HOW TO DEPLOY AN EJB APPLICATION IN WEBLOGIC SERVER 11GR1

Last update: June 2011
Table of Contents

1 PURPOSE OF DOCUMENT ......................................................... 2
  1.1 WHAT IS THE USE FOR THIS DOCUMENT ................................ 2
  1.2 PREREQUISITES .................................................................. 2
  1.3 BEFORE DEPLOYING ............................................................. 3
  1.4 STEPS ................................................................................. 3
    1.4.1 Start Weblogic ............................................................ 3
    1.4.2 Create data sources ...................................................... 4
    1.4.3 Deploy an application .................................................. 9
    1.4.4 Create a Queue ............................................................ 15
    1.4.5 Connection Pool .......................................................... 20
1 Purpose of document

1.1 What is the use for this document

This document explains how to deploy to WebLogic Server 11gR1 applications obtained from the EJB transformation engine.

1.2 Prerequisites

WebLogic Application Server allows working with different kinds of operating systems and databases, but for the sake of simplicity, this document uses as example a deployment for Windows XP. The database used in this document is Microsoft SQL Server.

In this case, the only prerequisite to deploy an EJB application in WebLogic is:

1-  WebLogic Application Server
1.3 Before deploying

Applications generate with the audit log files generation option need an additional library to run properly, this library is log4j.jar. A typical WebLogic installation contains this library, but maybe not the last version.

Last version of the library can be freely downloaded from http://logging.apache.org/log4j/1.2/download.html, and it has to be copied to the 
<WebLogicInstallationPath>/wlserver_10.3/beehive/apache-beehive-svn-snapshot/lib/common
and

1.4 Steps

1.4.1 Start Weblogic

To start WebLogic application server launch “Start Admin Server for Weblogic Server Domain” and start the “Admin Server”

Figure 1: Start the Administration Console of WebLogic

WebLogic asks for a Login and a Password, if during the installation any user had been created the default user and password is weblogic.

Figure 2: Administration Console, login and password
1.4.2 Create data sources

To deploy an application two datasources have to be created, one for Queries, and one for Services, with different JNDI names. These names are formed by using the package name of the application and the suffix ‘DB_SERVICE’ (data source for services) or ‘DB_QUERY’ (queries).

To create these data sources follow the next steps.

Go to Services →JDBC → Data Sources → New

Name and JNDI Name:

The name given to the datasources can be found in the ‘Constants.java’ file, in the ‘global’ package. If the name of our application is ‘ApplicationName’, our data source has to be named ‘ApplicationNameDB_QUERY’ when creating the Data source for queries, and ‘ApplicationNameDB_SERVICE’ when creating the Data source for services.

Use the same values for ‘JNDI Name’.

Database type and Database Driver:

In ‘Database Type’ choose MS SQL Server and in the option ‘Database Driver’ ‘*BEAS MS SQL Server Driver (Type 4)’ must be selected. In this How To the application is deployed to access Microsoft SQL Server. If the user needs to deploy an application accessing another database, the corresponding database type and driver have to be selected.

**NOTE:** In the list of ‘Database Driver’, if an * appears, this means that the drivers are installed in WebLogic Server; names without asterisk mean that the drivers are not installed.

![Figure 3: Creating a data source in WebLogic. JDBC data source properties.](image)

When all the options have been set click the ‘Next’ button to select the transaction options.
Figure 4: Creating a data source in WebLogic. Database driver selection.

Choose a new JDBD Data Source.

Database Properties
The following properties will be used to identify your new JDBD data source.

Database Type: MS SQL Server

What database driver would you like to use to create database connections? Note: * indicates that this driver is explicitly supported by Oracle Weblogic Server.

Database Driver:
Oracle's MS SQL Server Driver (Type 4) Version 7.0, 2000, 2005, 2008

Figure 5: Creating a data source in WebLogic. Transaction options.

Leave all the options by default and click 'Next'.
Now, the connection properties must be set:

**Database Name:** Name of the database

**Host Name:** Name or IP address of the database server

**Database User Name and password:** User who has these database permissions and also, the password (in the two final fields).

Figure 6: Creating a data source in WebLogic. Connection properties.

Click ‘**Next**’.
Now, the configuration can be tested with 'Test Table Name', for example executing a query in the Data Base that has been created before.

![Create a New J2EE Data Source](image)

**Figure 7: Creating a data source in WebLogic. Test table name1.**
Figure 8: Creating a data source in WebLogic. Testing table name 2.
If everything is all right, a success message will be shown.

Click 'Next', and select the server on which the user wants to deploy the application.

Click Finish.

**1.4.3 Deploy an application**

Once the two data sources have been created the application can be deployed.
First of all click the Lock & Edit button to modify, add or delete items in this domain.
Go to 'Deployments'. All the applications installed in the WebLogic application server will be shown. Press 'Install' to add a new one.
Figure 11: Deploy application. Install a new application.

Press `upload your file(s)` to navigate to the folder where the `.ear` file is located.

Figure 12: Deploy application. Upload the ear file.

Select the `.ear` file to deploy, and press `Next`:

Figure 13: Deploy application. Select the ear file.
Select the application and press ‘Next’:

![Messages]

The file AppName.ear has been uploaded successfully to C:\Oracle\Middleware\user_projects\domain\base_domain\servers\AdminServer\upload

![Install Application Assistant]

**Locate deployment to install and prepare for deployment.**

Select the file path that represents the application root directory, archive file, exploded archive directory, or application module descriptor that you want to install. You can also enter the path of the application directory or file in the path field.

**Note:** Only valid file paths are displayed below. If you cannot find your deployment files, upload your file(s) and/or confirm that your application contains the required deployment descriptors.

**Paths:**

- **Recently Used Paths:**
  - C:\Oracle\Middleware\user_projects\domain\base_domain\servers\AdminServer\upload\AppName.ear
- **Current Location:**
  - C:\Oracle\Middleware\user_projects\domain\base_domain\servers\AdminServer\upload

![Figure 14: Deploy application. Install and prepare for deployment.]

Select ‘Install this deployment as an application’ and press ‘Next’:

![Install Application Assistant]

**Choose targeting style.**

Targets are the servers, clusters, and virtualhosts on which this deployment will run. There are several ways you can target an application.

- **Install this deployment as an application**
  - The application and its components will be targeted to the same location. This is the most common usage.
- **Install this deployment as a library**
  - Application libraries or deployments that are available for other deployments to share. Libraries should be available on all of the targets running their referencing applications.

![Figure 15: Deploy application. Choose targeting style]

Leave all the optional settings by default, and press ‘Next’:
How to deploy a WebLogic Server 11gR1 application server /BLT_TREJB2

Figure 16: Deploy application. Optional Settings.

All the options by default and press ‘Finish’:

Figure 17: Deploy application. Additional configuration and Summary.
Finally press ‘Save’:

**Figure 18:  Deploy application. Save.**
Figure 19: Deploy application. Activate changes.
1.4.4 Create a Queue

In case that the user requires to deploy an application with Triggers, the creation of a queue is necessary. This is because the triggers that will be executed are messages added to this queue.

First, the user has to create the JMS (Java Message Service) Server. JMS Servers are environment-related configuration entities that act as management containers for destination resources within JMS modules, that will be created later.

Go to Services -> Messaging -> JMS Servers in order to create a new JMS Server.

Select ‘New’

**Figure 20: Creating a queue. JMS (Java Message Service) server.**

Enter the JMS Server name, for instance JMSServer-[ApplicationName], the rest of the values will be the default values. Click ‘Next’.

**Figure 21: Creating a queue. JMS server Properties.**

Select the target server instance where the JMS Server will be deployed and click Finish.

**Figure 22: Creating a queue. JMS server target server instance.**
Next step will be to create the JMS module and the associated queue.

Go to Services → Messaging → JMS Modules in order to create the new JMS.

Select 'Lock and Edit' and click 'New'.

**Figure 23: Creating a queue. JMS (Java Message Service) module.**

Enter the name of the 'System Module', for instance JMS_[ApplicationName], the rest of values will be the default values. Click 'Next'.

**Figure 24: Creating a queue. Creating JMS (Java Message Service) system module.**

Select the Target and press 'Next'
Once here, the user has created the JMS module and now, the queue must be created as well. Select the option to add new resources to the JMS Module, to create a new Queue associated to this JMS Module. And press ‘Finish’.

There the user can see the associated resources to the JMS Module. In the example shown here there are no resources, because the user has not created one yet.
Press ‘New’ in order to create a new Resource. Now, the different resources that can be created appear. Then select ‘Queue’ and click ‘Next’.

![Figure 28: Creating a queue. JMS module. Create a Queue.](image)

Enter a name for the queue and the JNDI and click ‘Next’.

It is important to notice that the JNDI name has to be the same as the one that is looked for in the application and it has to be named the same way it was specified in the deployment descriptor, as well.

The name of the queue in the application will be ‘ApplicationName_Triggers’, this name can be changed but then, it has to be changed in the code and in the descriptor as well (and it has to be the same name as in the application server).

![Figure 29: Creating a queue. JMS module. JMS Destination properties.](image)
Press 'Next' and select the Create new secondary deployment option.

![Figure 30: Creating a queue. JMS module. Create a new subdeployment.](image)

Leave the default name and press 'Ok':

![Figure 31: Creating a queue. JMS module. Create a new subdeployment.](image)

Select the JMS Server target server instance and press 'Finish':

![Figure 32: Creating a queue. JMS module. Select the JMS Server Target server instance.](image)

The queue has been created.
1.4.5 Connection Pool

When the application has global operations, it is necessary to modify the connection pool. So, select the service Datasource (ApplicationNameDB_SERVICE)

Figure 34: Connection pool. Service data source.

Select ‘Connection Pool’ and click ‘Lock and Edit’
Specify the same value for "Initial Capacity" and "Maximum Capacity". It is recommended to have a value greater than or equal to 25.

![Connection pool configuration settings](image)

Figure 35: Connection pool. Capacity.

Save and activate the changes.