

Manually Deploying a Web Module to OC4J; Author: James L. Colestock

Software Used: 9iAS Releases (1.0.2.2.2); OC4J (1.0.2.2.1); JDeveloper (9.0.2.822)

Make Sure that OC4J is installed

On the Version in Question, OC4J must be installed separately. Later versions bundle OC4J with the 9iAS Install

To Install OC4J:

Download oc4j.zip from Oracle
Unzip the file into the \$ORACLE_HOME:

```
% cd $ORACLE_HOME
% unzip oc4j.zip
% cd $ORACLE_HOME/j2ee/home
```

Install OC4J via the following command:

```
% java -jar orion.jar -install
```

Enter an Administrative Password when prompted

Start OC4J (Oracle Containers for Java):

```
% ${ORACLE_HOME}/Apache/jdk/bin/java -jar \
${ORACLE_HOME}/j2ee/home/orion.jar \
-config ${ORACLE_HOME}/j2ee/home/config/server.xml
```

Create a Simple J2EE Web Module

Example Servlet Class

```
package jcolesto;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;

public class FirstAppServlet extends HttpServlet
{
private static final String CONTENT_TYPE = "text/html; charset=windows-1252";
public void init(ServletConfig config) throws ServletException
{
super.init(config);
} public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
response.setContentType(CONTENT_TYPE);
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<head><title>FirstAppServlet</title></head>");
out.println("<body>");
out.println("<p>Can you see the First App Servlet</p>");
out.println("</body></html>");
out.close();
}
}
```

After Compiling and Testing ensure that this Servlet Class has a corresponding entry in the web.xml file. Just provide

values for essential elements.

Example web.xml file:

```
<?xml version = '1.0' encoding = 'windows-1252'?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
  2.2//EN" "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
< web-app>
<description>ColestockFirstAppServlet</description>
< servlet>
<servlet-name>FirstAppServlet</servlet-name>
<servlet-class>jcolesto.FirstAppServlet</servlet-class>
< /servlet>
< session-config><session-timeout>30</session-timeout>
< /session-config>
< mime-mapping>
<extension>html</extension>
<mime-type>text/html</mime-type>
< /mime-mapping>
< mime-mapping><extension>txt</extension>
<mime-type>text/plain</mime-type>
< /mime-mapping>
< welcome-file-list>
<welcome-file>index.jsp</welcome-file>
<welcome-file>index.html</welcome-file></welcome-file-list>
< /web-app>
```

Package the WAR file

Although JDeveloper and other IDEs have the ability to deploy a J2EE application via mechanisms such as RMI, this document will focus on the manual deployment.

Example:

```
% jar -cf FirstAppServlet META-INF WEB-INF or jar -cf FirstAppServer.war *
```

```
% jar -tf FirstAppServlet.war
```

```
META-INF/
META-INF/MANIFEST.MF
WEB-INF/
WEB-INF/classes/
WEB-INF/classes/Colestock.cdi
WEB-INF/classes/connections.xml
WEB-INF/classes/jcolesto/
WEB-INF/classes/jcolesto/FirstAppServlet.class
WEB-INF/src/
WEB-INF/src/jcolesto/
WEB-INF/src/jcolesto/FirstAppServlet.java
WEB-INF/web.xml
```

Add the application.xml and Package the Ear

Create an application.xml file by adding Web Module and corresponding War file entries

Example:

```
<?xml version="1.0"?>
<!DOCTYPE application PUBLIC "-//Sun Microsystems, Inc.//DTD J2EE Application 1.2
//EN" "http://java.sun.com/j2ee/dtds/application_1_2.dtd">
```

```

<application>
<display-name>FirstAppServlet</display-name>
<module>
<web>
<web-uri>FirstAppServlet.war</web-uri>
<context-root>/</context-root>
</web>
</module>
</application>

```

Now Jar into a EAR file:

```

mkdir META-INF
cd META-INF
vi application.xml
jar -cf FirstAppServlet.ear .
% jar -tf FirstAppServlet.ear

```

```

META-INF/
META-INF/MANIFEST.MF
META-INF/application.xml
FirstAppServlet.war

```

Deploy the Application and make Orion configuration file changes

Deploy the EAR via the Orion admin.jar

Example:

```

% $ORACLE_HOME/Apache/jdk/bin/java -jar $J2EE_HOME/admin.jar ormi://<localhost>:<rmi-port>
admin <password> -deploy -file FirstAppServlet.ear -deploymentName FirstAppServlet

```

Next you will need to Bind the Application to a Web Site

You will need to know the destination web site for your application - the default web site may be used or you can set up your own. In our example, we create our own by creating a web-site specifically for this application.

A xml file was created and placed in the \$J2EE_HOME/config directory with the rest of the config files

"First-app-site.xml" 8 lines, 427 characters

```

<?xml version="1.0"?>
<!DOCTYPE web-site PUBLIC "Orion Web-site" "http://xmlns.oracle.com/ias/dtds/web-site.dtd">

<web-site port="8080" display-name="Default Oracle9iAS Containers for J2EE Web Site">
<default-web-app application="FirstAppServlet" name="FirstAppServlet" />
<web-app application="FirstAppServlet" name="FirstAppServlet" root="/FirstAppServlet" />
<access-log path="../log/First-app-site.log" />
</web-site>

```

Bind the Application to a Web Site with the following command:

```

% $ORACLE_HOME/Apache/jdk/bin/java -jar ../admin.jar ormi://<localhost>:<rmi-port>
admin <password> -bindWebApp FirstAppServlet FirstAppServlet <default-web-site ||
FirstAppServlet
/FirstAppServlet

```

This should create the following entry in the server.xml (if not edit as appropriate). In our example, the new web site and xml file are referenced:

```
<application name="FirstAppServlet" path="../applications/FirstAppServlet.ear" auto-start="true" />
<web-site path="./First-app-site.xml" />
```

If you had chosen the default-web-site then the server.xml entry would have simply been:

```
<application name="FirstAppServlet" path="../applications/FirstAppServlet.ear" auto-start="true" />
```

And an entry in the default-web-site.xml would have been appropriate:

```
<web-app application="FirstAppServlet" name="FirstAppServlet" root="/FirstAppServlet" />
```

Now add the Proxying Entries to the httpd.conf file to allow to tunnel connections from Apache to Orion, this also enables SSL

```
% vi $APACHE_HOME/conf/httpd.conf
```

Example:

```
<IfModule mod_proxy.c>
```

```
ProxyRequests On
```

```
ProxyPass /j2ee/FirstAppServlet/ http://<hostname>:<web-site-port>/FirstAppServlet/
```

```
ProxyPassReverse /j2ee/FirstAppServlet/ http://<hostname>:<web-site-port>/FirstAppServlet/
```