;</div>

t : table of contents.

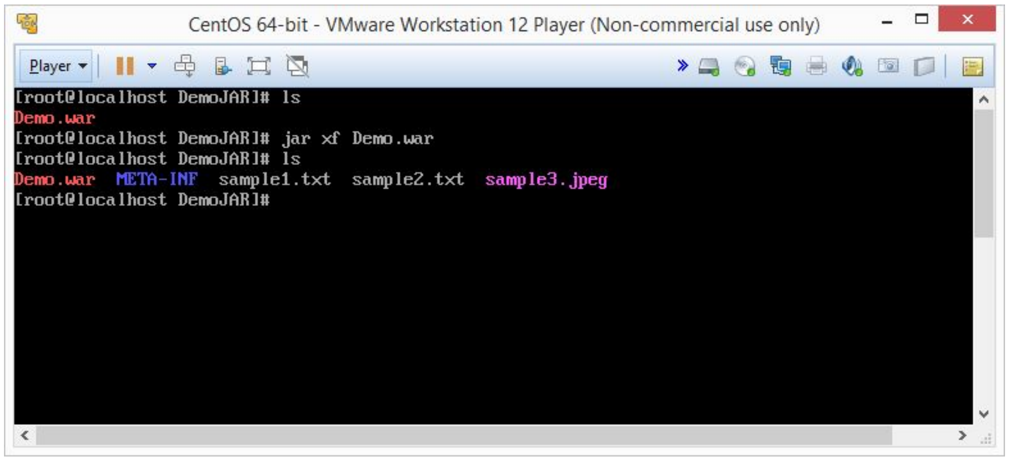
f : WAR file name is included.


CentOS 64-bit - VMware Workstation 12 Player (Non-commercial use only)
Player

```
[root@localhost DemoJAR]# ls
Demo.war  sample1.txt  sample2.txt  sample3.jpeg
[root@localhost DemoJAR]# jar tf Demo.war
META-INF/
META-INF/MANIFEST.MF
sample1.txt
sample2.txt
sample3.jpeg
[root@localhost DemoJAR]#
```

How to extract WAR file ?

<p>[root@localhost DemoJAR]# jar xf Demo.war</p>


CentOS 64-bit - VMware Workstation 12 Player (Non-commercial use only)
Player

```
[root@localhost DemoJAR]# ls
Demo.war
[root@localhost DemoJAR]# jar xf Demo.war
[root@localhost DemoJAR]# ls
Demo.war  META-INF  sample1.txt  sample2.txt  sample3.jpeg
[root@localhost DemoJAR]#
```

3. EAR File (Enterprise Application Archive):

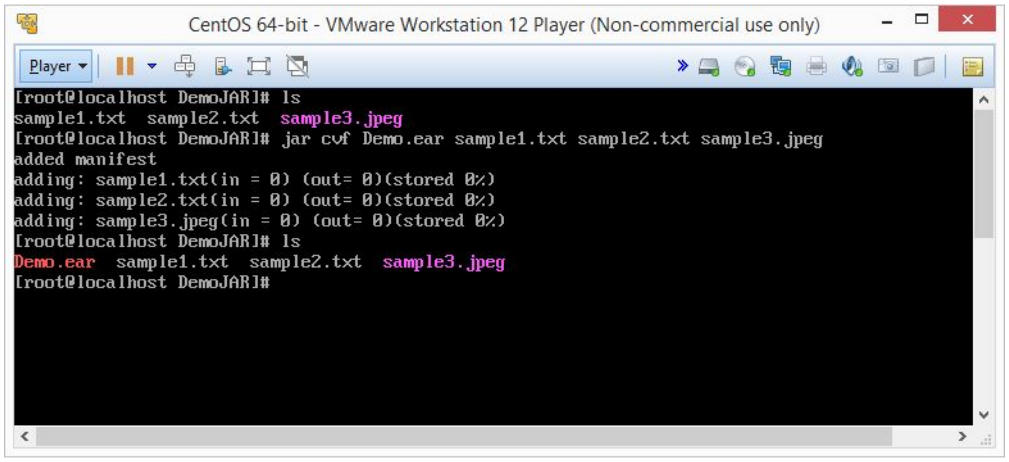
EAR file has Enterprise application related files (J2EE) like XML, EJB modules etc. It is also created with JAR command only but with extension .ear.
EAR file is deployed in an application server.

How to create EAR file?

Let's say we have 3 files as sample1.txt, sample2.txt, sample3.jpeg and we want to create Demo.ear for these files

Run this command on terminal :

<p>&nbsp;[root@localhost DemoJAR]# jar cvf Demo.ear sample1.txt sample2.txt sample3.jpeg</p>



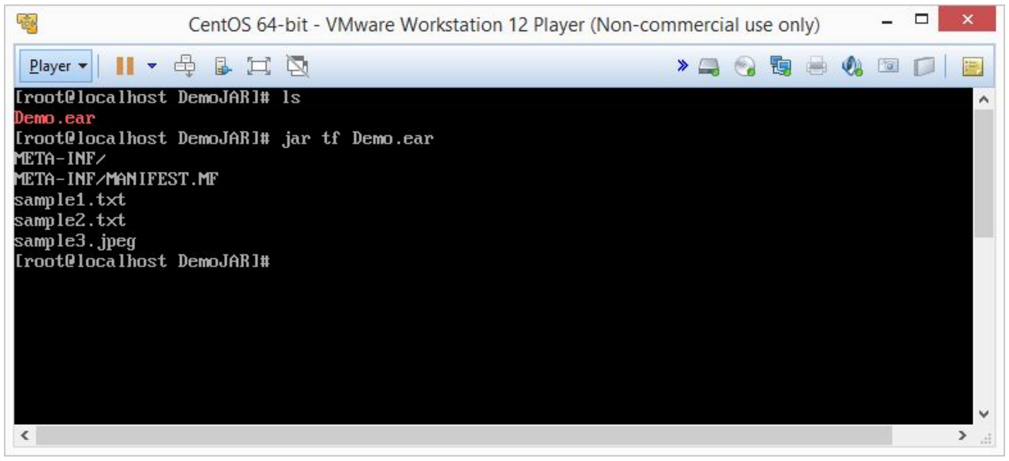
```
[root@localhost DemoJAR]# ls
sample1.txt  sample2.txt  sample3.jpeg
[root@localhost DemoJAR]# jar cvf Demo.ear sample1.txt sample2.txt sample3.jpeg
added manifest
adding: sample1.txt(in = 0) (out= 0)(stored 0%)
adding: sample2.txt(in = 0) (out= 0)(stored 0%)
adding: sample3.jpeg(in = 0) (out= 0)(stored 0%)
[root@localhost DemoJAR]# ls
Demo.ear  sample1.txt  sample2.txt  sample3.jpeg
[root@localhost DemoJAR]#
```

'cvf' Options from command are.

c : create a EAR file.
v : display the verbose output at standard output.
f : the EAR file name is included.

How to view contents of EAR file?

<p>&nbsp;[root@localhost DemoJAR]# jar tf Demo.ear</p>

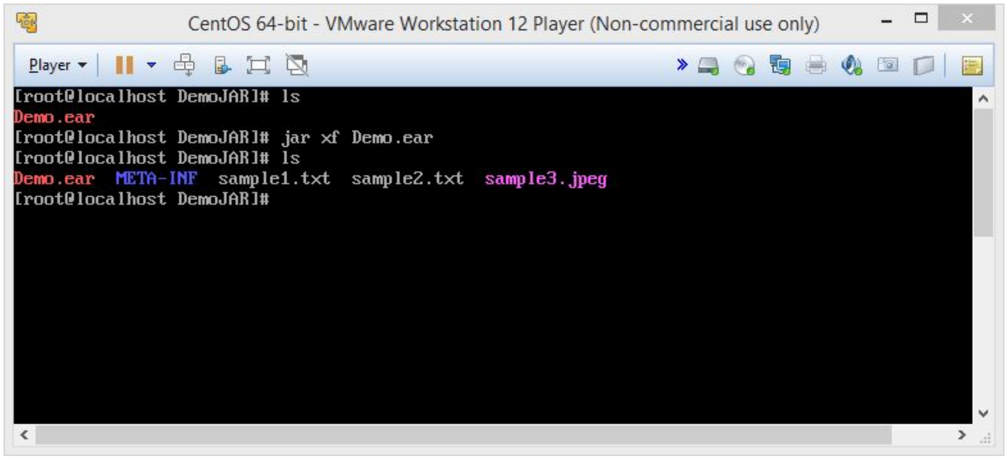


```
[root@localhost DemoJAR]# ls
Demo.ear
[root@localhost DemoJAR]# jar tf Demo.ear
META-INF/
META-INF/MANIFEST.MF
sample1.txt
sample2.txt
sample3.jpeg
[root@localhost DemoJAR]#
```

t : table of contents.
f : EAR file name is included.

How to extract EAR file ?

<p>&nbsp;[root@localhost DemoJAR]# jar xf Demo.ear</p>


CentOS 64-bit - VMware Workstation 12 Player (Non-commercial use only)
Player

```
[root@localhost DemoJAR]# ls
Demo.ear
[root@localhost DemoJAR]# jar xf Demo.ear
[root@localhost DemoJAR]# ls
Demo.ear  META-INF  sample1.txt  sample2.txt  sample3.jpeg
[root@localhost DemoJAR]#
```

x : extract the contents.

f : EAR file name is included.