# Deployable

### Definition

## Deployables are archives (WAR, EAR, etc) that can be deployed in the container

### Explanation

A `Deployable` class is a wrapper class around a physical archive. `Deployable` are constructed by directly instantiating them (e.g. `new WAR(...)` or `new TomcatWAR(...)`) or by using a `DeployableFactory` (e.g. `DefaultDeployableFactory`). There are 2 generic deployable classes:

- `o.c.c.c.deployable.WAR`
- `o.c.c.c.deployable.EAR`
  There are also some container-specific deployables such as:
- `o.c.c.c.deployable.tomcat.TomcatWAR`
- `o.c.c.c.deployable.jboss.JBossWAR`
  They are there to support container extensions to archives (for example, Tomcat supports `context.xml` files located in your WAR's `META-INF` directory, JBoss allows for a `jboss-web.xml` located in your WAR, etc).

The `DeployableFactory` interface offers a principal method for creating a `Deployable`: `DeployableFactory.createDeployable(String containerId, String deployableLocation, DeployableType type)`. `DeployableType` can be `DeployableType.WAR` or `DeployableType.EAR`.

Once you have a `Deployable` instance wrapping your archive, you'll need to deploy it. This can be done either using Static Deployment or using Hot Deployment.

### Example using the Java API

Deploying a WAR in Tomcat 5.x:

```
Container container = new Tomcat5xContainer(
    new
CatalinaStandalineConfiguration("target/tomcat5x"));
container.setHome("c:/apps/tomcat-5.0.29");

WAR war = new WAR("path/to/my.war");

[...]
```

**Example using the Generic API**

```
[...]

DeployableFactory factory = new
DefaultDeployableFactory();
WAR war =
factory.createDeployable("tomcat5x",
"path/to/my.war",
    DeployableType.WAR);
```

**Example using the Ant API**

Statically deploying a WAR in Tomcat 5.x:

```
<cargo containerId="tomcat5x"
home="c:/apps/tomcat-5.0.29" action="start">
  <configuration>
    <war warfile="path/to/my.war"/>
  </configuration>
</cargo>
```

**Example using the Maven 2 plugin**

```
<dependencies>
  [...]
  <dependency>
    <groupId>my.war.groupId</groupId>
    <artifactId>my-war</artifactId>
    <version>1.0.0</version>
    <packaging>war</packaging>
```

```
  </dependency>
  [...]
</dependencies>

[...]

<plugins>
  [...]

  <plugin>
    <groupId>org.codehaus.cargo</groupId>

<artifactId>cargo-maven2-plugin</artifactId>
    <configuration>
      <container>
        [...]
      </container>
      <configuration>
        <type>standalone</type>
        [...]
      </configuration>
      <deployables>
        <deployable>
          <groupId>my.war.groupId</groupId>
          <artifactId>my-war</artifactId>
          <type>war</type>
        </deployable>
      </deployables>
```

```
    </configuration>
  </plugin>
</plugins>
```