



EntireX XML Servlet Deployment Configuration

December 2004

Theo Beack
Software AG, New York, NY

Table of Contents

Document History	4
Introduction	5
1.1 Tomcat 4.1.x & 5.x Deployment	6
1.2 JBoss 3.2.x & 4.0.0 Deployment	10

Document History

Document Data

Document Name	EntireX XML Servlet Deployment Configuration
Document Description	This document describes the steps to follow when deploying the EntireX XML Servlet and the XML Wrapper under Tomcat or JBoss.
Author	Theo Beack

Revision History

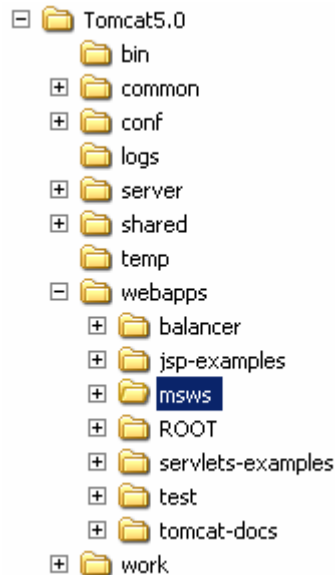
Version	Date	Author	Brief Summary of Changes
1	12/16/04	Theo Beack	Initial Draft

Introduction

The purpose of this document is to provide a brief overview and some simple guidelines on how to deploy and configure the EntireX XML Servlet & XML Wrapper mapping files generated by the EntireX Communicator Workbench. This document describes the steps to follow in order to deploy the EntireX XML Servlet under Apache Tomcat or JBoss.

1.1 Tomcat 4.1.x & 5.x Deployment

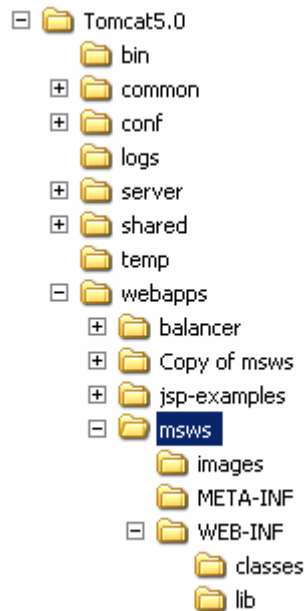
- Step 1 Create a new application context within the Tomcat **"webapps"** directory. In this example I created a new application called **"msws"**.



- Step 2 Unzip the **"entirex.war"** file which can be located in the **"C:\Program Files\Software AG\EntireX\Webapps"** directory. Unzip the contents of **"entirex.war"** into the newly created application context within your Tomcat webapps directory; in this example it will be the **"msws"** directory. The contents of the **"entirex.war"** file is as follows:

Name	Path
example.xmm	
index.html	
xml-init.xml	
logo_enx_comm_m.gif	images\
PPT_BrandStrip.jpg	images\
sag_logo_xcomp_250.gif	images\
Manifest.mf	meta-inf\
web.xml	web-inf\
ExampleServlet.class	WEB-INF\classes\
entirex.jar	WEB-INF\lib\

- Step 3 After the unzip you will note that a number of new subdirectories have been added to the **"msws"** directory and the directory structure should look something like this:



You will also find that two configuration files have been added, these files are:

- **"xml-init.xml"** - it has been added to the root directory of **"msws"**.
- **"web.xml"** - it has been added to the **"msws\web-inf"** directory.

These two files have to be modified in order to deploy the generated XML Wrapper Mapping file.

Step 4 Deploy the generated XML Mapping file into the root directory of **"msws"**. In this example the example mapping file is called **"webserv.xmm"**.

Step 5 Once this is done it is time to configure the **"xml-init.xml"** file. The default file extracted from the **"entirex.war"** file, should look something like this:

```
<?xml version="1.0" encoding="utf-8"?>
<EntireXMLInit Version="7.2.1" xmlns="
http://namespaces.softwareag.com/entirex/xml/rt">
  <xmms>
    <exx-xmm>/example.xmm</exx-xmm>
  </xmms>
</EntireXMLInit>
```

To make the required changes add the Broker id and mapping file to the **"xml-init.xml"** file. After the changes have been made the file should look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<EntireXXMLInit Version="7.2.1" xmlns="
http://namespaces.softwareag.com/entirex/xml/rt">
  <xmms exx-brokerID="ETB172:5172">
    <exx-xmm>/webserv.xmm</exx-xmm>
  </xmms>
</EntireXXMLInit>
```

- Step 6 Next you should edit the "**web.xml**" file located in the "**webapps\msws\web-inf**" directory. The file should look something like this:

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

<!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>
  <display-name>Example Web Services Deployment</display-name>
  <servlet>
    <servlet-name>xmldr</servlet-name>
    <servlet-class>com.softwareag.entirex.xml.rt.XMLServlet</servlet-class>
    <init-param>
      <param-name>exx-xml-init</param-name>
      <param-value>/xml-init.xml</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>xmldr</servlet-name>
    <url-pattern>/xmldr</url-pattern>
  </servlet-mapping>
</web-app>
```

- Step 7 Start Tomcat and verify that the application context is available and properly started. You can do this by opening a browser and typing the Tomcat Manager URL, for example:

<http://localhost:8800/manager/html>

This should open the "**Tomcat Web Application Manager**" in the browser window, which will list all the different application running states. If "**msws**" started properly its "*running status*" will be displayed as "**true**".

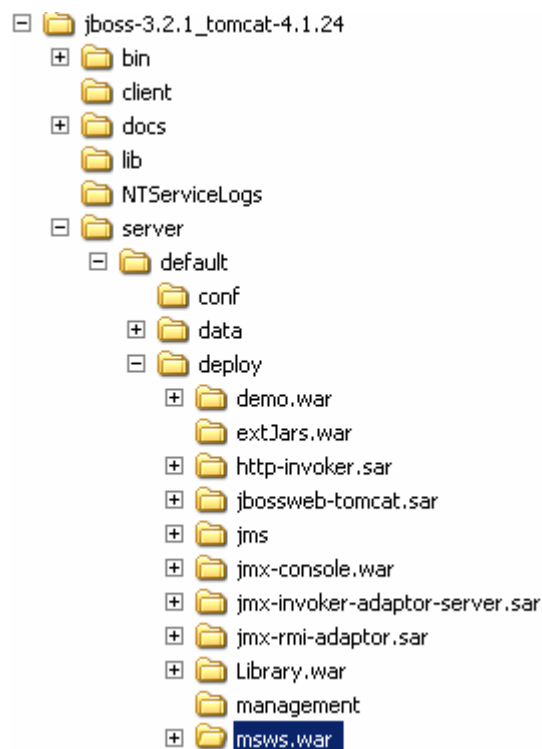
Applications		
Path	Display Name	Running
/	Welcome to Tomcat	true
/Copy of msws	Morgan Stanley Web Services PoC	true
/admin	Tomcat Administration Application	true
/balancer		true
/jsp-examples	JSP 2.0 Examples	true
/manager	Tomcat Manager Application	true
/msws	Example Web Services Deployment	true

Step 8 You are now ready to test the deployment by sending a request (in the form of a XML document of SOAP Request document, depending on the mapping selected in the Workbench) to the EntireX XML Servlet.

The URL to use would be: <http://localhost:8800/msws/xmlrt>

1.2 JBoss 3.2.x & 4.0.0 Deployment

- Step 1 Create a new application context within the JBoss **"deploy"** directory. In this example I created a new application called **"msws.war"**, which is located in **"C:\jboss-3.2.1_tomcat-4.1.24\server\default\deploy"**



Note: Please note that in order to deploy your application under Jboss you need to create a **".war"** extension to the application directory name. You don't need to do this if you plan to package your application into a **".war"** or **".ear"** file.

- Step 2 Unzip the **"entirex.war"** file which can be located in the **"C:\Program Files\Software AG\EntireX\Webapps"** directory. Unzip the contents of **"entirex.war"** into the newly created application context within your JBoss deploy directory; in this example it will be the **"msws.war"** directory. The contents of the **"entirex.war"** file is as follows:

Name	Path
example.xmm	
index.html	
xml-init.xml	
logo_enx_comm_m.gif	images\
PPT_BrandStrip.jpg	images\
sag_logo_xcomp_250.gif	images\
Manifest.mf	meta-inf\
web.xml	web-inf\
ExampleServlet.class	WEB-INF\classes\
entirex.jar	WEB-INF\lib\

Step 3 After the unzip you will note that a number of new subdirectories have been added to the **"msws.war"** directory and the directory structure should look something like this:

```

jboss-3.2.1_tomcat-4.1.24
├── bin
├── client
├── docs
├── lib
├── NTServiceLogs
├── server
│   ├── default
│   │   ├── conf
│   │   ├── data
│   │   └── deploy
│   │       ├── demo.war
│   │       ├── extJars.war
│   │       ├── http-invoker.sar
│   │       ├── jbossweb-tomcat.sar
│   │       ├── jms
│   │       ├── jmx-console.war
│   │       ├── jmx-invoker-adaptor-server.sar
│   │       ├── jmx-rmi-adaptor.sar
│   │       ├── Library.war
│   │       ├── management
│   │       └── msws.war
│   │           ├── images
│   │           ├── META-INF
│   │           └── WEB-INF
│   │               ├── classes
│   │               └── lib

```

You will also find that two configuration files have been added, these files are:

- **"xml-init.xml"** - it has been added to the root directory of **"msws"**.
- **"web.xml"** - it has been added to the **"msws.war\web-inf"** directory.

These two files have to be modified in order to deploy the generated XML Wrapper Mapping file.

Step 4 Deploy the generated XML Mapping file into the root directory of **"msws.war"**. In this example the example mapping file is called **"webserv.xmm"**.

Step 5 Once this is done it is time to configure the **"xml-init.xml"** file. The default file extracted from the **"entirex.war"** file, should look something like this:

```
<?xml version="1.0" encoding="utf-8"?>
<EntireXXMLInit Version="7.2.1" xmlns="
http://namespaces.softwareag.com/entirex/xml/rt">
  <xmms>
    <exx-xmm>/example.xmm</exx-xmm>
  </xmms>
</EntireXXMLInit>
```

To make the required changes add the Broker id and mapping file to the **"xml-init.xml"** file. After the changes have been made the file should look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<EntireXXMLInit Version="7.2.1" xmlns="
http://namespaces.softwareag.com/entirex/xml/rt">
  <xmms exx-brokerID="ETB172:5172">
    <exx-xmm>/webserv.xmm</exx-xmm>
  </xmms>
</EntireXXMLInit>
```

Step 6 Next you should edit the **"web.xml"** file located in the **"deploy\msws.war\web-inf"** directory. The file should look something like this:

```

<?xml version="1.0" encoding="utf-8"?>

<!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>
  <display-name>Example Web Services Deployment</display-name>
  <servlet>
    <servlet-name>xmlrt</servlet-name>
    <servlet-class>com.softwareag.entirex.xml.rt.XMLServlet</servlet-class>
    <init-param>
      <param-name>exx-xml-init</param-name>
      <param-value>/xml-init.xml</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>xmlrt</servlet-name>
    <url-pattern>/xmlrt</url-pattern>
  </servlet-mapping>
</web-app>

```

- Step 7 Start JBoss and verify that the application context is available and properly started. You can do this by opening a browser and typing the newly created application context, for example:

<http://localhost:8800/msws/xmlrt>

This should open the "**EntireX XML Servlet**" information in the browser window, which will contain information about the Servlet version, the date it was built, the supported Servlet API version, etc. Once you get this page you are ready to test.

EntireX XMLServlet, Version: 7.2.1.0

Build: 2004-09-14 12:44, Date: 09 Sep 2004

Servlet Container: Apache Tomcat/5.0.28

supported Servlet API version: 2.4

Header/Parameter	Value
exx-xml-init	/xml-init.xml

exx-xmln	exx-brokerID	exx-service
/webserv.xmln	ETB172:5172	RPC/WEBSERV/CALLNAT

Incoming document is empty.

- Step 8 You are now ready to test the deployment by sending a request (in the form of a XML document of SOAP Request document, depending on the mapping selected in the Workbench) to the EntireX XML Servlet. The URL to use would be: <http://localhost:8800/msws/xmlrt>