Application Classpath

Definition

How to configure the classpath of the application without changing the container's own classpath

> This feature is only available for local containers

> This feature is not available on all containers! Currently, the only containers that support shared classpaths are:
>
> - JBoss 5.x to 6.1.x (inclusive)
> - JBoss 7.x and 7.1.x; by modifying your application's `MANIFEST.MF` file (read more on the how to put an external file in the classpath article from the JBoss website)
> - Jetty 6.x and onwards
> - All Tomcat containers
>
> On other containers, please change the container classpath instead.

Explanation

This topic is not about the classpath requirements to run Cargo (see the Installation page for this); it's about configuring the classpath for the application being executed on a container.

Some environments may require you to add some libraries to your application, without exposing them to the server itself. This concept is called a **Shared Classpath**.

Example using the Java API

Starting a WAR on Tomcat 6.x with some additional classpath entries:

```
LocalConfiguration configuration = new
Tomcat6xStandaloneLocalConfiguration("target
/tomcat6x");
Deployable war = new
WAR("target/war/grails-example-without-libs.
war");
configuration.addDeployable(war);

InstalledLocalContainer container = new
Tomcat6xInstalledLocalContainer(configuratio
n);
container.setHome("/srv/tomcat/catalina-home
");

List<String> jars = new List<String>();
for (File jar : new
File("/opt/grails").listFiles())
{
    if (jar.isFile())
    {
        jars.add(jar.getAbsolutePath());
    }
}

container.setSharedClasspath(jars.toArray())
;

container.start();
```

Example using the Ant API

Starting a WAR on Tomcat 6.x with some additional classpath entries:

```
<cargo containerId="tomcat6x"
home="/srv/tomcat/catalina-home"
action="start">
  <sharedClasspath>
    <fileset dir="/opt/grails">
      <include name="*.jar"/>
    </fileset>
  </sharedClasspath>
  <configuration home="target/tomcat6x">
    <deployable type="war"
file="target/war/grails-example-without-libs
.war"/>
  </configuration>
</cargo>
```

Example using the Maven Plugin

Configuring a Tomcat6 container with a com.foo bar-api-1.0.jar dependency provided using the shared classpath:

```
[...]
<dependencies>
  <dependency>
    <groupId>com.foo</groupId>
    <artifactId>bar-api</artifactId>
    <scope>provided</scope>
    <version>1.0</version>
  </dependency>
</dependencies>
[...]
<plugin>
  <groupId>org.codehaus.cargo</groupId>

<artifactId>cargo-maven2-plugin</artifactId>
  <configuration>
    <container>
      <containerId>tomcat6x</containerId>
      [...]
      <dependencies>
        <dependency>
          <groupId>com.foo</groupId>
          <artifactId>bar-api</artifactId>
          <classpath>shared</classpath>
        </dependency>
      </dependencies>
    </container>
    [...]
  </configuration>
</plugin>
```