



BEA WebLogic Server™

Weblogic Tutorials

Release 8.0
Document Date: December 2002
Revised: December 28, 2002

Copyright

Copyright © 2003 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

Contents

1. Porting and Deploying Smart Ticket with WebLogic Builder Tutorial	
Tutorial Overview	1-1
2. Porting and Deploying Smart Ticket with WebLogic Builder	
Setting Up Applications and Environment	2-2
Converting and Tuning with WebLogic Builder	2-3
Administration Tasks	2-5
Deploying and Running	2-7
Summary	2-11

Porting and Deploying Smart Ticket with WebLogic Builder Tutorial

Tutorial Overview

This example shows a rapid deployment scenario for Sun's BluePrint wireless application, Smart Ticket. We use WebLogic Builder to generate and edit the WebLogic Server-specific deployment descriptor files and to deploy the application on WebLogic Server.

WebLogic Builder is a visual environment for generating and editing an application's deployment descriptor files. You can view descriptor files while you visually edit them in WebLogic Builder, and you won't need to make textual edits to the XML. See [WebLogic Builder](http://edocs.bea.com/wls/docs70/wlbuilder/index.html) at <http://edocs.bea.com/wls/docs70/wlbuilder/index.html>.

Porting and Deploying Smart Ticket with WebLogic Builder

This document contains following sections:

- [Setting Up Applications and Environment](#)
 - Download and install WebLogic Server 7.0
 - Download and install Smart Ticket
 - Download and install the Sun Wireless Toolkit
 - Set your environment
 - Build Smart Ticket
- [Converting and Tuning with WebLogic Builder](#)
 - Use WebLogic Builder to generate weblogic.xml and weblogic-ejb-jar.xml
 - Use WebLogic Builder to edit descriptors
- [Administration Tasks](#)
 - Start server (WebLogic Server's Examples server)
 - Use WebLogic Server Administration Console to configure data sources
 - Modify populate.bat script and SQL query to use the Pointbase sample database
- [Deploying and Running](#)
 - Use WebLogic Builder to deploy Smart Ticket
 - Launch Smart Ticket.

- Query and write to the data sources to create a user account and reserve movie tickets.

Setting Up Applications and Environment

You'll need the following to run this example.

- WebLogic Server 7.0
- Smart Ticket
- Sun's wireless toolkit, J2ME
- `smarticketPointBase.sql` (attached)

Here are the installation and build tasks and the path-settings you'll need.

1. Download and install WebLogic Server 7.0.

Download WebLogic Server 7.0 from <http://www.bea.com> and install it to a location we will hereafter call *WL_HOME*. By default, *WL_HOME* is `c:\bea\weblogic700`.

2. Download and install Smart Ticket 1.1.

Download the Smart Ticket demo application source code at:
<http://developer.java.sun.com/developer/releases/smarticket/>. Extract it into a new directory on your machine. We will call this directory *SMARTICKET_HOME*.

3. Download and install J2ME to a location we will call *J2MEWTK_HOME* (say, `C:\J2mewtk`).

Download the Sun Wireless toolkit at:
<http://java.sun.com/products/j2mewtoolkit/download.html>. Install the toolkit. During installation you will be prompted to select a JDK. You can select the JDK included in your *BEA_HOME*/jdk131 directory.

4. Set `J2MEWTK_HOME=C:\J2mewtk`, assuming `C:\J2mewtk` is where you installed J2ME. Note: if you do not set `J2MEWTK_HOME`, you will not be able to build the application.
5. Set your environment by running the `setExamplesEnv` script located in `WL_HOME\samples\server\config\examples`.
6. To *SMARTICKET_HOME*\smarticket\localant.bat, add “%CLASSPATH%” to the end of the `ANT_CLASSPATH` line.

7. Change to the *SMARTICKET_HOME*\smarticket directory and build Smart Ticket by running `localant.bat`. WebLogic Builder requires compiled `.class` files and cannot use `.java` files.

Converting and Tuning with WebLogic Builder

In this section, WebLogic Builder generates deployment descriptors, and we edit some of them without having to look at any XML.

Generating Descriptors

In this sequence, WebLogic Builder reads the existing descriptor files and introspects the application's `.class` files to create the deployment descriptor files that help an application to run on WebLogic Server.

Note that WebLogic Builder will not overwrite the application's existing deployment descriptor files.

1. Open WebLogic Builder from Start | Programs | BEA WebLogic Platform | WebLogic Server 7.0 | WebLogic Builder.
2. In WebLogic Builder's File | Open menu, navigate to *SMARTICKET_HOME* \build\server and click Open. A dialog asks: "Unable to locate deployment descriptors. Would you like deployment descriptors created for you?" Click Yes, and WebLogic Builder will introspect the Smart Ticket class files and generate `weblogic.xml` and `weblogic-ejb-jar.xml`.
3. Select File | Save and save the application in WebLogic Builder.
4. In WebLogic Builder, create `smarticket.ear` by selecting File | Archive and specifying *SMARTICKET_HOME* \bin\smarticket.ear.
5. Start the Examples server from: Start | Programs | BEA WebLogic Platform | WebLogic Server 7.0 | Server Tour and Examples | Launch Examples Server. The Examples Server launches a WebLogic Server Examples page.
6. Open the WebLogic Server Administration Console by navigating to <http://localhost:7001/console> (or by following the link from the WebLogic Server Examples page), and sign in using username `weblogic` and password `weblogic`.

Specifying <context-root>

In this step, we set the web application's context path using the `<context-root>` element.

1. In WebLogic Builder, select the \web node's Context Path tab.
 2. In the Context Path text field, enter SmartTicketApp.
- Now the application's <context-root> element is specified.

Specifying JNDI Names

In this sequence, we:

- Assign JNDI names to the web application's EJB references and resource references
- Assign JNDI names to the EJB resource references

Select Builder's EJB Refs panel in the J2EE Refs node, and specify the JNDI names for the EJB Refs as follows:

Reference Name	EJB Type	JNDI Name
ejb/MovieInfo	Session	MovieInfo
ejb/TicketSales	Entity	TicketSales
ejb/Customer	Entity	Customer
ejb/LocaleInfo	Session	LocaleInfo

Select the Resource Refs panel in Builder's J2EE Refs node and set the Ref Name, EJB Type, and JNDI Name as in the table below. Set the Resource Authentication Type to Container.

Reference Name	Reference Type	JNDI Name
jdbc/MovieInfoDataSource	java.sql.DataSource	MovieInfoDataSource
jdbc/TicketSalesDataSource	java.sql.DataSource	TicketSalesDataSource
jdbc/CustomerDataSource	java.sql.DataSource	CustomerDataSource
jdbc/LocaleInfoDataSource	java.sql.DataSource	LocaleInfoDataSource

Select the Resource References panel from Builder's EJB Resources node and set JNDI names for EJB Resources as in the table below.

Reference	Reference Name	Resource Reference Type	JNDI Name
jdbc/MovieInfoDataSource		java.sql.DataSource	MovieInfoDataSource
jdbc/TicketSalesDataSource		java.sql.DataSource	TicketSalesDataSource
jdbc/CustomerDataSource		java.sql.DataSource	CustomerDataSource
jdbc/LocaleInfoDataSource		java.sql.DataSource	LocaleInfoDataSource

Use Builder's File | Save to save changes to the archive.

Administration Tasks

In this section we :

- Use the WebLogic Server Administration Console to configure the data sources
- Convert to the Pointbase example RDBMS.

Configuring Data Sources

Now we use the WebLogic Server Administration Console to configure the Data Sources for each of the four EJBs the Smart Ticket application uses.

1. If you haven't already started the WebLogic Server Examples Server and opened the Administration Console by pointing a browser to <http://localhost:7001/console>, do so now.
2. Select the JDBC node and click Tx Data Sources.
3. Select Configure a new JDBC Tx Data Source.
4. Enter a name for the data source. The first one is MyCustomerDataSource. Enter CustomerDataSource in the JNDI field. Enter demoPool as your Pool Name; this is the default connection pool that WebLogic Server examples use. Click Create.
5. Click the Targets Tab, then select the examplesServer in the Available column and click on the right arrow to target it. Click Apply.

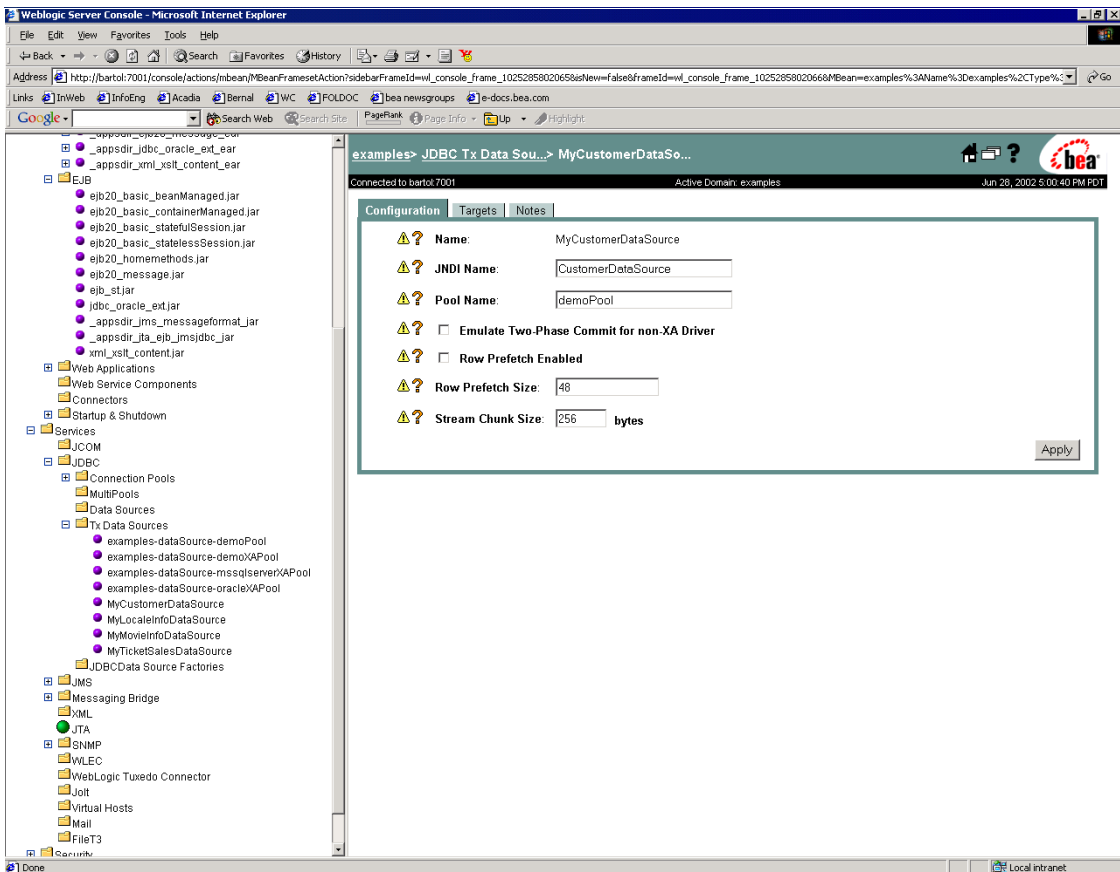


Figure 2-1 Setting up the Tx Data Sources in the Administration Console

6. Repeat steps 4 and 5 for the other four data sources (MyMovieInfoDataSource, MyLocaleInfoDataSource, and MyTicketSalesDataSource).
7. To replace Smart Ticket's Cloudscape database with Pointbase, the evaluation RDBMS included with WebLogic Server 7.0, just add the following to **SMARTTICKET_HOME** \smartticket\populate.bat:

```
set POINTBASEHOME=%SAMPLES_HOME%\server\eval\pointbase
java utils.Schema
jdbc:pointbase:server://localhost/demo,database.home=%POINTBASEHOME%
```

```
com.pointbase.jdbc.jdbcUniversalDriver -u examples -p examples -verbose  
./src/smartticketPointBase.sql
```

8. We are also making the `smartticket.sql` Pointbase friendly by replacing it with a script that substitutes “int” data types with “integer.” Copy `smartticketPointBase.sql` from into `SMARTICKET_HOME \smartticket\src`.
9. Set up the database by running `populate.bat`.

Deploying and Running

Now we are ready to deploy and run Smart Ticket on WebLogic Server 7.0.

1. Connect to the server using the Connect to Server dialog in WebLogic Builder’s Tools menu.

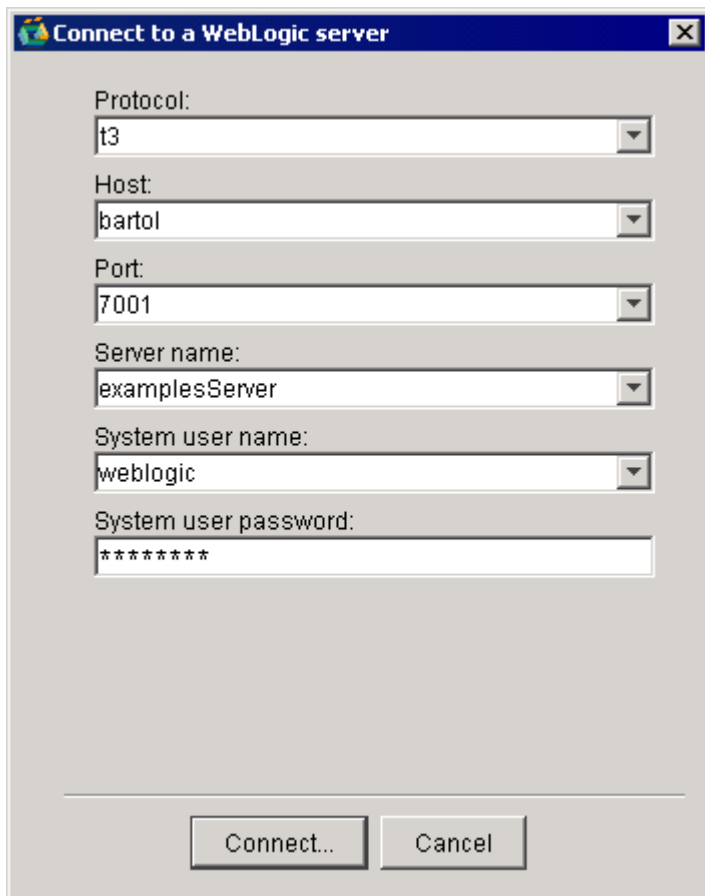
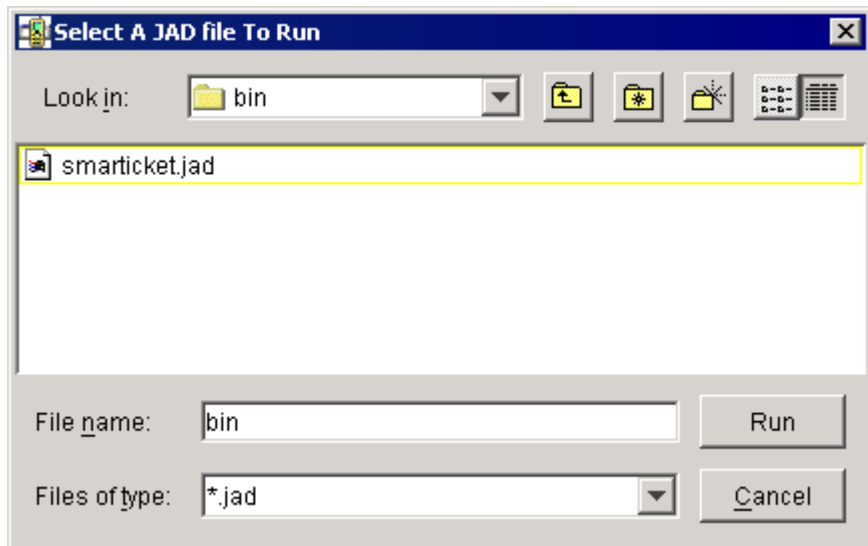


Figure 2-2 Connecting to the Examples Server

2. Deploy Smart Ticket by selecting Deploy Module from the WebLogic Builder tools menu.
3. Set the port in `SMARTTICKET_HOME\bin\smarticket.jad` by replacing the given port number (8000) with the port number of the Examples server, which by default is 7001.
4. Start `SMARTTICKET_HOME\bin\smarticket.jad` by double-clicking it or by selecting Start | J2ME Wireless Application | Run MIDP Application and selecting it:



5. You are running the Smart Ticket application on the Examples Server.



Running the Wireless Application

When creating a user account for the Smart Ticket Application, enter in 95130 or 95054 for your zip code; also, your password must be 6 characters long. Try selecting the 'Poster' mode when you create an account; this will enable your phone or emulator to view a picture of the movie you choose to see.

There are several options available for using the emulator. Go into your J2ME Wireless Ticket and manipulate the default device and preferences.

If you have run the client Smart Ticket Application before and are redeploying the application on WebLogic, you will have to clear the database of the previous users information. To do so, run the J2ME Wireless Ticket Utility which is available from the Windows start menu. Click on 'Clean Database'. Then restart the Smart Ticket Client.

Summary

We downloaded the Sun application, built it, and used WebLogic Builder to generate the weblogic.xml and weblogic-ejb-jar.xml files required to run it on WebLogic Server. We then edited the application's elements slightly using WebLogic Builder, and used the WebLogic Server Administration Console to configure the application's data sources. We used WebLogic Builder to deploy the application, which we then launched and ran.

Related Information

See the procedure for porting Smart Ticket to WebLogic Server 7.0 without using WebLogic Builder: [Java Smart Ticket Demo 1.1](http://edocs.bea.com/wls/docs70/quickstart/smartticket.html) at <http://edocs.bea.com/wls/docs70/quickstart/smartticket.html>.

