

WebLogic Server 11gR1 Foundation Lab

Introduction

The following hands-on labs are intended to provide an introduction to a number of core WebLogic Server 11gR1 administration tasks. These labs are intended to give you practice in configuring WLS, through the Admin Console and online WLST scripts, and also to practice deploying applications to WebLogic Server in a variety of Development and Production Mode scenarios using WLST, admin console and weblogic.Deployer.

There are 13 labs in all. These cover:

1. Domain Creation with the Configuration Wizard
2. Managing servers with WLST
3. Creating domain templates with Domain Template Builder
4. Creating a domain from a custom template with WLST offline
5. Using Pack/Unpack to distribute domain configuration
6. Managing Security Data with WLST Online
7. Fastswap (Development Mode)
8. Autodeploy (Development Mode)
9. Side-by-side Deployment (Production Mode)
10. Admin Mode Deployment (Production Mode)
11. In-place Partial Redeployment (Production Mode)
12. Creating a Deployment Plan with the Admin Console
13. Starting WebLogic Server as a Windows Service

The lab materials are included in a zipped archive file. You can unzip this and store it anywhere on your local machine, this directory will be referred as [%LAB_HOME%](#). In it, you will find a folder (Apps) which contains a number of simple JEE web applications that you will use for the lab exercises, as well [wlnav.war](#), a useful utility tool for browsing the WebLogic Server MBean trees. There is also a folder (Scripts) which contains a number of WLST and Windows command scripts used in some of the labs.

For labs 1-5, you will create and work with two domains ([FoundationA](#) and [FoundationB](#)). For labs 6-13, you can use a simple WebLogic Server domain ([FoundationLab](#)) with only a single AdminServer. You will be instructed about how to create these domains as you work through the labs.

Lab 1 - Domain Creation with the Configuration Wizard

The configuration wizard provides an easy way to create a custom WebLogic Server domain. In this lab, we will use it to create a basic domain with one admin server and two managed servers. You will find a shortcut in the Windows Start menu program group “Oracle Weblogic” under [Oracle WebLogic Server 11gR1 - > Tools -> Configuration Wizard](#). You can also find the executable ([config.exe/sh](#)) in the `%MIDDLEWARE_HOME%\wlserver_10.3\common\bin`.

Start the configuration wizard

- Choose the option to “[Create a new WebLogic Server domain](#)”, click [Next](#)
- Choose the option to “[Generate a domain configured automatically to support the following products](#)”, do not select any additional products, click [Next](#)
- Set domain name as [FoundationA](#), keep the default domain location, click [Next](#)
- Keep the default admin user [weblogic](#) and enter password [weblogic1](#) twice, click [Next](#)
- Choose [Development Mode](#) and select the [JRockit SDK](#), click [Next](#)
- Check [Administration Server](#) and [Managed Servers, Clusters and Machines](#) checkboxes, click [Next](#)
- Configure the Administration Server:
 - Name: [AdminServer](#)
 - Listen Address: [All Local Addresses](#)
 - Listen Port: [7001](#)
 - SSL Listen: [7002](#)
 - SSL Enabled: [checked](#)
- Configure 2 Managed Servers
 - [ManagedServer_1](#) (All local addresses; 7003; SSL 7004)
 - [ManagedServer_2](#) (All local addresses; 7005; SSL 7006)
- Configure Clusters: No (covered in Cluster Lab)
- Configure Machines: No (covered in Cluster Lab)
- Review domain configuration, click [Create](#)
- Click [Done](#) after domain creation (do not check [Start Admin Server](#) checkbox)

Observe the file structure of the newly created domain in the `%MIDDLEWARE_HOME%\user_projects\domains\FoundationA`

Lab 2 – Managing servers with WLST

Start a command shell window and cd to

`%MIDDLEWARE_HOME%\user_projects\domains\FoundationA`

Run `bin\setDomainEnv.cmd` to set your environment

Run WLST by issuing the following command: `java weblogic.WLST`, you should see the following command prompt: `wls:/offline>`

Start the WebLogic Node Manager on your system with the following command (replace `%MIDDLEWARE_HOME%` with your path before issuing the command):

```
wls:/offline> startNodeManager (verbose='true',  
NodeManagerHome='%MIDDLEWARE_HOME%\wlserver_10.3/common/nodemanager', ListenPort='5556')
```

You should see a message saying “Successfully launched the Node Manager.”

Start the admin server for your domain with the following command:

```
startServer (adminServerName="AdminServer", domainName="FoundationA")
```

You should see the startup output from the admin server as shown in the following screens, finishing with a message saying:

```
<Server started in RUNNING mode>
```

Connect to the admin server using the following command:

```
wls:/offline> connect("weblogic","weblogic1","t3://localhost:7001")
```

You should see a message saying “Successfully connected to Admin Server...”

Start ManagerServer_1 with the following command:

```
wls:/FoundationA/serverConfig> start("ManagedServer_1","Server")
```

You should see a message saying “Server with name ManagedServer_1 started.successfully”

Start Managed Server_2 with the following command:

```
wls:/FoundationA/serverConfig> start("ManagedServer_2","Server")
```

You should see a message saying “Server with name ManagedServer_1 started.successfully”

Deploy the `wlnav.war` utility (located in `%LAB_HOME%/apps/WLNav/lwnav.war`) with the following command (replace `c:\Labs` with the path on your machine):

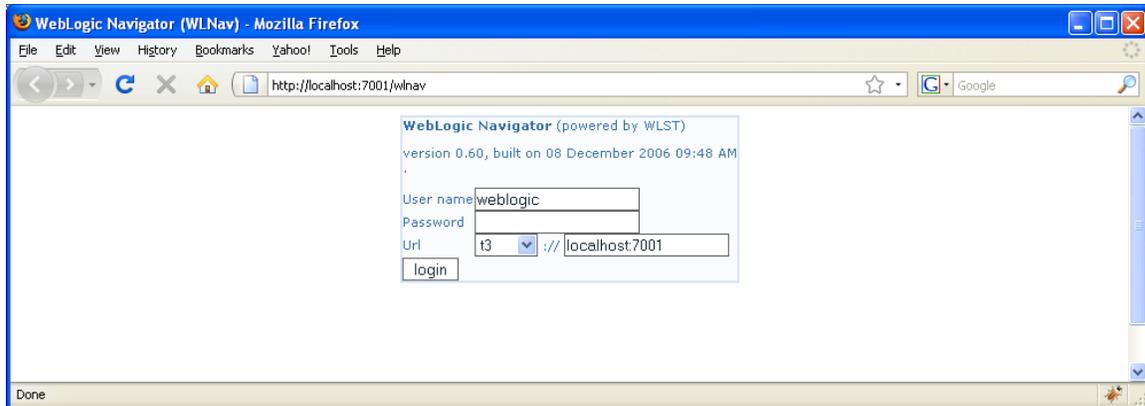
```
wls:/FoundationA/serverConfig> deploy("wlnav","c:\Labs\FoundationLab\apps\WLNav\wlnav.war",targets="AdminServer")
```

You should see a message “Completed deployment of Application with status Completed”)

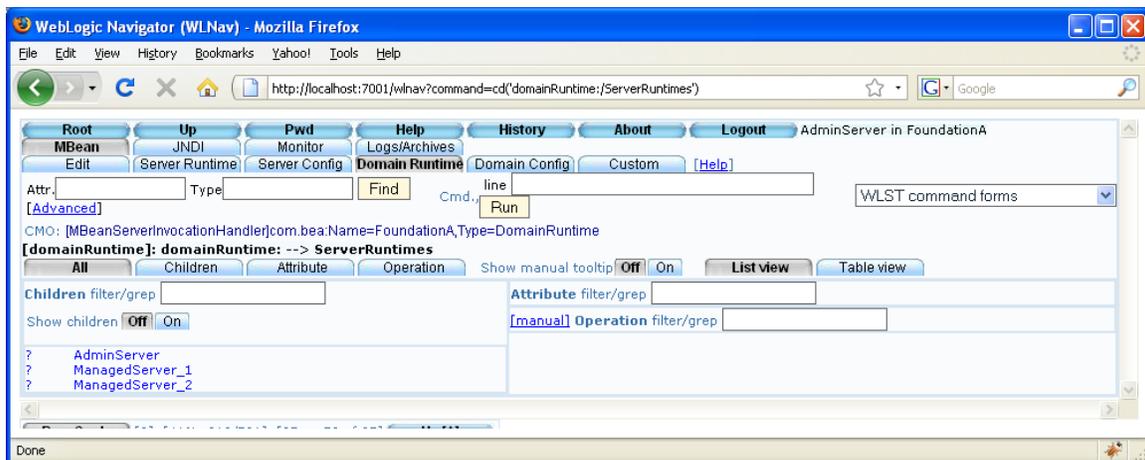
Test the `wlnav` application by opening a browser:

<http://localhost:7001/wlnav>

You should see something like this:



Note: WLNav is a very useful open-source utility originally developed by some Oracle consultants. You will find it useful for browsing the WLS MBean trees. Try logging on as weblogic/weblogic1, selecting the tab for the Domain Runtime tree and click on ServerRuntimes: you should see the three servers in FoundationA domain:



Connect to the Node Manager with the following command:

```
wls:/offline> nmConnect(username="weblogic", password="weblogic1",
domainName="FoundationA")
```

You should see a message saying “Successfully connected to Node Manager”

Check the status of ManagedServer_1 with the following command:

```
wls:/FoundationA/serverConfig> nmServerStatus("ManagedServer_1")
```

You should see the message “**RUNNING**”

Shutdown ManagedServer_1 with the following command:

```
wls:/FoundationA/serverConfig> shutdown(name="ManagedServer_1")
```

You should see a message “<**Server was shutdown normally**>”

Check the status of ManagedServer_1 with the following command:

```
wls:/FoundationA/serverConfig> nmServerStatus("ManagedServer_1")
```

You should see the message “SHUTDOWN”

Shutdown ManagedServer_2 with the following command:

```
wls:/FoundationA/serverConfig> shutdown(name="ManagedServer_2")
```

You should see a message “<Server was shutdown normally>”

Disconnect from Node Manager with the following command:

```
wls:/FoundationA/serverConfig> nmDisconnect()
```

You should see the message “Successfully disconnected from Node Manager”

View online help for the WLST shutdown() command:

```
wls:/FoundationA/serverConfig> help ("shutdown")
```

Shutdown the Admin Server with the following command:

```
wls:/FoundationA/serverConfig> shutdown(ignoreSessions="true")
```

You should see the message “Disconnected from weblogic server: AdminServer”

Exit WLST with the following command:

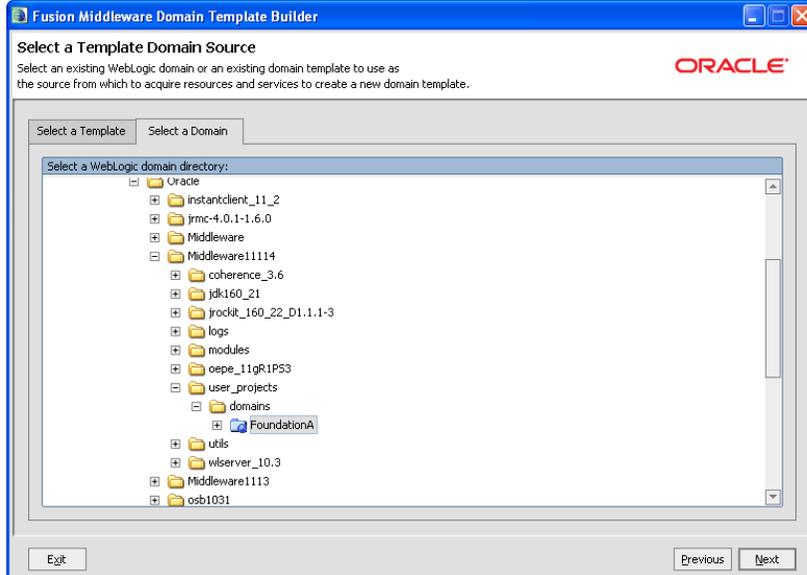
```
wls:/offline> exit()
```

Lab 3 – Creating domain templates with Domain Template Builder

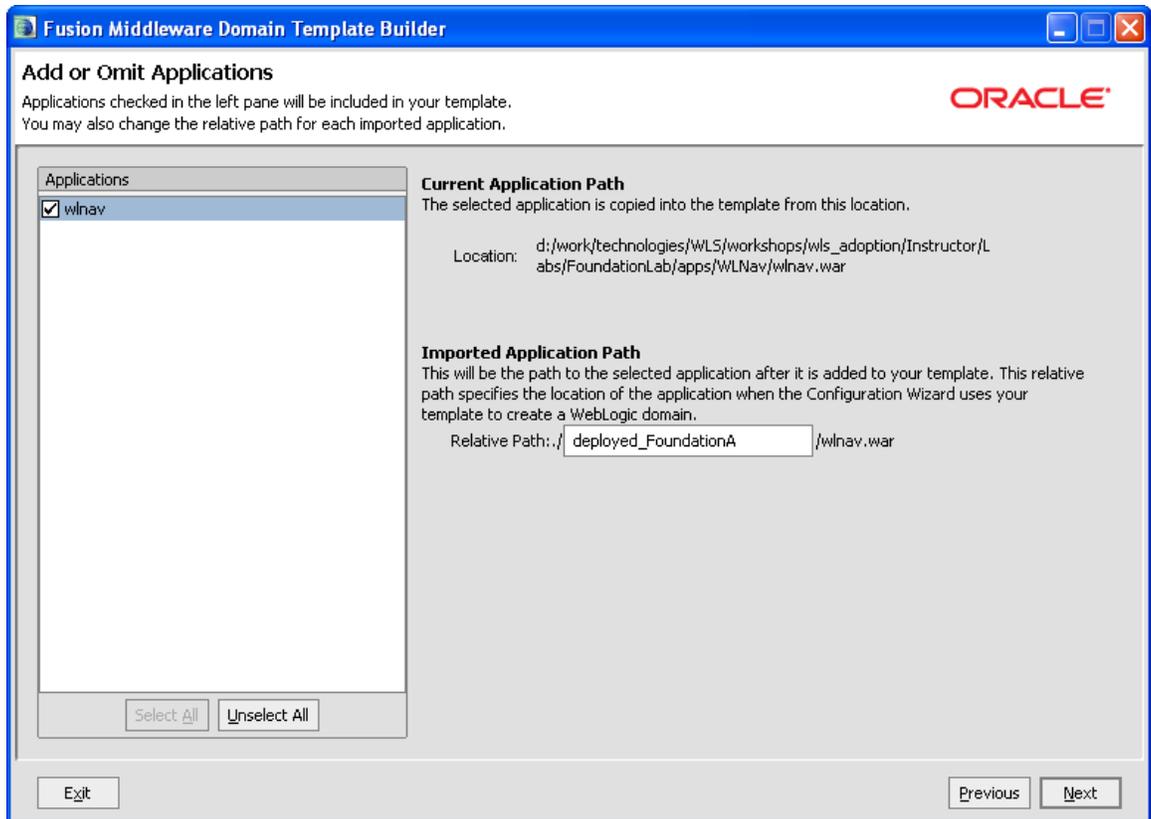
The domain template builder tool allows you to create domain templates (in the form of .jar files) that you can use to create new domains that are based on an existing one. In this lab, we will use the domain template builder to build a template based on the FoundationA domain you created in the previous lab.

You will find a shortcut in the Windows Start menu program group “Oracle Weblogic” under Oracle WebLogic Server 11gR1 -> Domain Template Builder. You can also find the executable ([builder.exe/sh](#)) in [%MIDDLEWARE_HOME%/wlserver_10.3/common/bin](#).

- Launch the Domain Template Builder from the Windows Start menu
- Choose the option “[Create a Domain Template](#)”, click [Next](#)
- Select the root directory of the [FoundationA](#) domain as a domain template source, click [Next](#)



- Enter “[FoundationA Lab Template](#)” as a template name, click [Next](#)
- Set [FoundationA](#) as a template jar name, keep default value for template location, click [Next](#)
- Make sure that the [wlnav](#) application is checked and specify an Imported Applications Path of “[./deployed_FoundationA/wlnav.war](#)”. This enables you to make sure that the wlnav.war deployment archive is available to deploy when creating new domains using the template (see the screenshot below). Click [Next](#)



- Click **Next** on the **Add Files** step
- Click **Next** on the **Add SQL Scripts** step
- Keep current settings on the **Configure Admin Server** step, click **Next**
- Keep current settings on the **Configure Administrator..** step, click **Next**
- Click **Next** on the **Specify Start Menu Items** step
- Click **Next** on the **Prepare Scripts and Files with Replacement Variables** step
- Review your template and click **Create** button.
- Press **Done** after successful template creation

Lab 4 – Creating a domain from a custom template with WLST offline

In this lab, we will use the template you just created to build an exact copy of the [FoundationA](#) domain, called [FoundationB](#). Open command line, cd to your %MIDDLEWARE_HOME% directory and run WLST in offline mode by issuing the command `.wlserver_10.3\common\bin\wlst.cmd`

Use the following commands to create the new [FoundationB](#) domain, based on the [FoundationA.jar](#) template (replace `c:\wls103` with the correct middleware home path on your machine):

```
wls:/offline> readTemplate("c:\wls103\user_templates\FoundationA.jar")
wls:/offline...>setOption("OverwriteDomain","true")
wls:/offline...> setOption("DomainName","FoundationB")
wls:/offline....>writeDomain("c:\wls103\user_projects\domains\FoundationB")
wls:/offline...>closeTemplate()
wls:/offline...>exit()
```

Check the domain has been created by opening Windows Explorer and browsing to `%MIDDLEWARE_HOME%\user_projects\domains\FoundationB`.

If you wish, start the admin server by double-clicking on `startWebLogic.cmd`.

You can view the new domain by launching WLNav: <http://localhost:7001/wlnav>.

Stop Admin Server by clicking Ctrl+C in the command window.

Lab 5 - Using Pack/Unpack to distribute domain configuration

In this lab, we will practice using the WebLogic Server *pack* and *unpack* utilities. These are convenient for transferring domain configuration data to remote hosts, in order to run remote managed servers in distributed domains. Pack allows you to create a template jar file which contains all the necessary domain configuration data; all you then need to do is to install WebLogic on the remote node and use the *unpack* command to prepare the domain and managed server information. In this short exercise, we will simply recreate the domain configuration in a temp directory to try out the *pack* and *unpack* commands.

Open a command shell, set your environment, *cd* to your temp directory and use the following command to create the **FoundationB** domain config template (note that for clarity, the command has been split across multiple lines but it should be entered as a single line), replace *c:\wls103* with the correct middleware home path on your machine:

```
C:\Temp>c:\wls103\wlserver_10.3\common\bin\pack.cmd
        -domain="c:\wls103\user_projects\domains\FoundationB"
        -template="managedFoundationB.jar"
        -template_name="FoundationB managed server template"
        -managed=true
```

You should see a message like this:

```
<< read domain from "c:\wls103\user_projects\domains\FoundationB"
>> succeed: read domain from
"c:\wls103\user_projects\domains\FoundationB"
<< set config option Managed to "true"
>> succeed: set config option Managed to "true"
<< write template to "C:\Temp\managedFoundationB.jar"
.....
>> succeed: write template to "C:\Temp\managedFoundationB.jar"
<< close template
>> succeed: close template
```

To unpack the template, use the following *unpack* command – remember that in reality you would use this on a separate server, after you had installed WebLogic Server (again, for clarity the command is shown spanning multiple lines, do not forget to replace *c:\wls103* with your OS path):

```
C:\Temp>c:\wls103\wlserver_10.3\common\bin\unpack.cmd
        -domain="c:\temp\FoundationB"
        -template="c:\temp\managedFoundationB.jar"
        -app_dir="c:\temp\apps"
```

You should see a message like this:

```
<< read template from "c:\temp\managedFoundationB.jar"
>> succeed: read template from "c:\temp\managedFoundationB.jar"
<< set config option DomainName to "FoundationB"
>> succeed: set config option DomainName to "FoundationB"
<< write Domain to "c:\temp\FoundationB"
.....
>> succeed: write Domain to "c:\temp\FoundationB"
<< close template
>> succeed: close template
```

If you wish, you can now start `ManagedServer_1` from the `c:\temp` directory, but you must first make sure that the admin server for the `FoundationB` domain is up and running, since managed servers need to connect to the domain admin server at least once to obtain their initial configuration (after that, they can start without the admin server running, in independent mode).

To start AdminServer for `FoundationB` domain run the following script:

```
%MIDDLEWARE_HOME%\user_projects\domains\FoundationB\startWebLogic.cmd
```

`cd` to `c:\temp\FoundationB\bin` dir and use the `startManagedServer_1.cmd` script to start the managed server. You will need to enter a username and password (weblogic/weblogic1) to start the server.

Go to Weblogic Administration Console (<http://localhost:7001/console>, weblogic/weblogic1), go to the `FoundationB->Environment->Servers` and check that `AdminServer` and `ManagedServer1` are running.

To stop the servers from AdminConsole, you can go to the `Control` tab, check running servers and click `Shutdown->Force Shutdown now`

The screenshot shows the Weblogic Administration Console interface. On the left is the 'Domain Structure' tree with 'FoundationB' selected. The main area is titled 'Summary of Servers' and has tabs for 'Configuration' and 'Control'. Below the tabs is a table of servers:

Server	Machine	State	Status of Last Action
<input type="checkbox"/> Server			
<input checked="" type="checkbox"/> AdminServer(admin)		RUNNING	None
<input checked="" type="checkbox"/> ManagedServer_1		RUNNING	None
<input type="checkbox"/> ManagedServer_2		SHUTDOWN	None

Buttons for 'Start', 'Resume', 'Suspend', 'Shutdown', and 'Restart SSL' are visible above the table. The 'Shutdown' dropdown menu is open, showing 'When work completes' and 'Force Shutdown Now' options.

Lab 6 - Managing Security Data with WLST Online

For the remainder of these lab exercise, we will use a simple domain called [FoundationLab](#), with the following parameters:

- only one server ([AdminServer](#))
- listen ports (7001/7002)
- user weblogic, password weblogic1
- Development mode
- JRockit JDK
- Domain location:
%MIDDLEWARE_HOME%/user_projects/domains/FoundationLab

Please create this domain now, using the Configuration Wizard.

Start [AdminServer](#) with [startWebLogic.cmd](#) script located inside domain directory.

To practice a little with WLST, let's create a user, create a group and add the user to the group. To do this, we'll use two utility classes ([UserEditorMBean](#) and [GroupEditorMBean](#)) from weblogic.jar. We will need to access the default authentication provider for the default security realm and we'll use a couple of variables ([realm](#) and [atnr](#)) to refer to these. Open a command shell, set your environment with [setDomainEnv.cmd](#) (located in the domain's bin directory), start WLST in offline mode and connect to the FoundationLab admin server running on port 7001:

```
java weblogic.WLST
wls:/offline>> connect('weblogic', 'weblogic1', 't3://localhost:7001')
```

Now enter the following commands in interactive mode to create your user and group, and add the user to the new group:

```
> from weblogic.management.security.authentication import UserEditorMBean
> realm=cmo.getSecurityConfiguration().getDefaultRealm()
> atnr=realm.lookupAuthenticationProvider("DefaultAuthenticator")
> atnr.createUser('mark','password1','New wls user')
> from weblogic.management.security.authentication import GroupEditorMBean
> atnr.createGroup('plainUsers','Ordinary People')
> atnr.addMemberToGroup('plainUsers','mark')
```

Keeping the same interactive WLST session, try iterating through the list of users registered with the default authentication provider:

```
> from weblogic.management.security.authentication import UserReaderMBean
> realm=cmo.getSecurityConfiguration().getDefaultRealm()
> atnr=realm.lookupAuthenticationProvider("DefaultAuthenticator")
> cursor = atnr.listUsers("*",0)
> groupReader=atnr
```

```
> atnr.getCurrentName(cursor)
> atnr.advance(cursor)
> atnr.getCurrentName(cursor)
> atnr.advance(cursor)
> atnr.getCurrentName(cursor)
> atnr.close(cursor)
> disconnect()
> exit()
```

Obviously, this is pretty painful and a better alternative is to create a WLST script file that we can run with a single command. There is a simple WLST script to list out all the users and groups registered with the default authentication provider in the %LAB_HOME%/Scripts folder, called [ListUsersGroups.py](#). Try running this by typing the following command (replace %LAB_HOME% with the actual path in your filesystem):

```
> java weblogic.WLST %LAB_HOME%\Scripts\ListUsersGroups.py
```

You should see a list of users and groups printed out. There is more detail (and more examples) of how to manage security data using WLST in the WebLogic Server docs:

http://download.oracle.com/docs/cd/E17904_01/web.1111/e13715/config_wls.htm#i1028177

You will find the full WebLogic Server 10gR3 MBean Reference here:

http://download.oracle.com/docs/cd/E17904_01/apirefs.1111/e13951/core/index.html

Lab 7 – Fastswap (Development Mode)

In this lab, you will explore some of the features of WebLogic Server when started in Development mode. In particular we will look at the Fastswap feature, which is WebLogic Server's implementation of the Java Hotswap specification. With Fastswap, changes to Java class files are automatically picked up and the classes reloaded without having to redeploy the application – a considerable benefit in iterative development cycles.

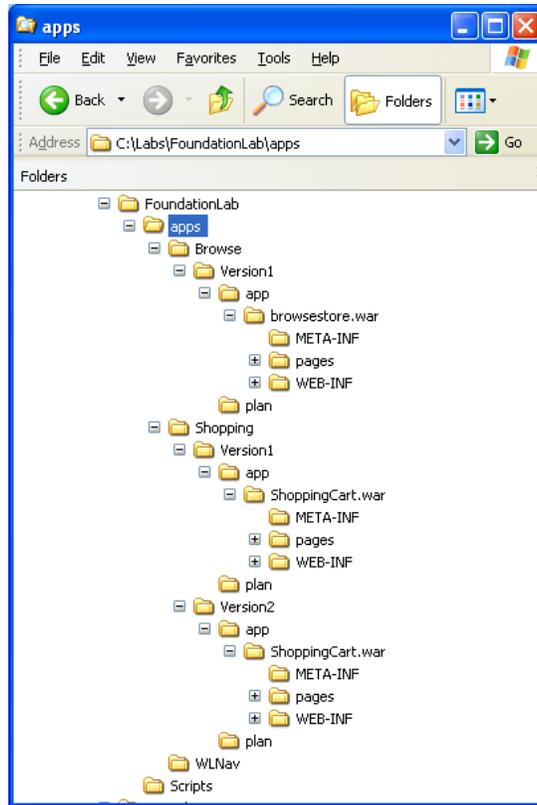
To use Fastswap, applications must be deployed in exploded archive format, and must be Fastswap-enabled via an entry in the `weblogic.xml` (for web apps) or `weblogic-application.xml` (for .ear files) deployment descriptors. We will also enable Fastswap in the domain's `setDomainEnv.cmd`: although we won't be using the feature in this lab, this lets you use Fastswap with the WebLogic Diagnostic Framework (WLDF) application instrumentation (AspectJ) support.

Edit `FoundationLab\bin\setDomainEnv.cmd` to configure Fastswap:

Search for “`set enableHotswapFlag=`” and change to:

```
set enableHotswapFlag=-javaagent:%WL_HOME%\server\lib\diagnostics-agent.jar
```

A number of sample applications have been provided for this lab, in the `%LAB_HOME%/apps` folder. Note the directory structures we will be using: these follow Oracle's best practice guidelines to simplify the use of application versions and deployment plans (we will look at these later in this lab):



There are two applications, which we will use for most of the labs:

- **Browsestore**: a basic web application with servlets and JSPs
- **ShoppingCart**: a basic “online shopping” model web application, that we will use in the Clustering Lab to test session state replication in WebLogic clusters.

Both applications are provided in exploded archive format, and both use separate folders for the deployed app and the deployment plan, while **ShoppingCart.war** has separate directory structures for different versions of the application:

We will work with the **browsestore** application. To enable the application for Fastswap, you will need to edit **WEB-INF/weblogic.xml** and add the single line stanza `<fast-swap/>`. The deployment descriptor should look like this:

```
<?xml version='1.0' encoding='UTF-8'?>
<weblogic-web-app xmlns="http://www.bea.com/ns/weblogic/90"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <fast-swap/>
</weblogic-web-app>
```

Now we are going to deploy browsestore application.

- Go to Weblogic Administration Console (<http://localhost:7001/console>)
- click on **Deployments** link in the **Domain Structure**.

- Click [Install](#)
- Select the `%LAB_HOME%\apps\Browse\Version1\app\browsestore.war` directory, click [Next](#)
- Keep [Install this deployment as an application option](#), click [Next](#)
- Keep the default settings and click [Finish](#)

Check the deployment by opening a browser to: <http://localhost:7001/browsestore>
Click [Browse Store](#) hyperlink, then check [Furniture](#) and click [Retrieve Items](#) button. Look at AdminServer's output and you will see the following messages:

```
serviced request for the welcome page
serviced the request to browse the store
serviced request to print items
```

To see Fastswap in action, edit the `browseCategories.java` servlet (you will find this in `browsestore.war/WEB-INF/classes/com/servlets`).

Replace this line

```
System.out.println("serviced request to print items");
```

with the following line:

```
System.out.println("CHANGED: serviced request to print items");
```

recompile `browseCategories.java`:

- Open command line
- Execute `setWLSEnv.cmd` script located in `%MIDDLEWARE_HOME%\wlserver_10.3\server\bin`
- Cd to `apps\Browse\Version1\app\browsestore.war\WEB-INF\classes\com\servlets` inside your lab home
- Compile the class: `javac browseCategories.java`

Reload the browsestore webapp in your browser, repeat the sequence of navigation steps. You should see the changes immediately, without needing to deploy the application:

```
serviced request for the welcome page
serviced the request to browse the store
CHANGED: serviced request to print items
```

Of course, JSPs just work normally – you can try editing one of the JSPs for the browsestore application and note how it is automatically recompiled when you reload the browser window. The big difference with fast-swap is with compiled Java classes. To reload static content, you use the admin console or the `weblogic.Deployer` command to do a partial redeploy (we will cover this in a later lab exercise). However, in Development mode it is easier and faster to use the Autodeploy feature and we will look at this next.

Lab 8 – Autodeploy (Development Mode)

First, use the Admin console to undeploy the Browsestore application:

- Go to Weblogic Administration Console (<http://localhost:7001/console>)
- click on **Deployments** link in the **Domain Structure**.
- Select browsestore application and click **Delete**

Navigate to the FoundationLab domain directory in you file system. You will see a directory called “**Autodeploy**”. Simple copy the browsestore.war exploded archive and paste it into this directory. You will see the application deployment messages in the server window and you should shortly be able to run the application: <http://localhost:7001/browsestore>

To trigger a redeploy (this is what tools like WebLogic Workshop use to support iterative development), simply create an empty file in the [autodeploy/browsestore.war/WEB-INF](#) directory, called **REDEPLOY**. Make a change to the application, then open and save the **REDEPLOY** file to update its timestamp (on Unix the **touch** command is a simple way to do this). WebLogic Server will automatically deploy the application with your changes.

Lab 9 – Side-by-side Deployment (Production Mode)

First, we need to change the server to run in Production Mode. You do this by editing `setDomainEnv.cmd` located inside domain's bin directory: look for "PRODUCTION_MODE=" at the beginning of the file and change to: `PRODUCTION_MODE=true`. You will need to restart the server for this change to take effect.

For this lab, we will use the `ShoppingCart.war` web application. You will see that two versions of the application have been provided (as exploded .war archives) in the `FoundationLab/apps` folder:

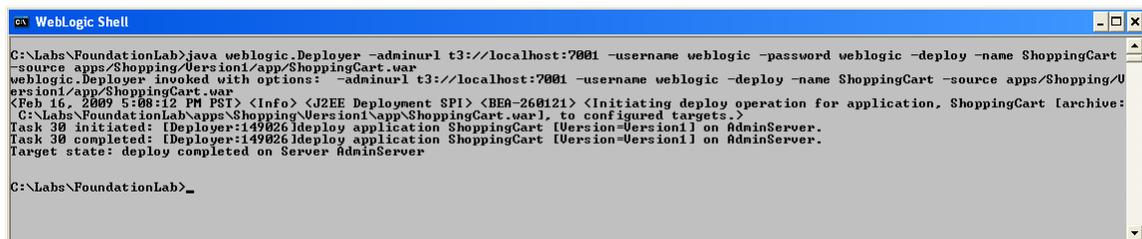
```
FoundationLab->apps->Shopping->Version1->app->ShoppingCart.war
FoundationLab->apps->Shopping->Version1->plan (unused)
FoundationLab->apps->Shopping->Version2->app->ShoppingCart.war
FoundationLab->apps->Shopping->Version2->plan (unused)
```

Note the `Weblogic-Application-Version` strings in `META-INF/MANIFEST.MF` for these two deployments. The other changes are minimal, but you should see "Version 2" in the browser window title for the v2 application.

- Open the command line
- Execute `setWLSEnv.cmd` script located in `%MIDDLEWARE_HOME%\wlserver_10.3\server\bin`
- Cd to `FoundationLab` directory

Deploy .war archive version 1 via weblogic.Deployer

```
java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic1 -deploy -name ShoppingCart -source apps/Shopping/Version1/app/ShoppingCart.war
```

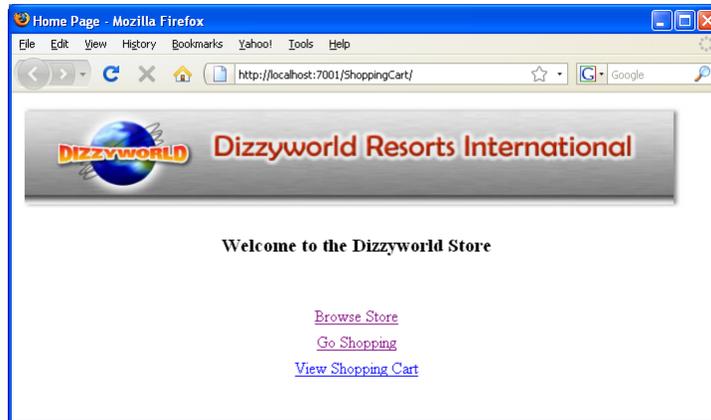


```
WebLogic Shell
C:\Labs\FoundationLab>java weblogic.Deployer -adminurl t3://localhost:7001 -username weblogic -password weblogic -deploy -name ShoppingCart
-source apps/Shopping/Version1/app/ShoppingCart.war
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -username weblogic -deploy -name ShoppingCart -source apps/Shopping/Version1/app/ShoppingCart.war
<Feb 16, 2009 5:08:12 PM PST> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating deploy operation for application, ShoppingCart [archive:
C:\Labs\FoundationLab\apps\Shopping\Version1\app\ShoppingCart.war], to configured targets.>
Task 30 initiated: [Deployer:149026] deploy application ShoppingCart [Version=Version1] on AdminServer.
Task 30 completed: [Deployer:149026] deploy application ShoppingCart [Version=Version1] on AdminServer.
Target state: deploy completed on Server AdminServer

C:\Labs\FoundationLab>_
```

Open browser sessions to ShoppingCart version 1

<http://localhost:7001/ShoppingCart>



View deployed version info via weblogic.Deployer

```
java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic1 -listapps
```

```

WebLogic Shell
C:\Labs\FoundationLab>java weblogic.Deployer -adminurl t3://localhost:7001 -username weblogic -password weblogic -deploy -name ShoppingCart
-source apps/Shopping/Version1/app/ShoppingCart.war
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -username weblogic -deploy -name ShoppingCart -source apps/Shopping/V
ersion1/app/ShoppingCart.war
<Feb 16, 2009 5:08:12 PM PST> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating deploy operation for application, ShoppingCart [archive:
C:\Labs\FoundationLab\apps\Shopping\Version1\app\ShoppingCart.war, to configured targets.>
Task 30 initiated: [Deployer:149026]deploy application ShoppingCart [Version=Version1] on AdminServer.
Task 30 completed: [Deployer:149026]deploy application ShoppingCart [Version=Version1] on AdminServer.
Target state: deploy completed on Server AdminServer

C:\Labs\FoundationLab>java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic -listapps
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -user weblogic -listapps
ShoppingCart [Version=Version1] <ACTIVE VERSION>
Number of Applications Found : 1

C:\Labs\FoundationLab>

```

Production deployment of .war archive version 2 via weblogic.Deployer

```
java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic1 -redeploy -name ShoppingCart -source apps/Shopping/Version2/app/ShoppingCart.war -retiretimeout 120
```

```

WebLogic Shell
C:\Labs\FoundationLab>java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic -listapps
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -user weblogic -listapps
ShoppingCart [Version=Version1] <ACTIVE VERSION>
Number of Applications Found : 1

C:\Labs\FoundationLab>java weblogic.Deployer -adminurl t3://localhost:7001 -username weblogic -password weblogic -redeploy -name ShoppingCart
t -source apps/Shopping/Version2/app/ShoppingCart.war -retiretimeout 60
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -username weblogic -redeploy -name ShoppingCart -source apps/Shopping
/Version2/app/ShoppingCart.war -retiretimeout 60
<Feb 16, 2009 5:10:34 PM PST> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating redeploy operation for application, ShoppingCart [archiv
e: C:\Labs\FoundationLab\apps\Shopping\Version2\app\ShoppingCart.war, to configured targets.>
Task 31 initiated: [Deployer:149026]deploy application ShoppingCart [Version=Version2] on AdminServer.
Task 31 completed: [Deployer:149026]deploy application ShoppingCart [Version=Version2] on AdminServer.
Target state: redeploy completed on Server AdminServer

C:\Labs\FoundationLab>

```

Open another browser session and access <http://localhost:7001/ShoppingCart>. You should still see Version 1 of the ShoppingCart application, as we specified a timeout of 120 seconds before retiring Version 1. Note: Version 2 will have the word "Version 2" in the browser title bar. Leave these browsers running for the time being.

View the deployed application versions using weblogic.Deployer

```
java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic1 -listapps
```

```

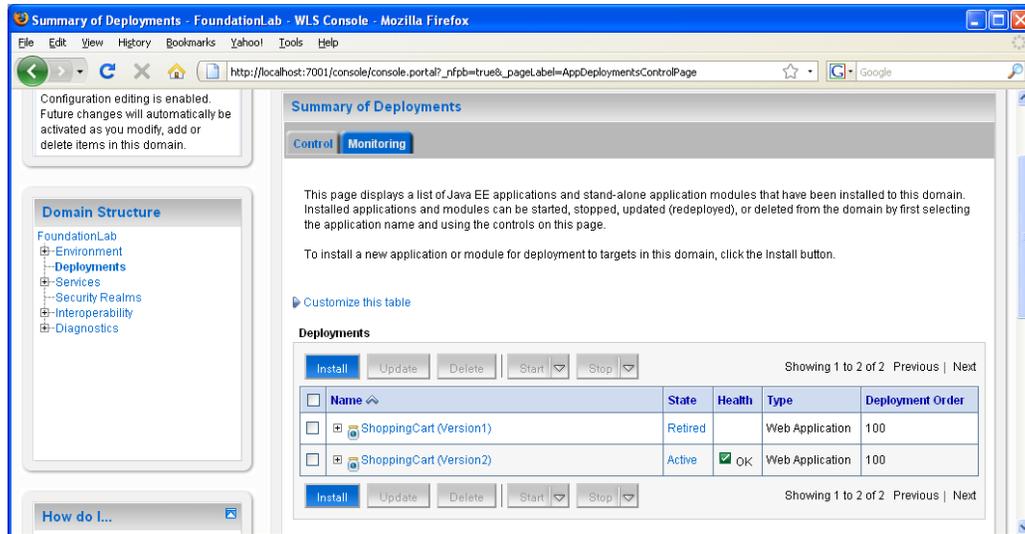
C:\WebLogic Shell
t -source apps/Shopping/Version2/app/ShoppingCart.war -retiretimeout 60
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -username weblogic -redeploy -name ShoppingCart -source apps/Shopping
/Version2/app/ShoppingCart.war -retiretimeout 60
<Feb 16, 2009 5:18:34 PM PST> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating redeploy operation for application, ShoppingCart [archiv
e: C:\Labs\FoundationLab\apps\Shopping\Version2\app\ShoppingCart.war], to configured targets.>
Task 31 initiated: [Deployer:149026]deploy application ShoppingCart [Version=Version2] on AdminServer.
Task 31 completed: [Deployer:149026]deploy application ShoppingCart [Version=Version2] on AdminServer.
Target state: redeploy completed on Server AdminServer

C:\Labs\FoundationLab>java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic -listapps
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -user weblogic -listapps
ShoppingCart [Version=Version2] <ACTIVE VERSION>
ShoppingCart [Version=Version1] <ACTIVE VERSION>
Number of Applications Found : 2

C:\Labs\FoundationLab>

```

View the deployed application versions using the Admin Console:



After 120 seconds, open another browser: <http://localhost:7001/ShoppingCart>
 This time you should see the Version 2 application (look at the browser title bar):



Welcome to the New, Improved Dizzyworld Store

- [Browse Store](#)
- [Go Shopping](#)
- [View Shopping Cart](#)

View the deployed application versions using weblogic.Deployer
`java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic1 -listapps`

```

WebLogic Shell
Task 31 completed: [Deployer:149026]deploy application ShoppingCart [Version=Version2] on AdminServer.
Target state: redeploy completed on Server AdminServer

C:\Labs\FoundationLab>java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic -listapps
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -user weblogic -listapps
  ShoppingCart [Version=Version2] <ACTIVE VERSION>
  ShoppingCart [Version=Version1] <ACTIVE VERSION>
Number of Applications Found : 2

C:\Labs\FoundationLab>java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic -listapps
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -user weblogic -listapps
  ShoppingCart [Version=Version2] <ACTIVE VERSION>
  ShoppingCart [Version=Version1]
Number of Applications Found : 2

C:\Labs\FoundationLab>_

```

Rollback to version 1 via weblogic.Deployer
 java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic1 -redeploy -name ShoppingCart -source apps/Shopping/Version1/app/ShoppingCart.war

```

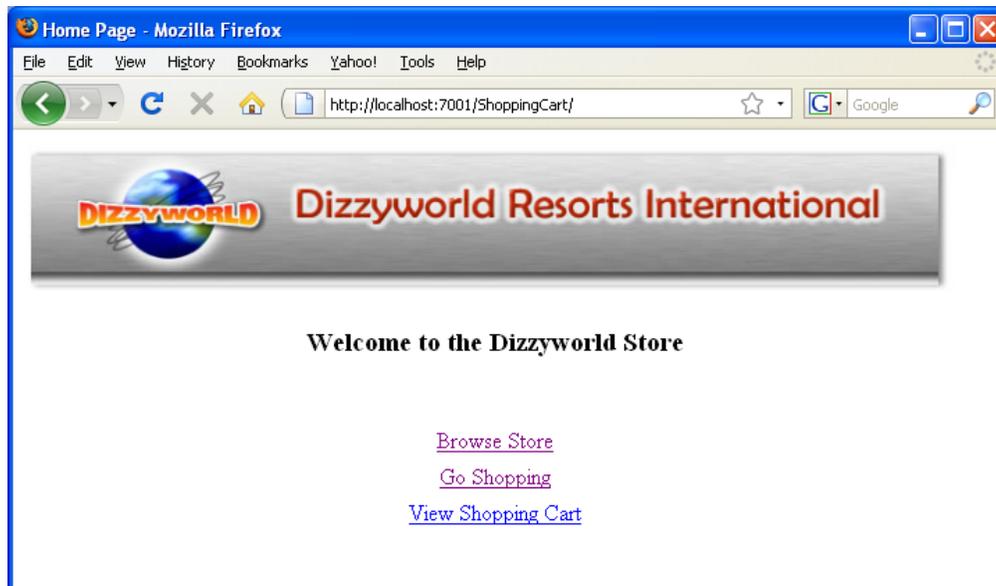
WebLogic Shell
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -user weblogic -listapps
  ShoppingCart [Version=Version2] <ACTIVE VERSION>
  ShoppingCart [Version=Version1]
Number of Applications Found : 2

C:\Labs\FoundationLab>java weblogic.Deployer -adminurl t3://localhost:7001 -username weblogic -password weblogic -redeploy -name ShoppingCart
 -source apps/Shopping/Version1/app/ShoppingCart.war
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -username weblogic -redeploy -name ShoppingCart -source apps/Shopping
/Version1/app/ShoppingCart.war
<Feb 16, 2009 5:16:22 PM PST> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating redeploy operation for application, ShoppingCart [archiv
e: C:\Labs\FoundationLab\apps\Shopping\Version1\app\ShoppingCart.war], to configured targets.>
Task 32 initiated: [Deployer:149026]deploy application ShoppingCart [Version=Version1] on AdminServer.
Task 32 completed: [Deployer:149026]deploy application ShoppingCart [Version=Version1] on AdminServer.
Target state: redeploy completed on Server AdminServer

C:\Labs\FoundationLab>

```

Open a new browser – now accessing version 1



Undeploy Version 2 of the application
 java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic1 -undeploy -name ShoppingCart -appversion Version2

```
WebLogic Shell
C:\Labs\FoundationLab>java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic -listapps
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -user weblogic -password weblogic -listapps
ShoppingCart [Version=Version2] <ACTIVE VERSION>
ShoppingCart [Version=Version1] <ACTIVE VERSION>
Number of Applications Found : 2

C:\Labs\FoundationLab>java weblogic.Deployer -adminurl t3://localhost:7001 -username weblogic -password weblogic -undeploy -name ShoppingCart
 -appversion Version2
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -username weblogic -password weblogic -undeploy -name ShoppingCart -appversion Version2
<Feb 16, 2009 5:18:10 PM PST> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating undeploy operation for application, ShoppingCart [archive:
 null], to configured targets.>
Task 33 initiated: [Deployer:149026]remove application ShoppingCart [Version=Version2] on AdminServer.
Task 33 completed: [Deployer:149026]remove application ShoppingCart [Version=Version2] on AdminServer.
Target state: undeploy completed on Server AdminServer

C:\Labs\FoundationLab>
```

View the deployed application versions using weblogic.Deployer
java weblogic.Deployer -adminurl <t3://localhost:7001> -user weblogic -password weblogic1 -listapps

```
WebLogic Shell
C:\Labs\FoundationLab>java weblogic.Deployer -adminurl t3://localhost:7001 -username weblogic -password weblogic -undeploy -name ShoppingCart
 -appversion Version2
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -username weblogic -password weblogic -undeploy -name ShoppingCart -appversion Version2
<Feb 16, 2009 5:18:10 PM PST> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating undeploy operation for application, ShoppingCart [archive:
 null], to configured targets.>
Task 33 initiated: [Deployer:149026]remove application ShoppingCart [Version=Version2] on AdminServer.
Task 33 completed: [Deployer:149026]remove application ShoppingCart [Version=Version2] on AdminServer.
Target state: undeploy completed on Server AdminServer

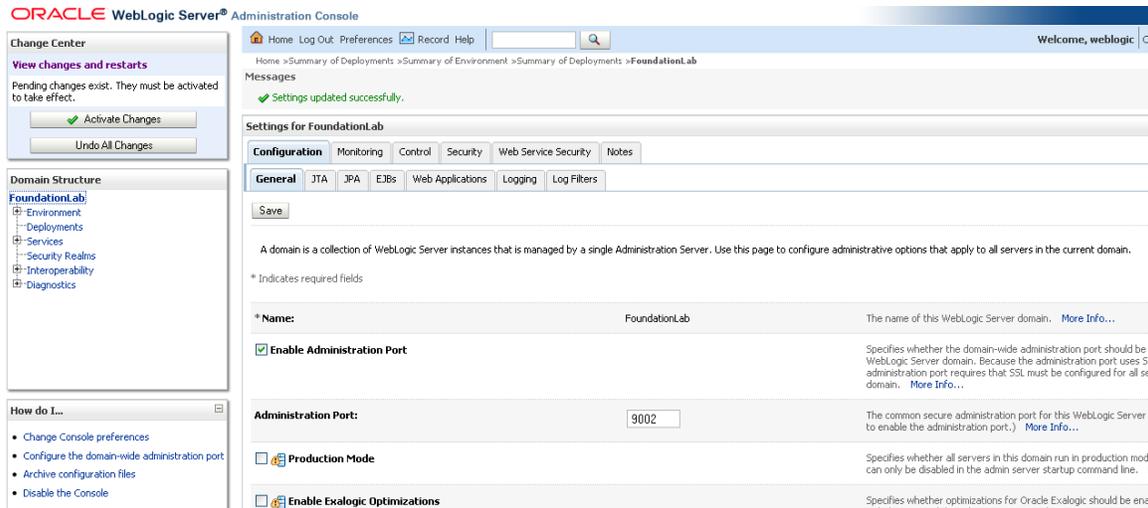
C:\Labs\FoundationLab>java weblogic.Deployer -adminurl t3://localhost:7001 -user weblogic -password weblogic -listapps
weblogic.Deployer invoked with options: -adminurl t3://localhost:7001 -user weblogic -password weblogic -listapps
ShoppingCart [Version=Version1] <ACTIVE VERSION>
Number of Applications Found : 1

C:\Labs\FoundationLab>
```

Lab 10 – Admin Mode Deployment (Production Mode)

Navigate to the [FoundationLab \(domain\)](#) -> [Configuration](#) -> [General](#) page using the admin console:

- Click [Lock&Edit](#) button
- check the '[Enable Administration Port](#)' box
- Keep the default admin Port (9002), click [Save](#)
- Click [Activate Changes](#)



The screenshot shows the Oracle WebLogic Server Administration Console interface. The main content area displays the 'Settings for FoundationLab' page, specifically the 'General' tab. The 'Enable Administration Port' checkbox is checked, and the 'Administration Port' is set to 9002. The 'Production Mode' checkbox is unchecked. The 'Change Center' on the left indicates that pending changes exist and must be activated to take effect. The 'Domain Structure' on the left shows the hierarchy of the domain.

If you prefer, you can also use the following WLST command sequence to enable the administration port:

```
>connect('weblogic','weblogic1','t3://localhost:7001')
>edit()
>startEdit()
>cd('/')
>cmo.setAdministrationPort(9002)
>cmo.setAdministrationPortEnabled(true)
>activate()
>disconnect()
>exit()
```

Once you have the admin channel enabled, you need to reconnect to the admin console using the secure admin port: <https://localhost:9002/console> and to switch to [t3s://localhost:9002](https://localhost:9002/console) for [weblogic.Deployer](#) commands. You will also need to run the Deployer command with `-Dweblogic.security.TrustKeyStore=DemoTrust` to specify the trusted CA configuration for the client (assuming you are using the demo trust keystore, which is the default for WebLogic Server).

Open the command line, Execute `setWLSEnv.cmd` script located in `%MIDDLEWARE_HOME%\wlserver_10.3\server\bin` and Cd to FoundationLab directory

Distribute version 2 for testing via weblogic.Deployer

`java -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.Deployer -adminurl t3s://localhost:9002 -user weblogic -password weblogic1 -distribute -name ShoppingCart -source apps/Shopping/Version2/app/ShoppingCart.war`

```
WebLogic Shell
weblogic.Deployer invoked with options: -adminurl t3s://localhost:9002 -username weblogic -listapps
ShoppingCart [Version=Version1] <ACTIVE VERSION>
Number of Applications Found : 1
C:\Labs\FoundationLab>java -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.Deployer -adminurl t3s://localhost:9002 -username weblogic -
password weblogic -distribute -name ShoppingCart -source apps/Shopping/Version2/app/ShoppingCart.war
weblogic.Deployer invoked with options: -adminurl t3s://localhost:9002 -username weblogic -distribute -name ShoppingCart -source apps/Shopp
ing/Version2/app/ShoppingCart.war
<Feb 16, 2009 8:07:12 PM PST> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating distribute operation for application, ShoppingCart [archi
ve: C:\Labs\FoundationLab\apps\Shopping\Version2\app\ShoppingCart.war], to configured targets.>
Task 1 initiated: [Deployer:149026]distribute application ShoppingCart [Version=Version2] on AdminServer.
Task 1 completed: [Deployer:149026]distribute application ShoppingCart [Version=Version2] on AdminServer.
Target state: distribute completed on Server AdminServer
C:\Labs\FoundationLab>
```

```
WebLogic Shell
ing/Version2/app/ShoppingCart.war
<Feb 16, 2009 8:07:12 PM PST> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating distribute operation for application, ShoppingCart [archi
ve: C:\Labs\FoundationLab\apps\Shopping\Version2\app\ShoppingCart.war], to configured targets.>
Task 1 initiated: [Deployer:149026]distribute application ShoppingCart [Version=Version2] on AdminServer.
Task 1 completed: [Deployer:149026]distribute application ShoppingCart [Version=Version2] on AdminServer.
Target state: distribute completed on Server AdminServer
C:\Labs\FoundationLab>java -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.Deployer -adminurl t3s://localhost:9002 -username weblogic -
password weblogic -listapps
weblogic.Deployer invoked with options: -adminurl t3s://localhost:9002 -username weblogic -listapps
ShoppingCart [Version=Version2]
ShoppingCart [Version=Version1] <ACTIVE VERSION>
Number of Applications Found : 2
C:\Labs\FoundationLab>
```

Start Version 2 in admin mode

`java -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.Deployer -adminurl t3s://localhost:9002 -user weblogic -password weblogic1 -start -adminmode -name ShoppingCart -appversion Version2`

```
WebLogic Shell
ShoppingCart [Version=Version2]
ShoppingCart [Version=Version1] <ACTIVE VERSION>
Number of Applications Found : 2
C:\Labs\FoundationLab>java -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.Deployer -adminurl t3s://localhost:9002 -username weblogic -
password weblogic -start -adminmode -name ShoppingCart -appversion Version2
weblogic.Deployer invoked with options: -adminurl t3s://localhost:9002 -username weblogic -start -adminmode -name ShoppingCart -appversion
Version2
<Feb 16, 2009 8:10:01 PM PST> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating start operation for application, ShoppingCart [archive:
null], to configured targets.>
Task 2 initiated: [Deployer:149026]start application ShoppingCart [Version=Version2] on AdminServer.
Task 2 completed: [Deployer:149026]start application ShoppingCart [Version=Version2] on AdminServer.
Target state: start completed on Server AdminServer
C:\Labs\FoundationLab>
```

View deployed version info via weblogic.Deployer

`java -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.Deployer -adminurl t3s://localhost:9002 -user weblogic -password weblogic1 -listapps`

```
WebLogic Shell
Version2
<Feb 16, 2009 8:10:01 PM PST> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating start operation for application, ShoppingCart [archive:
null], to configured targets.>
Task 2 initiated: [Deployer:149026]start application ShoppingCart [Version=Version2] on AdminServer.
Task 2 completed: [Deployer:149026]start application ShoppingCart [Version=Version2] on AdminServer.
Target state: start completed on Server AdminServer
C:\Labs\FoundationLab>java -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.Deployer -adminurl t3s://localhost:9002 -username weblogic -
password weblogic -listapps
weblogic.Deployer invoked with options: -adminurl t3s://localhost:9002 -username weblogic -listapps
ShoppingCart [Version=Version2]
ShoppingCart [Version=Version1] <ACTIVE VERSION>
Number of Applications Found : 2
C:\Labs\FoundationLab>
```

Name	State	Health	Type	Deployment Order
ShoppingCart (Version1)	Active	OK	Web Application	100
ShoppingCart (Version2)	Admin	OK	Web Application	100

Open browser and access via admin channel

<https://localhost:9002/ShoppingCart/>

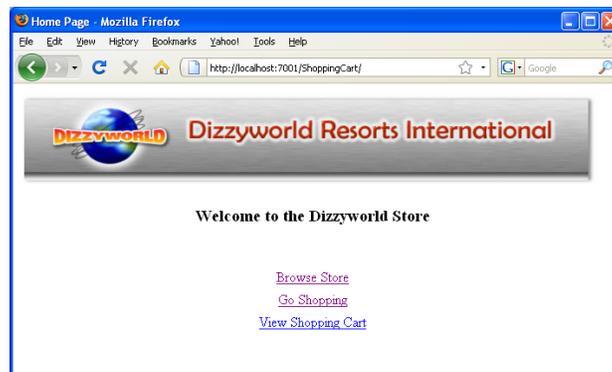
Note that the secure administration channel will require you to authenticate as weblogic/weblogic1. Log on and you should see Version 2 of the application:



Close the admin channel browser. Open another browser and access the webapp via the normal public HTTP port

<http://localhost:7001/ShoppingCart/>

You should see Version 1 of the application:



Close this browser session.

Make version 2 available to clients via weblogic.Deployer

```
java -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.Deployer -adminurl t3s://localhost:9002 -user weblogic -password weblogic1 -start -name ShoppingCart -appversion Version2 -retiretimeout 30
```

```

WebLogic Shell
weblogic.Deployer invoked with options: -adminurl t3s://localhost:9002 -username weblogic -listapps
ShoppingCart [Version=Version2]
ShoppingCart [Version=Version1] <ACTIVE VERSION>
Number of Applications Found : 2

C:\Labs\FoundationLab>java -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.Deployer -adminurl t3s://localhost:9002 -username weblogic -
password weblogic -start -name ShoppingCart -appversion Version2
weblogic.Deployer invoked with options: -adminurl t3s://localhost:9002 -username weblogic -start -name ShoppingCart -appversion Version2
<Feb 16, 2009 8:23:00 PM PST> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating start operation for application. ShoppingCart [archive:
null], to configured targets.>
Task 3 initiated: [Deployer:149026]start application ShoppingCart [Version=Version2] on AdminServer.
Task 3 completed: [Deployer:149026]start application ShoppingCart [Version=Version2] on AdminServer.
Target state: start completed on Server AdminServer

C:\Labs\FoundationLab>

```

Wait another 30 seconds, open another browser and access the webapp via the normal public HTTP port

<http://localhost:7001/ShoppingCart/>

You should see Version 2 of the application:



Welcome to the New, Improved Dizzyworld Store

[Browse Store](#)

[Go Shopping](#)

[View Shopping Cart](#)

View deployed version info via weblogic.Deployer and the Admin Console

java -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.Deployer -adminurl t3s://localhost:9002 -user weblogic -password weblogic1 -listapps

Name	State	Health	Type	Deployment Order
ShoppingCart (Version1)	Retired		Web Application	100
ShoppingCart (Version2)	Active	OK	Web Application	100

Lab 11 – In-place Partial Redeployment (Production Mode)

Next we will do a partial redeployment of one of the application JSPs. Edit the Version 2 file [apps/Shopping/Version2/app/ShoppingCart.war/welcome.jsp](#):

Replace the line:

```
<tr><td><CENTER><b><h3>Welcome to the Dizzyworld  
Store</h3></b></CENTER></td>
```

with the following:

```
<tr><td><CENTER><b><h3>Welcome to the New, Improved Dizzyworld  
Store</h3></b></CENTER></td>
```

Perform partial redeployment of version 2 JSP

```
java -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.Deployer -adminurl  
t3s://localhost:9002 -user weblogic -password weblogic1 -redeploy -name  
ShoppingCart welcome.jsp
```

Open a browser window and access the webapp via the normal public HTTP port <http://localhost:7001/ShoppingCart/>

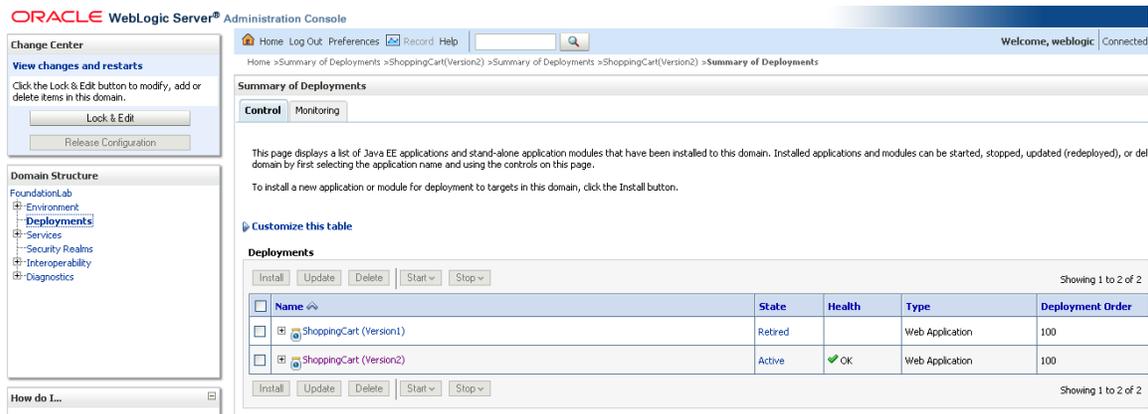
You should see Version2 of the application with the updates to the JSP available immediately. In-place redeployment works for other static content (e.g HTML and graphics), as well as JSPs. You can also use wildcards, such as:

```
java -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.Deployer -adminurl  
t3s://localhost:9002 -user weblogic -password weblogic1 -redeploy -name  
ShoppingCart *.jsp
```

For more information on in-place redeployment for production environments, see: http://download.oracle.com/docs/cd/E17904_01/web.1111/e13702/redeploy.htm

Lab 12 – Creating a Deployment Plan with the Admin Console

It is easy to create a Deployment Plan using the Admin Console. Open Administration Console, navigate to the [Deployments](#) page, click on the [Lock&Edit](#) button to start making configuration changes.



ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help Welcome, weblogic Connected

Home > Summary of Deployments > ShoppingCart (Version2) > Summary of Deployments > Summary of Deployments

Summary of Deployments

Control Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or del domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

Customize this table

Deployments

Install Update Delete Start Stop Showing 1 to 2 of 2

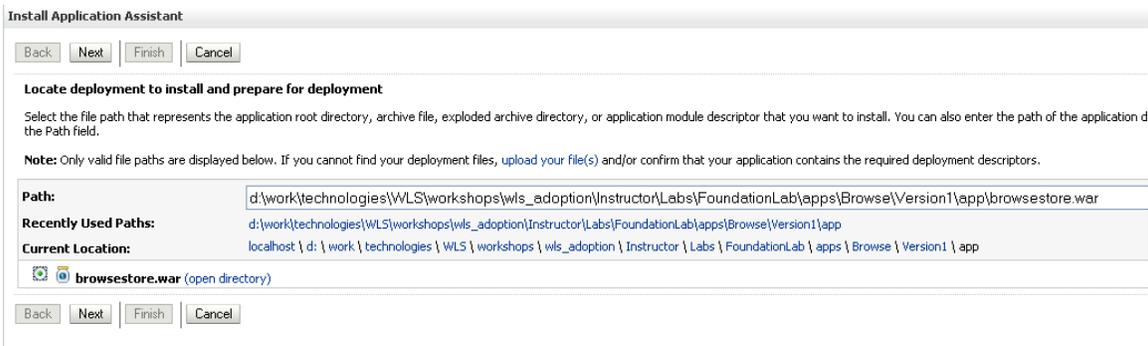
Name	State	Health	Type	Deployment Order
ShoppingCart (Version1)	Retired		Web Application	100
ShoppingCart (Version2)	Active	OK	Web Application	100

Install Update Delete Start Stop Showing 1 to 2 of 2

Click on the [Install](#) button. This will take you to the Install Application Assistant page. In the [Path](#) field you should type in the full pathname for the browsestore application inside your file system:

[%YOUR_PATH%/FoundationLab/apps/Browse/Version1/app/browsestore.war](#)

:



Install Application Assistant

Back Next Finish Cancel

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 d 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.

Path: d:\work\technologies\WLS\workshops\wls_adoption\Instructor\Labs\FoundationLab\apps\Browse\Version1\app\browsestore.war

Recently Used Paths: d:\work\technologies\WLS\workshops\wls_adoption\Instructor\Labs\FoundationLab\apps\Browse\Version1\app

Current Location: localhost | d: | work | technologies | WLS | workshops | wls_adoption | Instructor | Labs | FoundationLab | apps | Browse | Version1 | app

browsestore.war (open directory)

Back Next Finish Cancel

Click [Next](#) three times, and then click [Finish](#) to start the deployment. Click [Activate Changes](#). These changes are dynamic and there is no need to restart the server.

Go to the Deployments tab, check [browsestore](#) deployment and click [Start->Servicing all requests](#) (click Yes in the confirmation dialog).

Check that application is accessible using the following URL:

<http://localhost:7001/browsestore/>

Note that a deployment plan has automatically been created for [browsestore](#) application in the 'plan' folder (Plan.xml):

```

<?xml version="1.0" encoding="UTF-8" ?>
- <deployment-plan xmlns="http://www.bea.com/ns/weblogic/deployment-plan"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.bea.com/ns/weblogic/deployment-plan
  http://www.bea.com/ns/weblogic/deployment-plan/1.0/deployment-plan.xsd"
  global-variables="false">
  <application-name>browsestore</application-name>
- <module-override>
  <module-name>browsestore.war</module-name>
  <module-type>war</module-type>
- <module-descriptor external="false">
  <root-element>weblogic-web-app</root-element>
  <uri>WEB-INF/weblogic.xml</uri>
  </module-descriptor>
- <module-descriptor external="false">
  <root-element>web-app</root-element>
  <uri>WEB-INF/web.xml</uri>
  </module-descriptor>
- <module-descriptor external="false">
  <root-element>wldf-resource</root-element>
  <uri>META-INF/weblogic-diagnostics.xml</uri>
  </module-descriptor>
</module-override>
<config-root>C:\Labs\FoundationLab\apps\Browse\Version1\plan</config-root>
</deployment-plan>

```

You can use this deployment plan to override the information contained in the application’s deployment descriptors – for example, changing the context root. To do this, navigate to the Deployments page and click on the ‘browsestore’ deployment. Then select the **Configuration** tab to view and edit the configuration of the webapp deployment descriptor:

ORACLE WebLogic Server® Administration Console

Welcome, weblogic | Connected to: ...

Settings for browsestore

Overview | Deployment Plan | **Configuration** | Security | Targets | Control | Testing | Monitoring | Notes

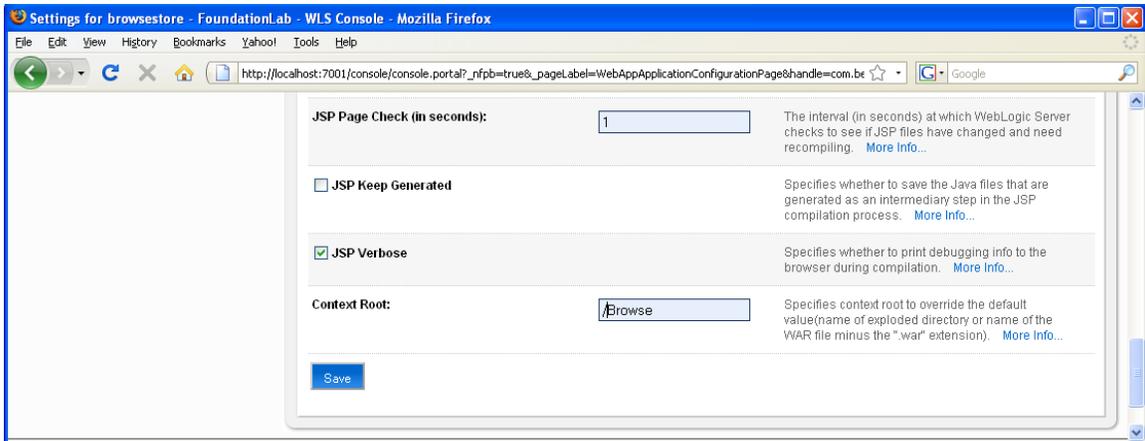
General | Logging | Workload | Instrumentation

Click the **Lock & Edit** button in the Change Center to modify the settings on this page.

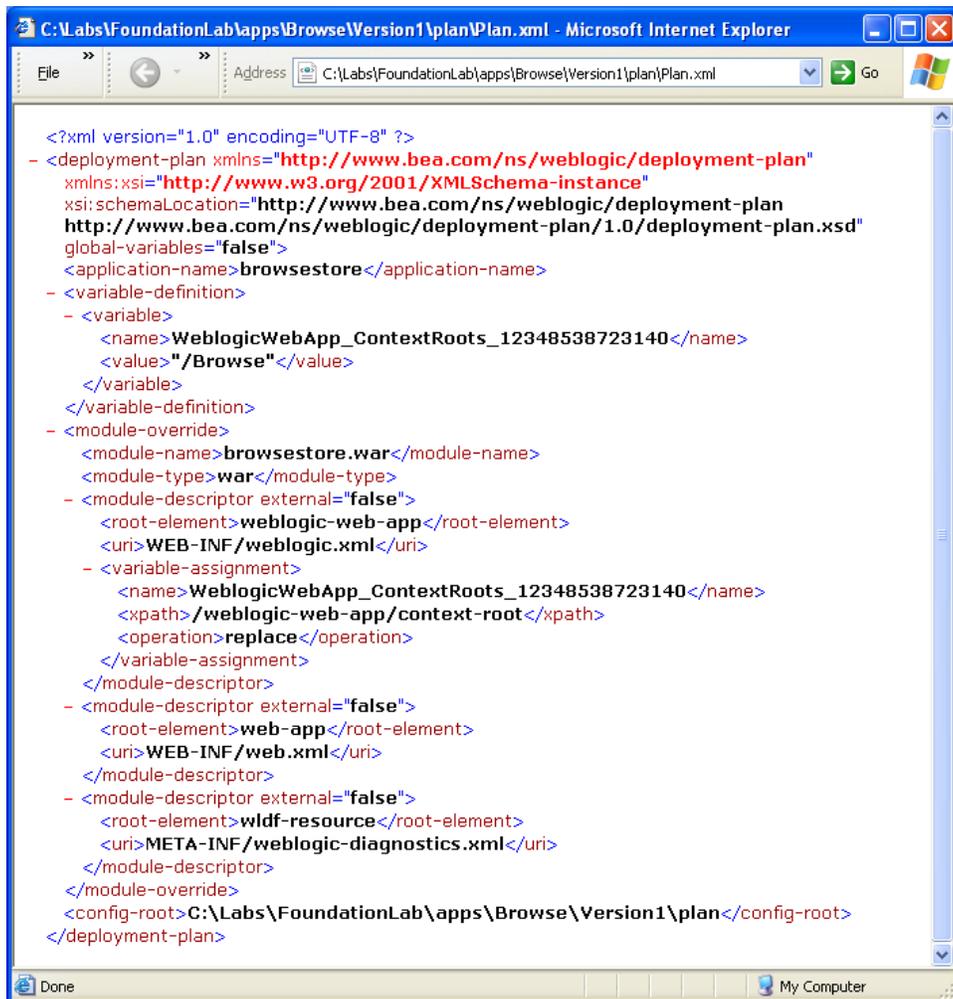
In this page, you define the configuration of the application deployment descriptor file that is associated with this Web application module.

Session cookies max age (in seconds):	<input type="text" value="-1"/>	The life span of the session cookie (in seconds) after which it expires on the client. The value can be set as any integer; the default value is -1 (unlimited). More Info...
Session Invalidation Interval (in seconds):	<input type="text" value="60"/>	The time (in seconds) that WebLogic Server waits between doing house-keeping for timed-out and invalid sessions, and deleting the old sessions and freeing up memory. More Info...
Session Timeout (in seconds):	<input type="text" value="3600"/>	The amount of time (in seconds) that a session can remain inactive before it is invalidated. More Info...
<input type="checkbox"/> Debug Enabled		Specifies whether to add JSP line numbers to generated class files to aid in debugging. More Info...
Maximum in-memory Sessions:	<input type="text" value="-1"/>	The maximum number of sessions to retain in memory. The default is -1 (unlimited). A negative value works the same as -1. More Info...
Monitoring Attribute Name:	<input type="text"/>	The monitoring attribute. More Info...
<input type="checkbox"/> Index Directory Enabled		Specifies whether the target should automatically generate an HTML directory suitable index file is found. More Info...

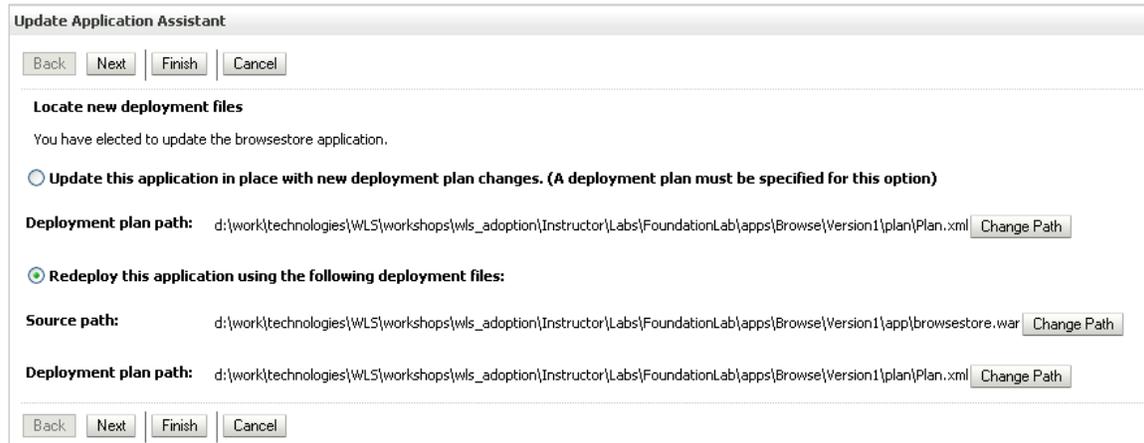
Click **Lock&Edit** button. Scroll down to the bottom of the page and set the **Context Root** for the webapp to **/Browse**. Save the changes:



Don't forget to re-deploy the application to reflect the new deployment plan (you will be prompted to do this by a message in the admin console):
 Take a look at the updated deployment plan (Plan.xml) and note how it includes the new context-root information:



Now you will need to update the application deployment to use this new deployment plan. Go to the [Deployments](#), check [browsestore](#) and click [Update](#) button, this will take you to the Update Application Assistant page. Select the option for “[Redeploy the application using the following deployment files](#)” (the path for the deployment plan should automatically be filled in for you) and click [Next](#):



Review the redeployment settings and click [Finish](#)

Notice the message reminding you to activate the pending changes. If you first navigate to the deployments page, you can see the [browsestore](#) application has status “[Update initializing](#)”. Click on the “[Activate Changes](#)” button in the admin console Change Center and you will see the status change to “[Active](#)”.

Open a browser and test the new context-root:

<http://localhost:7001/Browse>



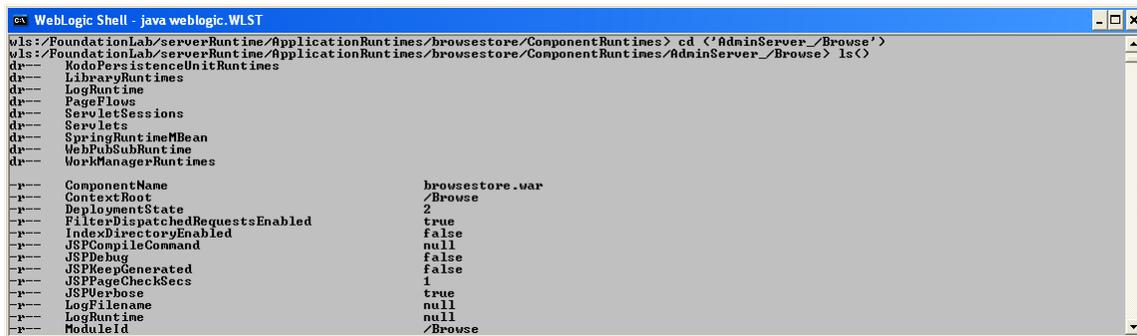
We will use WLST in interactive mode to browse the ServerRuntime MBean tree and view the configuration for the [browsestore.war](#) web application.

Open a command shell, set your environment (%MIDDLEWARE_HOME%\wls\server_10.3\server\bin\setWLSEnv.cmd) and start WLST. Note, that we still use admin port, and that's why we have to point trust key store

```
> java -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.WLST
```

You are now running WSLT in interactive mode (offline). Use the following commands to connect to the FoundationLab domain Admin Server , change to the ServerRuntime MBean tree and view the runtime configuration for the browsestore web application:

```
wls:/offline> connect ('weblogic', 'weblogic1', 't3s://localhost:9002')
...> serverRuntime()
...> cd ('ApplicationRuntimes')
...> cd ('browsestore')
...> cd('ComponentRuntimes')
...> cd ('AdminServer_/Browse')
...> ls()
```



Again, note the new [ContextRoot](#) for the application.

Lab 13 - Starting WebLogic Server as a Windows Service

To run a WebLogic Server instance as a Windows Service (daemon), you will need to create a service install script that sets a number of environment variables and calls the `installSvc.cmd` script from the `%MIDDLEWARE_HOME%/wlserver_10.3/server/bin` directory. There is an example that you can use to install the admin server for the `FoundationLab` domain as a Windows service in the `%LAB_HOME%/Scripts` folder (`installWinSvc.cmd`). Edit this file if necessary to pick up your WebLogic Server installation directory. There is also a second script for uninstalling the Windows service (`uninstallWinSvc.cmd`) – again, check that the paths are correct. You will find both these files listed at the end of this lab guide.

Since we are choosing to start the server in Production mode, an administrator username and password must be supplied, so we should make sure that `boot.properties` file exists in `<DOMAIN_HOME>/servers/AdminServer/security`. If the file does not exist, you need to create a text file called `boot.properties` and edit it to include the following two lines:

```
username=weblogic  
password=weblogic1
```

When WebLogic Server start up, it will look for the `boot.properties` file in this location, and use the username/password information it finds there to boot the server. When it finds the username/password in plaintext, it will replace these with an encrypted username/password, so that the security credentials are not visible. If you accidentally delete the `boot.properties` – don't worry, as long as you can remember the username/password – you can simply recreate the file as above and WebLogic Server will re-encrypt the credentials.

Copy the `installWindSvc.cmd` and `uninstallWinSvc.cmd` install script to your `FoundationLab` domain home directory, make sure that you have changed all the paths inside the scripts to reflect your directory structure, and double-click to run the script. You should see something like the output shown below, with a final message:

```
beasvc FoundationLab _AdminServer installed.
```

```
C:\WINDOWS\system32\cmd.exe
I~1.1-3\jre\bin;D:\dev\Oracle\MIDDLE~3\JROCKI~1.1-3\bin;D:\dev\Oracle\MIDDLE~3\W
LSERU~1.3\server\native\win\32\oci920_8" -password:"" -cmdline:"-jrockit -Xms128
m -Xmx256m -Dweblogic.Stdout="d:\dev\Oracle\Middleware11114\user_projects\domain
s\FoundationLab\stdout.txt" -Dweblogic.Stderr="d:\dev\Oracle\Middleware11114\use
r_projects\domains\FoundationLab\stderr.txt" -Xverify:none -classpath "D:\dev\
Oracle\MIDDLE~3\patch_wls1034\profiles\default\sys_manifest_classpath\weblogic_p
atch.jar;D:\dev\Oracle\MIDDLE~3\patch_oepe1040\profiles\default\sys_manifest_cla
sspath\weblogic_patch.jar;D:\dev\Oracle\MIDDLE~3\patch_ocp360\profiles\default\s
ys_manifest_classpath\weblogic_patch.jar;D:\dev\Oracle\MIDDLE~3\JROCKI~1.1-3\lib
\tools.jar;D:\dev\Oracle\MIDDLE~3\WLSERU~1.3\server\lib\weblogic_sp.jar;D:\dev\O
racle\MIDDLE~3\WLSERU~1.3\server\lib\weblogic.jar;D:\dev\Oracle\MIDDLE~3\modules
\features\weblogic.server.modules_10.3.4.0.jar;D:\dev\Oracle\MIDDLE~3\WLSERU~1.3
\server\lib\webservices.jar;D:\dev\Oracle\MIDDLE~3\modules\ORGAPPA~1.1\lib\ant-al
l.jar;D:\dev\Oracle\MIDDLE~3\modules\NETSFA~1.0_1\lib\ant-contrib.jar;.;C:\Progr
am Files\Java\jre6\lib\ext\QTJava.zip" -Dweblogic.Name=AdminServer -Dweblogic.
management.username= -Dweblogic.ProductionModeEnabled= -Djava.security.policy="\
D:\dev\Oracle\MIDDLE~3\WLSERU~1.3\server\lib\weblogic.policy\" weblogic.Server"

beasvc FoundationLab_AdminServer installed.

D:\dev\Oracle\Middleware11114\user_projects\domains\FoundationLab>ENDLOCAL

D:\dev\Oracle\Middleware11114\user_projects\domains\FoundationLab>ENDLOCAL

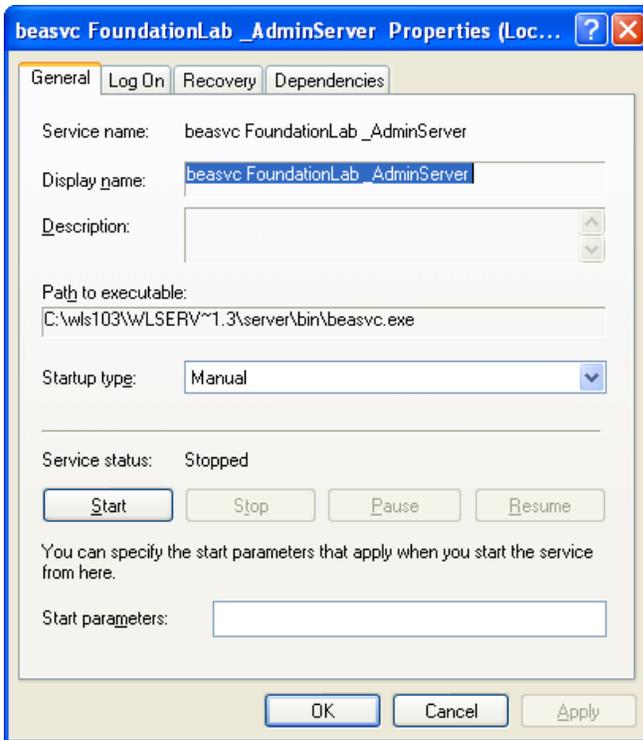
D:\dev\Oracle\Middleware11114\user_projects\domains\FoundationLab>
```

This (“beasvc FoundationLab_AdminServer”) is the name of the Windows service. Note that in multi-server domain, you would need to install all the managed servers as separate services and these would be called “beasvc FoundationLab_ManagedServer1” and so on.

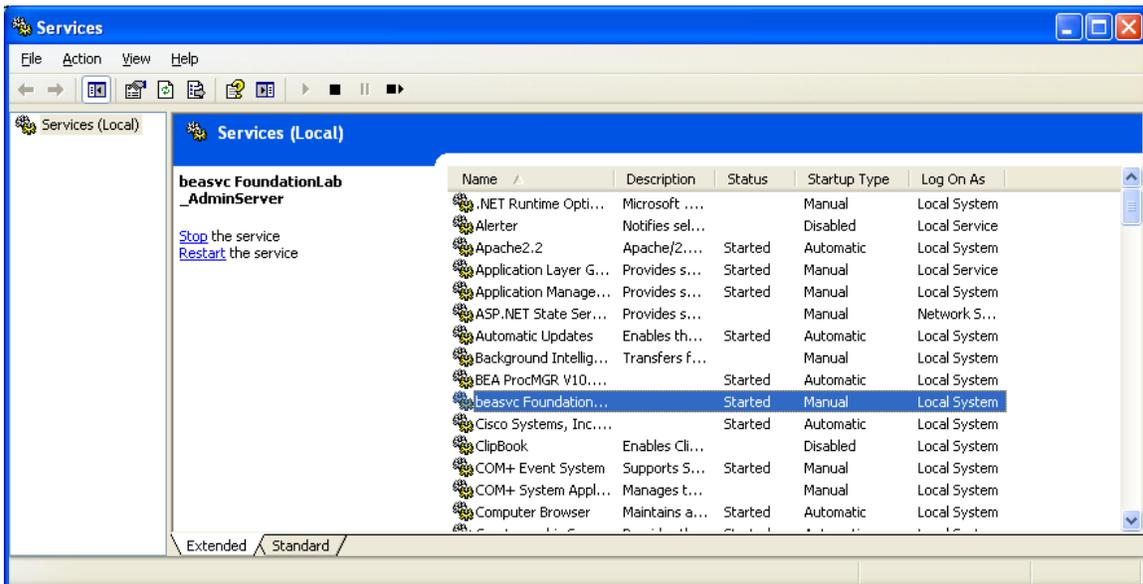
In practice, for a Production environment you would probably not want to install the Admin Server as a Windows Service, since for security reasons you would only want to run the server whenever a change to the domain configuration was required.

Make sure, that AdminServer for FoundationLab domain is not running. If it is running, please stop it before you continue.

Now open the Services folder from the Windows All Programs -> Administrative Tools -> Services menu. Open “beasvc FoundationLab_AdminServer” and you should see a dialog like the following:



Click on 'Start' and you should see a dialog box indicating that the service is starting. When startup is complete, you should now see that the service status has changed to Started:



Try opening the admin console to check that the server has started correctly:
<https://localhost:9002/console>

Stop the Windows Service by clicking the Stop button.
To uninstall the Windows service, use the [uninstallSvc.cmd](#) script.

More details can be found here:

- Setting Up a WebLogic Server Instance as a Windows Service - http://download.oracle.com/docs/cd/E17904_01/web.1111/e13708/winservice.htm#i1186180
- Installing and Uninstalling the Node Manager Service - http://download.oracle.com/docs/cd/E17904_01/doc.1111/e14142/postinst.htm#WLSIG235

The installSvc and uninstallSvc scripts are given below – you can adapt them for use with your own domains:

installWinSvc.cmd

```
@echo off
SETLOCAL
set DOMAIN_NAME=FoundationLab
set USERDOMAIN_HOME=c:\wls103\user_projects\domains\FoundationLab
set SERVER_NAME=AdminServer
set PRODUCTION_MODE=true
set JAVA_OPTIONS=-
Dweblogic.Stdout="c:\wls103\user_projects\domains\FoundationLab\stdout.txt" -
Dweblogic.Stderr="c:\wls103\user_projects\domains\FoundationLab\stderr.txt"
set MEM_ARGS=-Xms128m -Xmx256m
call "c:\wls103\wlserver_10.3\server\bin\installSvc.cmd"
ENDLOCAL
```

uninstallWinSvc.cmd

```
echo off
SETLOCAL
set DOMAIN_NAME=FoundationLab
set USERDOMAIN_HOME=c:\wls103\user_projects\domains\FoundationLab
set SERVER_NAME=AdminServer
call "c:\wls103\wlserver_10.3\server\bin\uninstallSvc.cmd"
ENDLOCAL
```