Module 6

Deploying Applications

At the end of this module you will be able to:

- ✓ Describe the Web server capabilities of WebLogic Server
- ✓ Use static and dynamic deployment
- ✓ Work with the built-in WebLogic Server Servlets
- ✓ Define and Work with Enterprise Applications

© 2006 BEA Systems, Inc.

Road Map

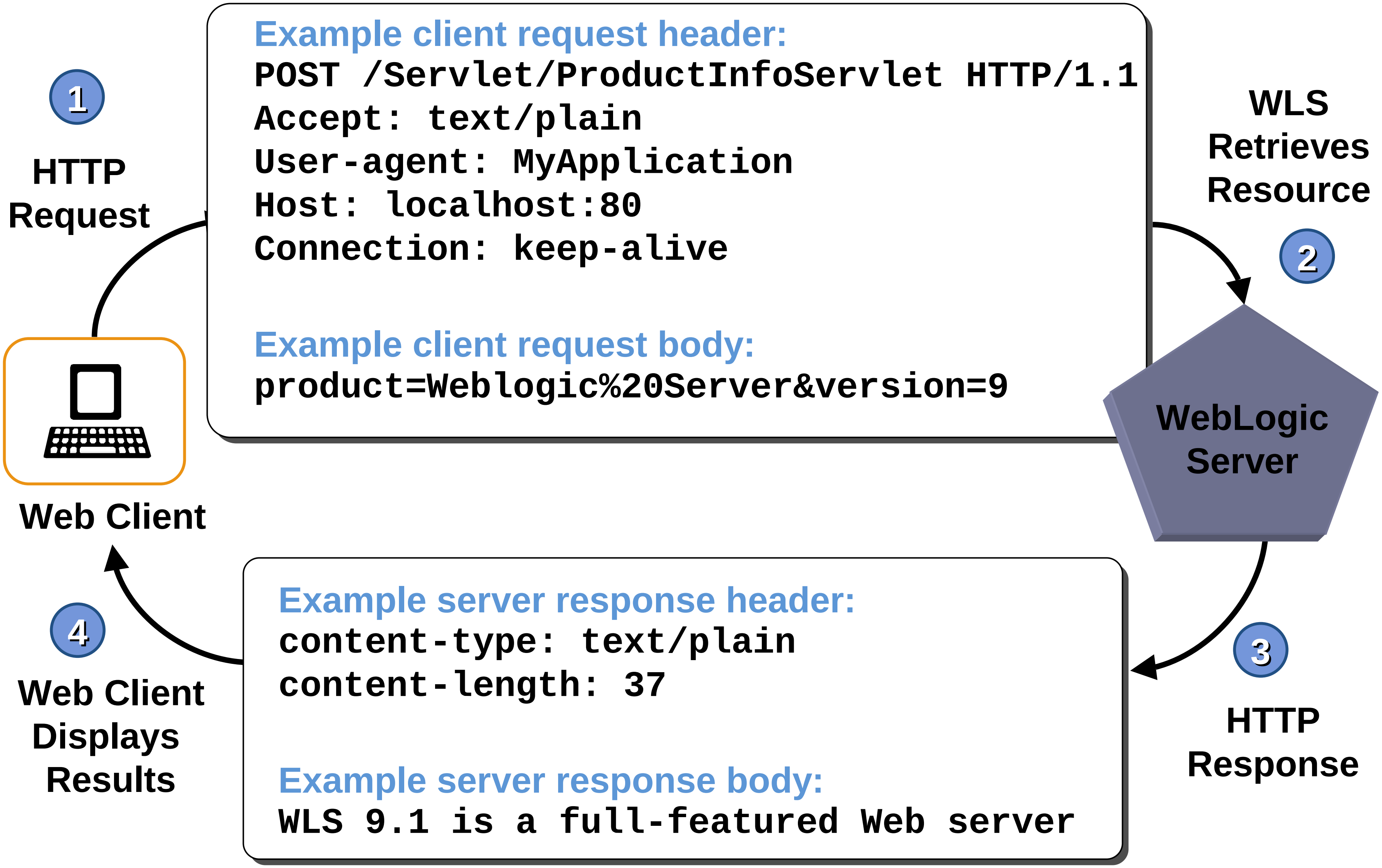
1. **Web Servers**
   - Web Servers Defined
   - HTTP
   - Static & Dynamic Content
2. Web Applications
3. EJB Applications
4. Enterprise Applications
5. Deployment
6. Advanced Deployment

The Role of Web Servers

- Web servers are responsible for handling HTTP requests from clients.
- Web servers typically return:
  - static content (HTML pages, graphics, …)
  - dynamic content (Servlet, JSPs, CGIs, …)

© 2006 BEA Systems, Inc.

A Typical Web Interaction



1 HTTP Request

Example client request header:

```
POST /Servlet/ProductInfoServlet HTTP/1.1
Accept: text/plain
User-agent: MyApplication
Host: localhost:80
Connection: keep-alive
```

Example client request body:

```
product=Weblogic%20Server&version=9
```

WLS Retrieves Resource 2

WebLogic Server

3 HTTP Response

Example server response header:

```
content-type: text/plain
content-length: 37
```

Example server response body:

```
WLS 9.1 is a full-featured Web server
```

4 Web Client Displays Results

Web Client

MIME Types

- Multipurpose Internet Mail Extensions (MIME) is a protocol for identifying and encoding binary data.
- All HTTP response data is encoded with a MIME content type.
- Browsers interpret HTTP response data differently depending upon the MIME type of the data:
  - HTML pages are parsed and displayed
  - PDF documents can be sent to Adobe Acrobat
  - application code can be directly executed

HTTP Status Codes

- HTTP status codes:
  - indicate to the client whether or not the request was successful
  - provide the client a reason for a failed request
  - used by clients to provide alternate behavior

Indicating success:

The default status code is 200, which indicates success.

Reason for failure:

A status code of 404 tells the client the requested resource was not found.

Providing alternate behavior:

If a browser receives a 401 status code, the browser can prompt the user for an ID and password to login. WLS 9.1 is a full-featured Web server.

© 2006 BEA Systems, Inc.

Static Content

- Static content documents are predefined on the server and do not change.
- WebLogic Server can be used to serve static content such as:
  - HTML documents
  - images
  - PDF documents
- WebLogic Server can serve static documents:
  - over standard HTTP
  - through SSL using HTTPS

© 2006 BEA Systems, Inc.

Dynamic Content

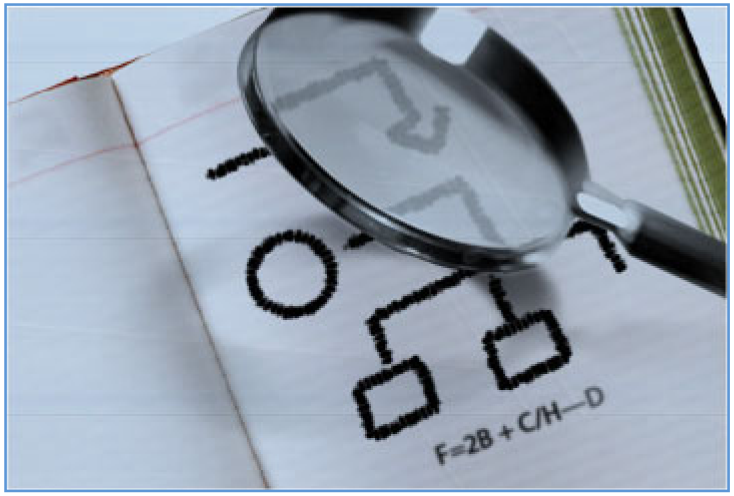
- Dynamic content documents may change based on the client’s request.
- HTML documents can be created on the fly by using:
  - Servlets
  - JavaServer Pages (JSPs)
  - Common Gateway Interface (CGI) programs

© 2006 BEA Systems, Inc.

Section Review

In this section we discussed:

- ✓ The role of Web servers
- ✓ HTTP requests, responses, MIME types, status codes
- ✓ Serving static HTML, images and files
- ✓ Serving JSP and Servlet requests



© 2006 BEA Systems, Inc.

Road Map

1. Web Servers
2. **Web Applications**
   - Web Applications
   - Directory Structure and Deployment Descriptors
   - Using the Console to Deploy Web Applications
   - Monitoring Web Applications
3. EJB Applications
4. Enterprise Applications
5. Deployment
6. Advanced Deployment

© 2006 BEA Systems, Inc.

What Is a Web Application?

- A *Web Application* is a group of server-side resources that create an interactive online application.
- Server-side resources include:
  - Servlets (small server-side applications)
  - JavaServer Pages (dynamic content)
  - static documents (HTML, images)
  - server-side classes
  - client-side applets and beans

© 2006 BEA Systems, Inc.

Packaging Web Applications

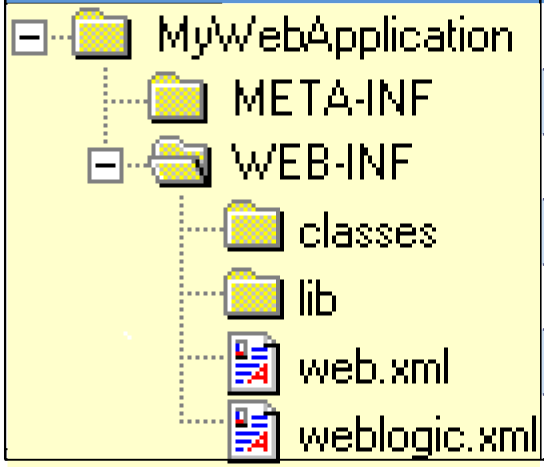
- Before deploying an application package and registering it with a WLS server.
- Follow these steps to package a Web App:

1. Arrange resources in a prescribed directory structure
2. Develop `web.xml` Deployment Descriptor (or copy as required)
3. Develop `weblogic.xml` Deployment Descriptor (WLS-Specific)
4. Archive Web App into `.war` file using jar
5. Deploy Web App onto WLS
6. Configure Web App with WLS Administration Console

© 2006 BEA Systems, Inc.

Web Application Structure

- The structure of Web Applications is defined by the Servlet specification.
- A Web Application can be either:
  - an archived file (`.war` file)
  - An expanded directory structure



Directory/File	Description
MyWebApplication	Document root of Web Application
META-INF	Information for archive tools (manifest)
WEB-INF	Private files that will not be served to clients
classes	Server-side classes such as Servlets and applet
lib	`.jar` files used by Web App
web.xml	Web App deployment descriptor
weblogic.xml	WLS-specific deployment descriptor

Configuring Web Applications

- Web applications are configured through *deployment descriptors* `web.xml` and `weblogic.xml` which:
  - Define run-time environment
  - Map URLs to Servlets and JSPs
  - Define application defaults such as welcome and error pages
  - Specify J2EE security constraints
  - Define work managers for applications
  - Set the context-root for the application

© 2006 BEA Systems, Inc.

The `web.xml` File

- The `web.xml` file is a deployment descriptor for configuring:
  - Servlets and JSP registration
  - Servlet initialization parameters
  - JSP tag libraries
  - MIME type mappings
  - Welcome file list
  - Error pages
  - Security constraints and roles
  - Resources
  - EJB references

© 2006 BEA Systems, Inc.

The `weblogic.xml` File

- The `weblogic.xml` is a WebLogic Server specific deployment descriptor for configuring:
  - JSP properties
  - JNDI mappings
  - security role mappings
  - HTTP session parameters
  - Work managers
  - Context root
  - Virtual directory mappings
  - Logging parameters
  - Library modules

© 2006 BEA Systems, Inc.

weblogic.xml Deployment Descriptor

- Example of weblogic.xml deployment descriptor.

```
<?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">

</weblogic-web-app>
```

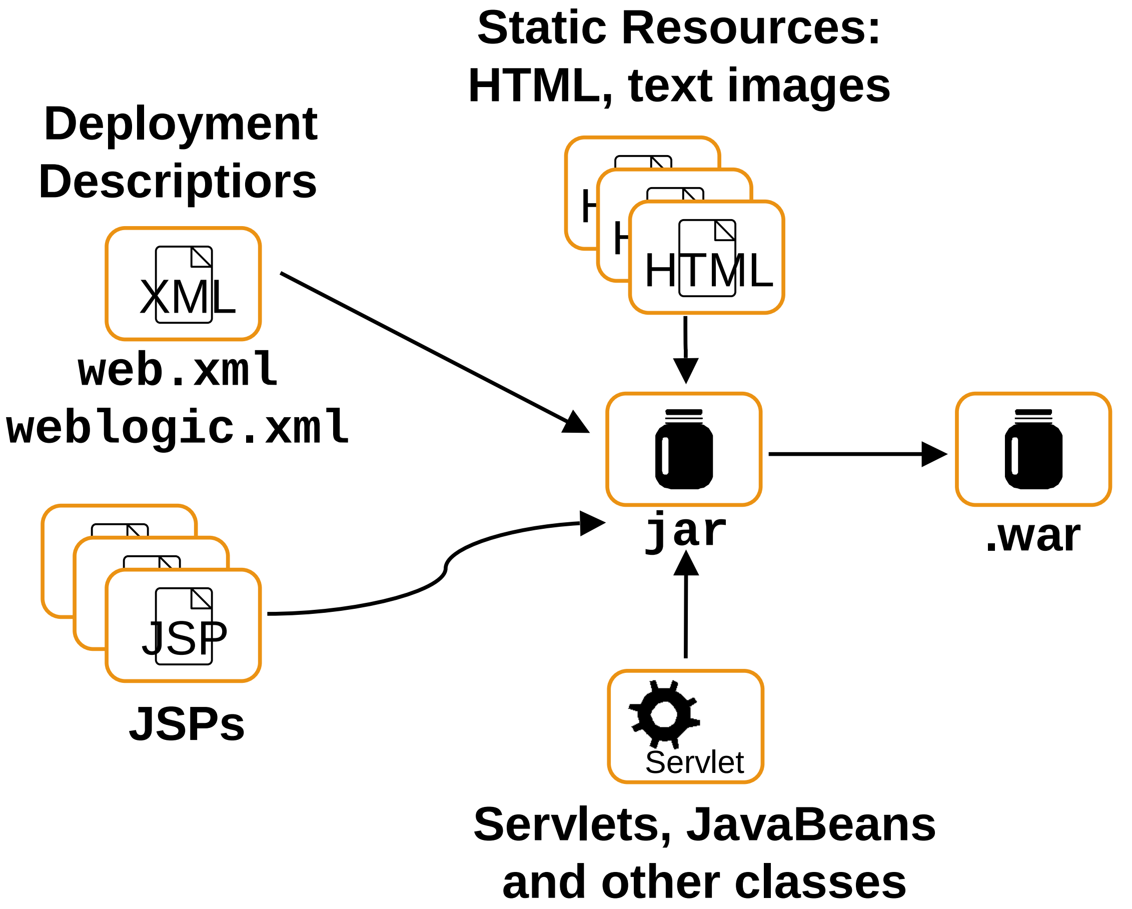
Web Application Archive…

- Web Application archives (`.war` files)
  - are compressed files that contain directory structures that represent Web Applications
  - simplify the distribution and sharing of Web Applications across a network
  - can share common resources
  - can be combined into larger applications
- For ease of development and debugging Web Applications are not archived until the end of production.

© 2006 BEA Systems, Inc.

...Web Application Archive

- Web archives are created using the `jar` utility:



Static Resources:
HTML, text images

Deployment
Descriptors

`web.xml`
`weblogic.xml`

JSPs

`jar`

.war

Servlets, JavaBeans
and other classes

© 2006 BEA Systems, Inc.

URLs and Web Apps

- The URL used to reference a resource in a Web Application must include the name of the Web Application.

Accessing a resource in a Web Application:

http://*hostname*:*port/MyWebApplication/resource*

Where:

Hostname	**Host name mapped to virtual host or** *hostname:port*
MyWebApplication	**Name of the Web Application; not necessary if this is the default Web Application**
resource	**Static page, Servlet mapping, or JSP**

Virtual Directory Mappings

- Virtual directories:
  - can be used to refer to physical directories
  - let you avoid the need to hard-code paths to physical directories
  - allow multiple Web applications to share common physical directories for specific requests such as images
  - decrease duplication of files across applications
  - are configured in `weblogic.xml`

Virtual Directory Mapping Example

Virtual Directory Mapping Example:

```
<virtual-directory-mapping>
   <local-path>c:/usr/gifs</local-path>
   <url-pattern>/images/*</url-pattern>
   <url-pattern>*.jpg</url-pattern>
</virtual-directory>
<virtual-directory-mapping>
   <local-path>c:/usr/common_jsps.jar</local-path>
   <url-pattern>*.jsp</url-pattern>
</virtual-directory>
```

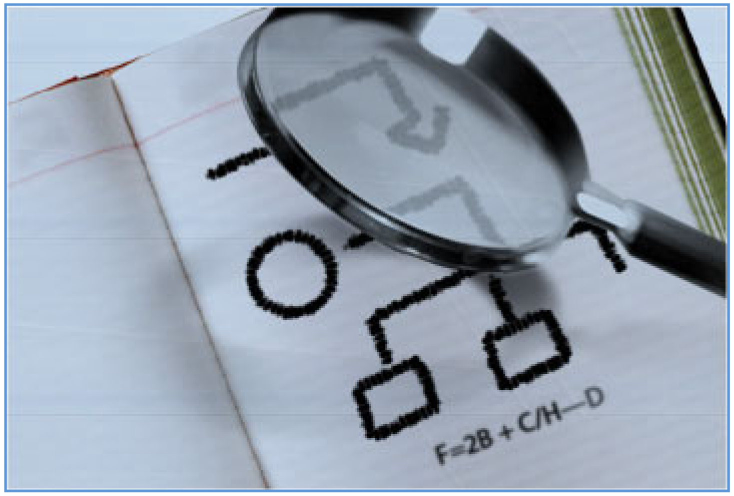
Archive vs. Expanded Directory

- Archive Web Applications if:
  - in production phase
  - deploying to several machines
- Do not archive Web Applications if:
  - in development/debugging phase
  - application will be updated frequently
  - deploying to a single machine (Administration server)

Section Review

In this section we discussed:

- ✓ Web applications
- ✓ Deployment descriptors
- ✓ Deployment & monitoring of Web Application
- ✓ Updating production Web Applications
- ✓ Virtual directories



© 2006 BEA Systems, Inc.

Road Map

1. Web Servers
2. Web Applications
3. **EJB Applications**
   - Major EJB Types and Their Purpose
4. Enterprise Applications
5. Deployment
6. Advanced Deployment

© 2006 BEA Systems, Inc.

Enterprise JavaBeans™

- *Enterprise JavaBeans*™ (EJB) standardizes development and deployment of Java server components.
- The EJB specification defines relationships between:
  - the EJB and its container
  - the container and the application server
  - the container and the client

© 2006 BEA Systems, Inc.

Types of EJBs

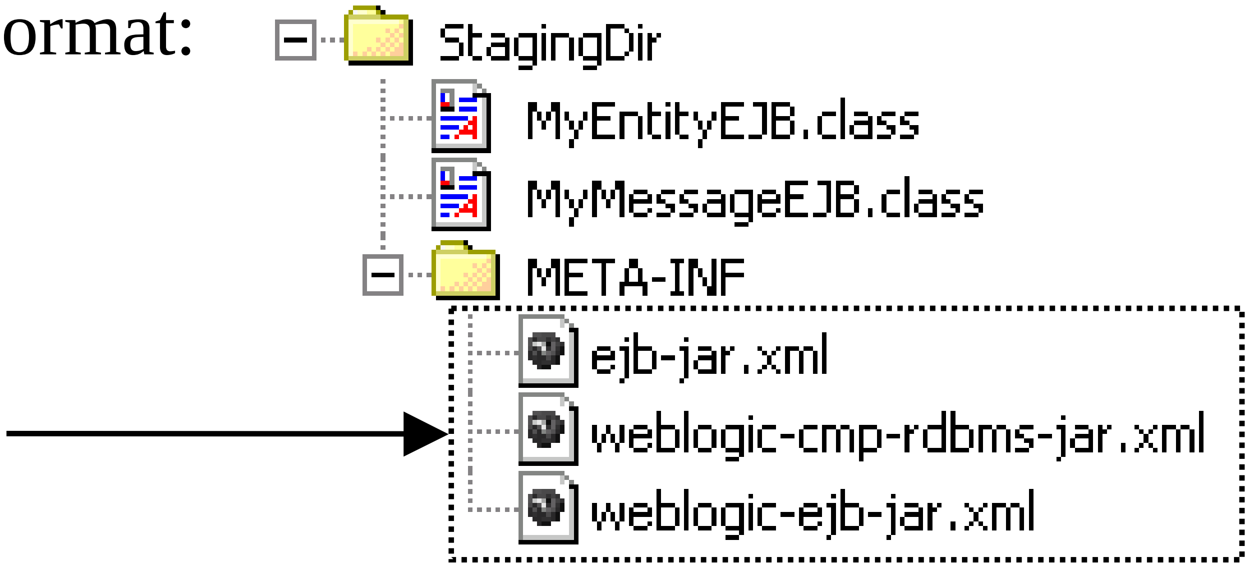
EJB Type	Description	Example
Stateless Session	▸Do not maintain state ▸Are synchronous ▸Are maintained in memory	▸Check validity of stock symbol ▸Calculate billing of a phone call
Stateful Session	▸Conversational interaction ▸Maintain state for client ▸Are synchronous	▸Book a flight & car rental for travel ▸Manage a shopping cart
Entity	▸Represent persisted data ▸Are synchronous	▸Represent a player's statistics ▸Represent a stock's history
Message Driven	▸Asynchronous & stateless ▸Consume JMS messages	▸Store logging messages

© 2006 BEA Systems, Inc.

EJB Application Directory Structure

- EJB components come packaged in JAR files.
- EJBs are configured by modifying deployment descriptors.

EJB App JAR Format:



```
StagingDir
    MyEntityEJB.class
    MyMessageEJB.class
    META-INF
        ejb-jar.xml
        weblogic-cmp-rdbms-jar.xml
        weblogic-ejb-jar.xml
```

EJB Administrator Tasks with WLS

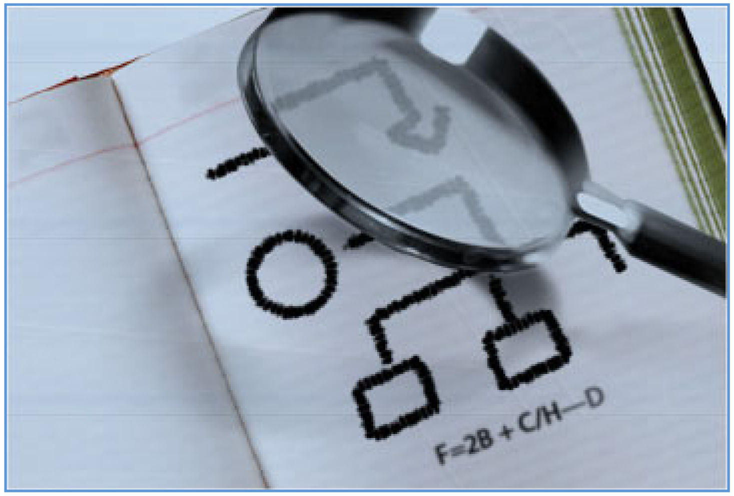
- EJB administrator tasks include:
  - configure and deploy
  - resolve JNDI and other infrastructure issues
  - monitor EJB caches and pools

© 2006 BEA Systems, Inc.

Section Review

In this section we discussed:

- ✓ EJB applications
- ✓ Major types of Enterprise JavaBeans (EJBs)



Road Map

1. Web Servers
2. Web Applications
3. EJB Applications
4. **Enterprise Applications**
   - Enterprise Application Concepts
   - Enterprise Archive (.ear) File Structure
   - Enterprise Application Configuration
5. Deployment
6. Advanced Deployment

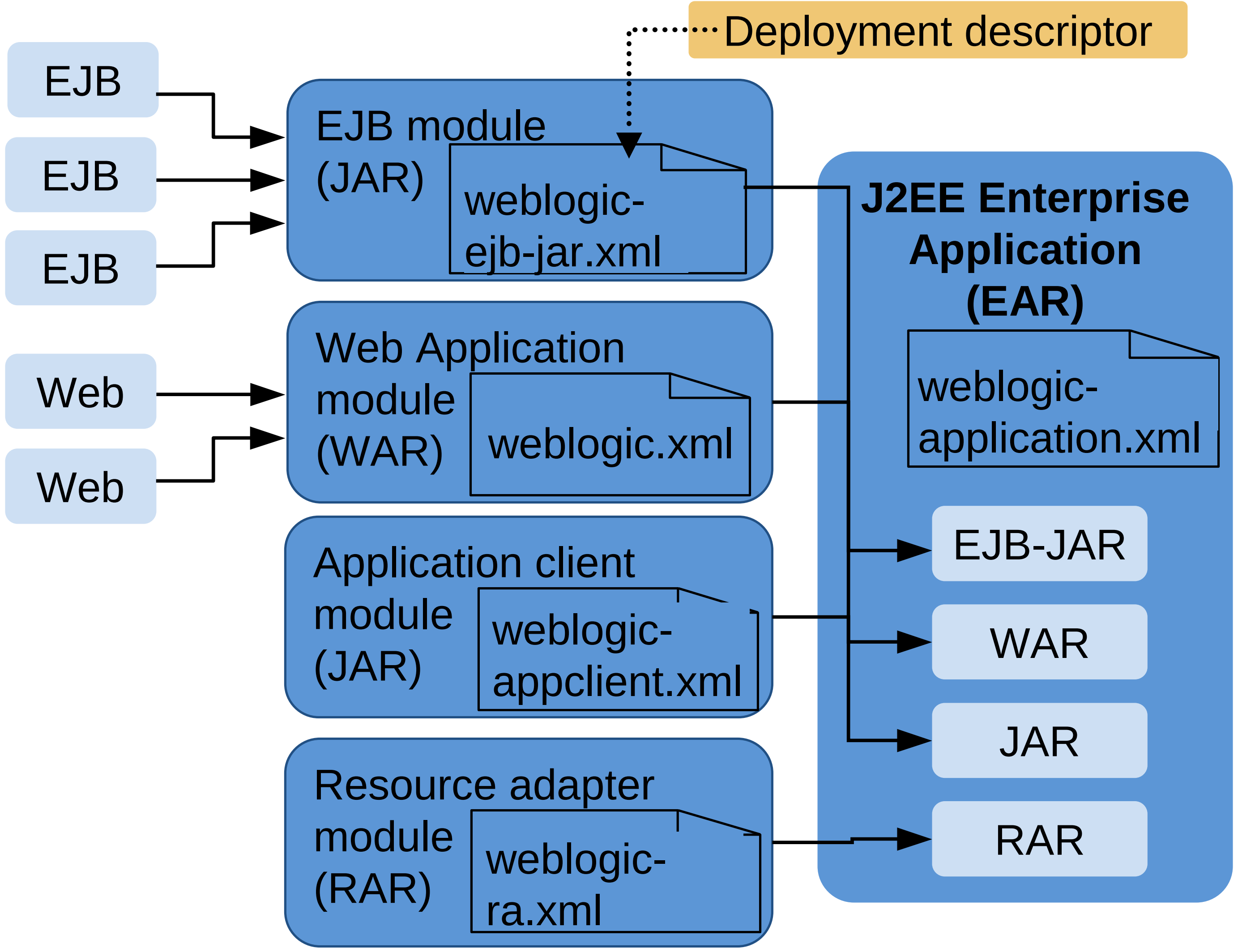
© 2006 BEA Systems, Inc.

What Is an Enterprise Application?

- An *enterprise application* is a grouping of several resources into one deployable unit packaged in an `.ear` file.
- These resources include:
  - Web applications (`.war`)
  - EJB applications (`.jar`)
  - Java applications (`.jar`)
  - Resource adapters (`.rar`)

© 2006 BEA Systems, Inc.

J2EE Enterprise Application



J2EE Modules

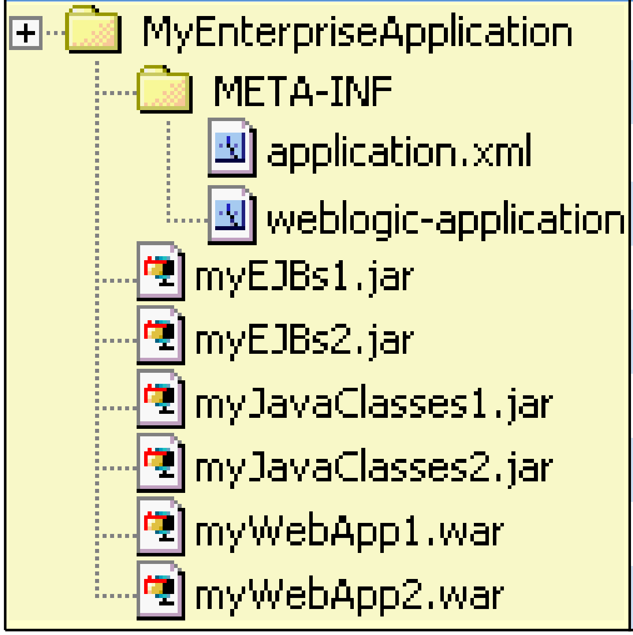
© 2006 BEA Systems, Inc.

Why Enterprise Applications?

- Use enterprise applications to:
  - Avoid name space clashes
  - Declare application-wide security roles
  - Deploy an application as one unit
  - Share application-wide EJB resources
  - Configure local JDBC datasources
  - Configure local JMS resources
  - Configure local XML resources

EAR File Structure

- An example directory structure of an enterprise application is shown below:



Directory / File	Description
MyEnterpriseApplication	Document root of enterprise application
META-INF	META-INF directory
application.xml	Enterprise application deployment descriptor
weblogic-application	WLS Enterprise application deployment descriptor
myEJBs1.jar	An EJB module
myEJBs2.jar	Another EJB module
myJavaClasses1.jar	A Java module
myJavaClasses2.jar	Another Java module
myWebApp1.war	A Web Application module
myWebApp2.war	Another Web Application module

© 2006 BEA Systems, Inc.

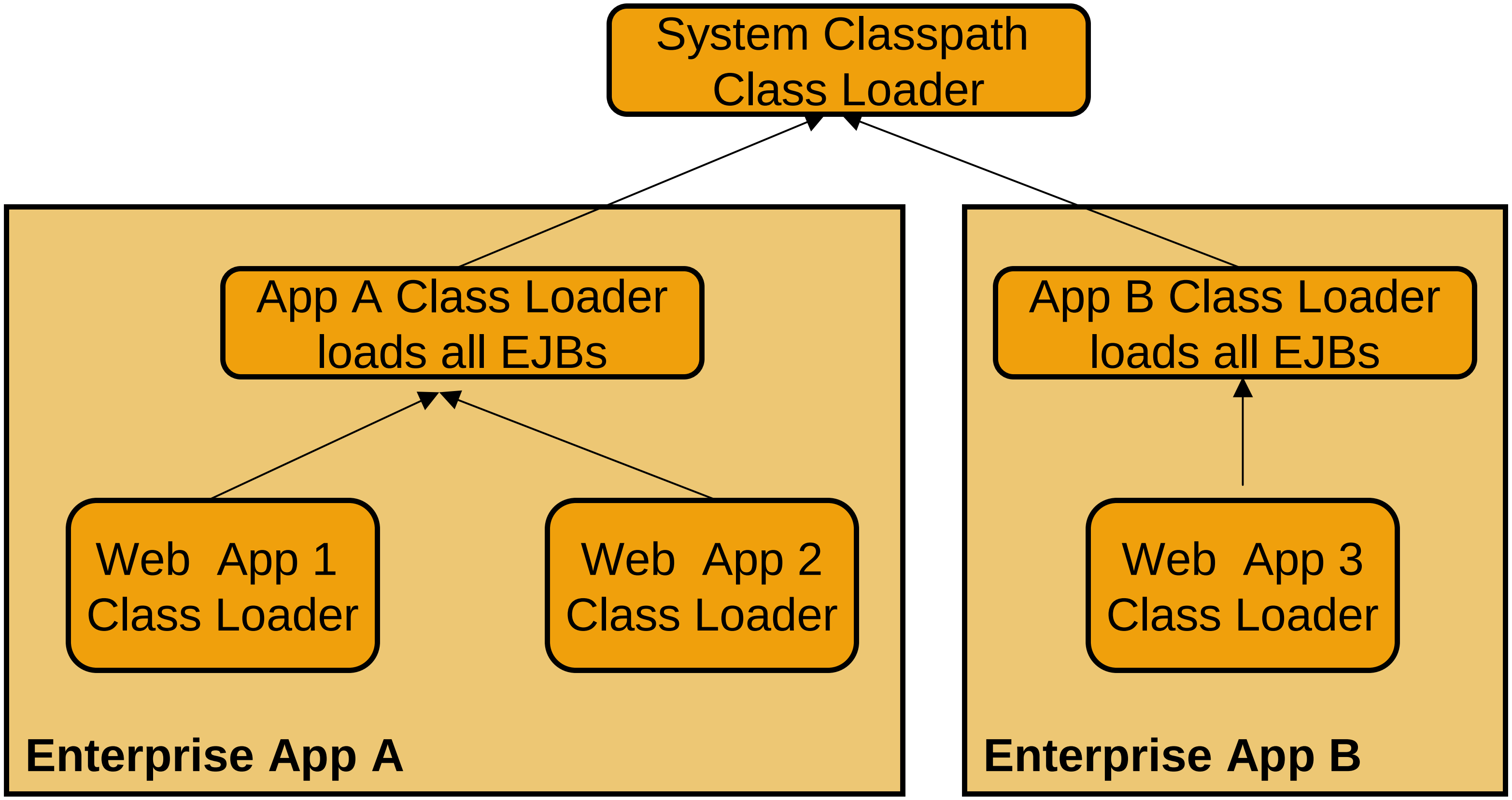
Configuring WLS Specific Features

- Configure Enterprise-wide WLS specific features with `weblogic-application.xml` deployment descriptor:
  - XML parsers
  - XML entity mappings
  - JDBC datasources
  - JMS connection factories and destinations
  - Security realms

© 2006 BEA Systems, Inc.

WLS Application Class Loader

- Each application receives its own class loader hierarchy with the system class loader as its parent.



© 2006 BEA Systems, Inc.

EAR Class Libraries

- Extending the J2EE spec, BEA has added `APP-INF/lib` and `APP-INF/classes` to the standard J2EE ear file structure.
- When the application is initialized, paths extracted are appended to the beginning of the application’s CLASSPATH
- Classes are added to the root classloader of the application.

© 2006 BEA Systems, Inc.

J2EE Library Support

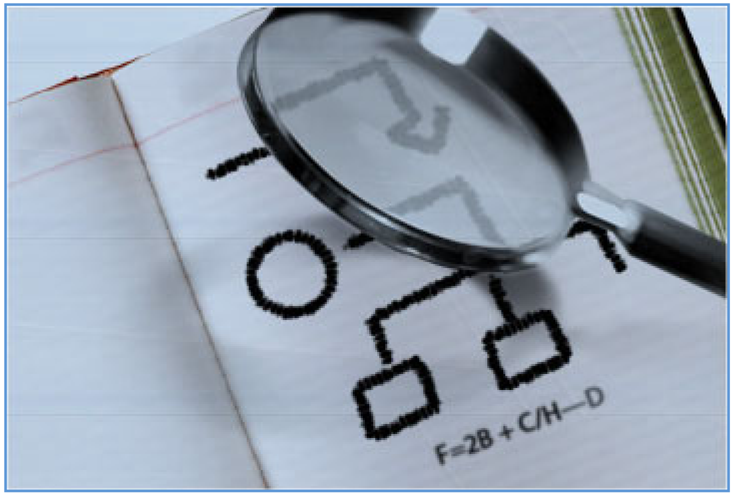
- To make things easier, you can create a library of J2EE modules, package them into an Enterprise application (EAR) then deploy and register it with the application container.
- Afterwards, other applications can use the modules as if they were packaged in their own EAR files.
- This allows for more reusability between applications.

© 2006 BEA Systems, Inc.

Section Review

In this section we discussed:

- ✓ The structure of Enterprise Applications
- ✓ Deploying Enterprise Applications



Road Map

1. Web Servers
2. Web Applications
3. EJB Applications
4. Enterprise Applications
5. **Deployment**
   - Auto-deployment
   - Console Deployment
   - Command-line Deployment
6. Advanced Deployment

© 2006 BEA Systems, Inc.

Deployment Process Overview

- Deploying an application involves the following tasks:
  1. *Preparing* - Choosing whether to package the application as an archived file or keeping it in an exploded directory
  2. *Configuring* – Creating a deployment plan to maintain configuration changes without changing the deployment descriptors
  3. *Deploying* – Targeting and distributing the application to WebLogic servers in a domain

© 2006 BEA Systems, Inc.

Deployment Methods

- WLS supports three deployment methods:
  - auto-deployment
  - console deployment
  - command-line deployment
- You can deploy:
  - Enterprise applications
  - Web applications
  - Web Services
  - J2EE libraries
  - EJB components
  - Resource adapters
  - Optional packages
  - Client application archives
  - JDBC, JMS and Diagnostic Framework modules
- Applications and EJBs can be deployed:
  - in an archived file (.ear, .war, .jar)
  - or in exploded (open) directory format

© 2006 BEA Systems, Inc.

Auto-Deployment – Copying Files

- If Production Mode is OFF:
  - You can *install* an application simply by copying it (manually or using the console) to the ‘`autodeploy`’ folder of the domain
  - The Administration Server monitors this folder for new, changed or removed applications
  - This *configures*, *targets* and *deploys* the application to the Administration server only

Location of Applications Directory:

```
%BEA_HOME%\user_projects\domains\domain_name\autodeploy
```

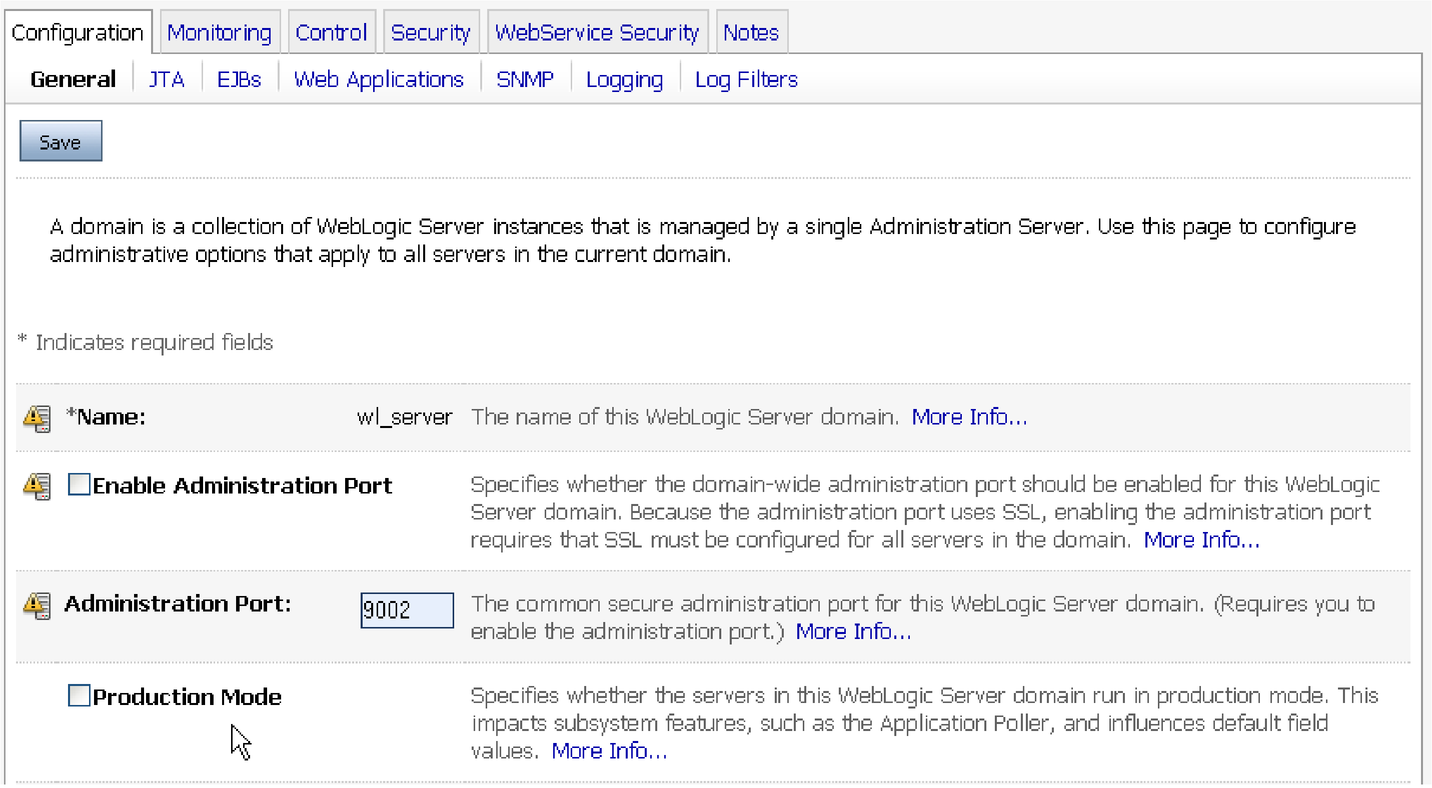
Development vs. Production Modes

- An Administration server starts either using:
  - *development mode*, which turns auto-deployment on
  - *production mode*, which turns auto-deployment off
- The Administration server starts in the mode selected at domain creation time.
- The mode is set for all WebLogic servers in a given domain.

© 2006 BEA Systems, Inc.

Production Mode Flag

- When Production mode is disabled, applications can be dynamically deployed.
  - Application poller will be enabled in Development Mode.



A domain is a collection of WebLogic Server instances that is managed by a single Administration Server. Use this page to configure administrative options that apply to all servers in the current domain.

* Indicates required fields

*Name: wl_server The name of this WebLogic Server domain. More Info...

□ Enable Administration Port Specifies whether the domain-wide administration port should be enabled for this WebLogic Server domain. Because the administration port uses SSL, enabling the administration port requires that SSL must be configured for all servers in the domain. More Info...

Administration Port: 9002 The common secure administration port for this WebLogic Server domain. (Requires you to enable the administration port.) More Info...

□ Production Mode Specifies whether the servers in this WebLogic Server domain run in production mode. This impacts subsystem features, such as the Application Poller, and influences default field values. More Info...

© 2006 BEA Systems, Inc.

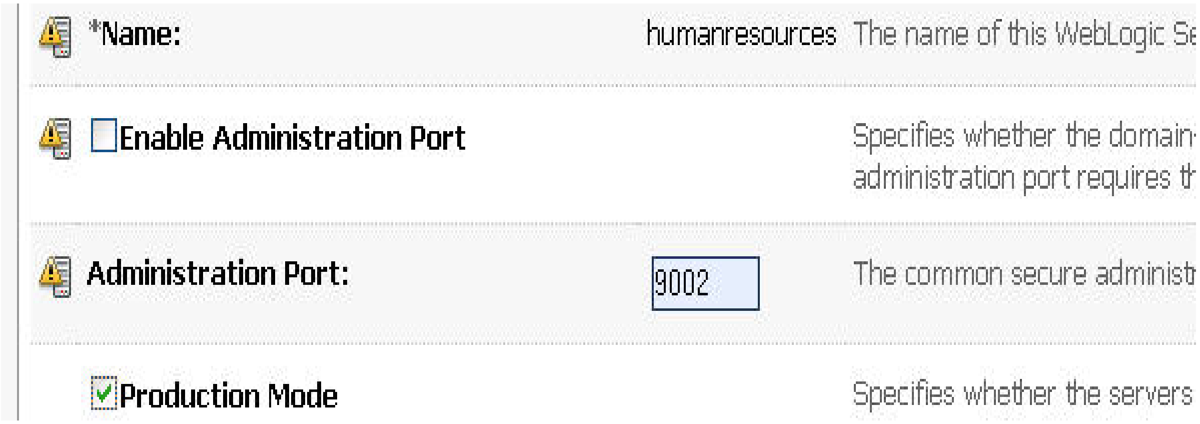
Console Deployment Method…

- Deploying with the console allows full administrator control:
  - *Installation* from a location of your choice
  - Manual *configuration* of application name
  - *Targeting* of application to individual servers and/or clusters
  - *Configuring* the application without targeting it
  - Activating *deployment* when desired

© 2006 BEA Systems, Inc.

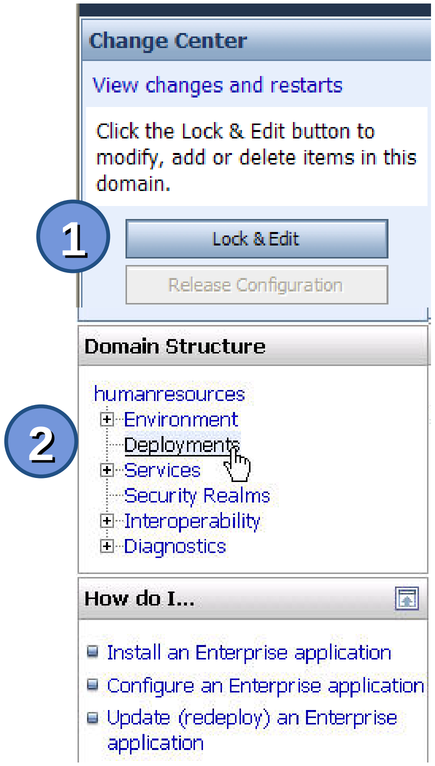
...Console Deployment Method

- Best used with Production Mode

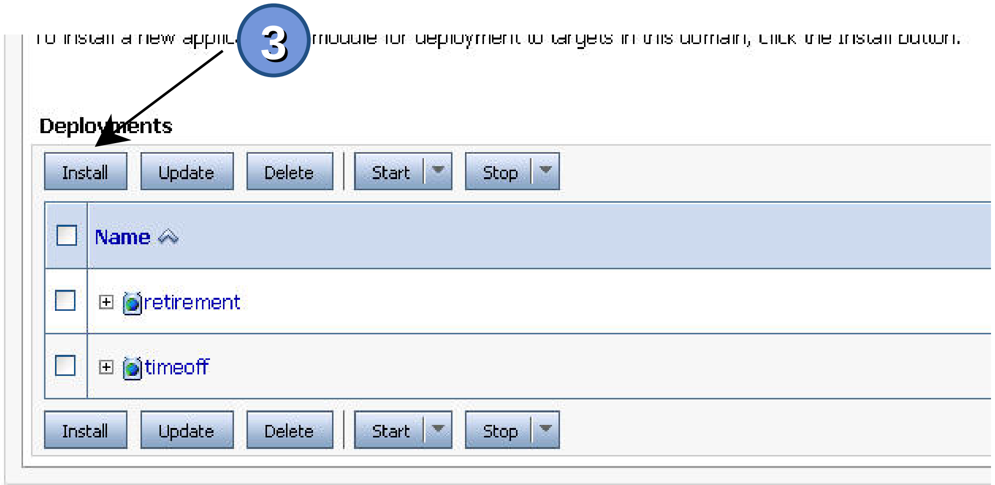


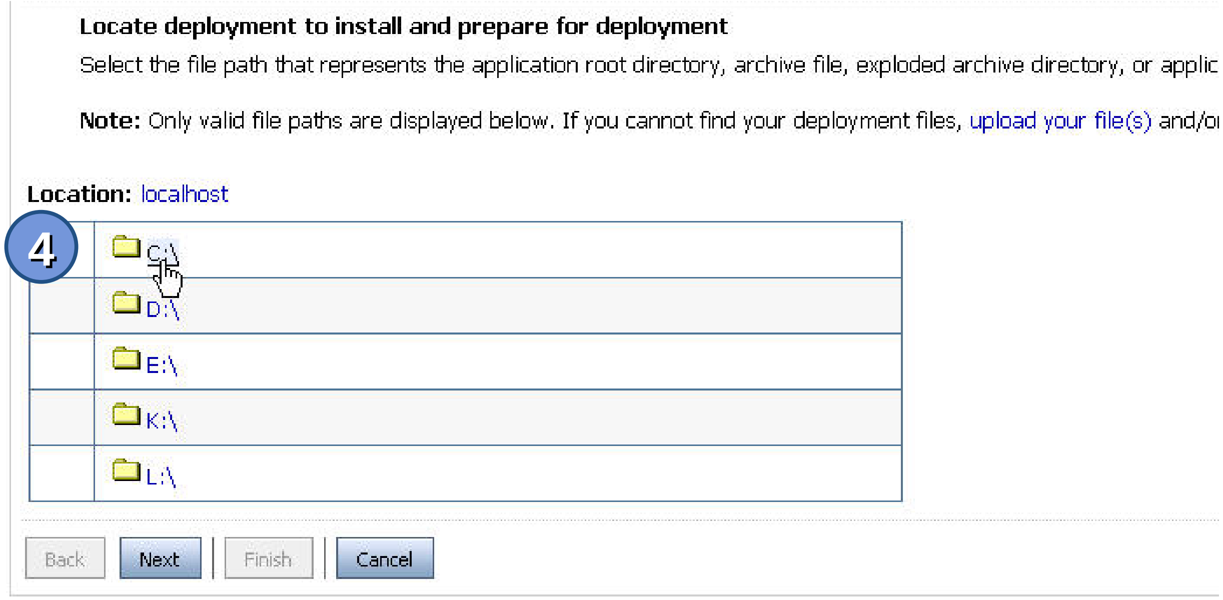
*Name:	humanresources	The name of this WebLogic Se
□ Enable Administration Port		Specifies whether the domain administration port requires th
Administration Port:	9002	The common secure administ
☑ Production Mode		Specifies whether the servers

Console Deployment…



Click the Lock & Edit button to modify, add or delete items in this domain.


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


Locate deployment to install and prepare for deployment

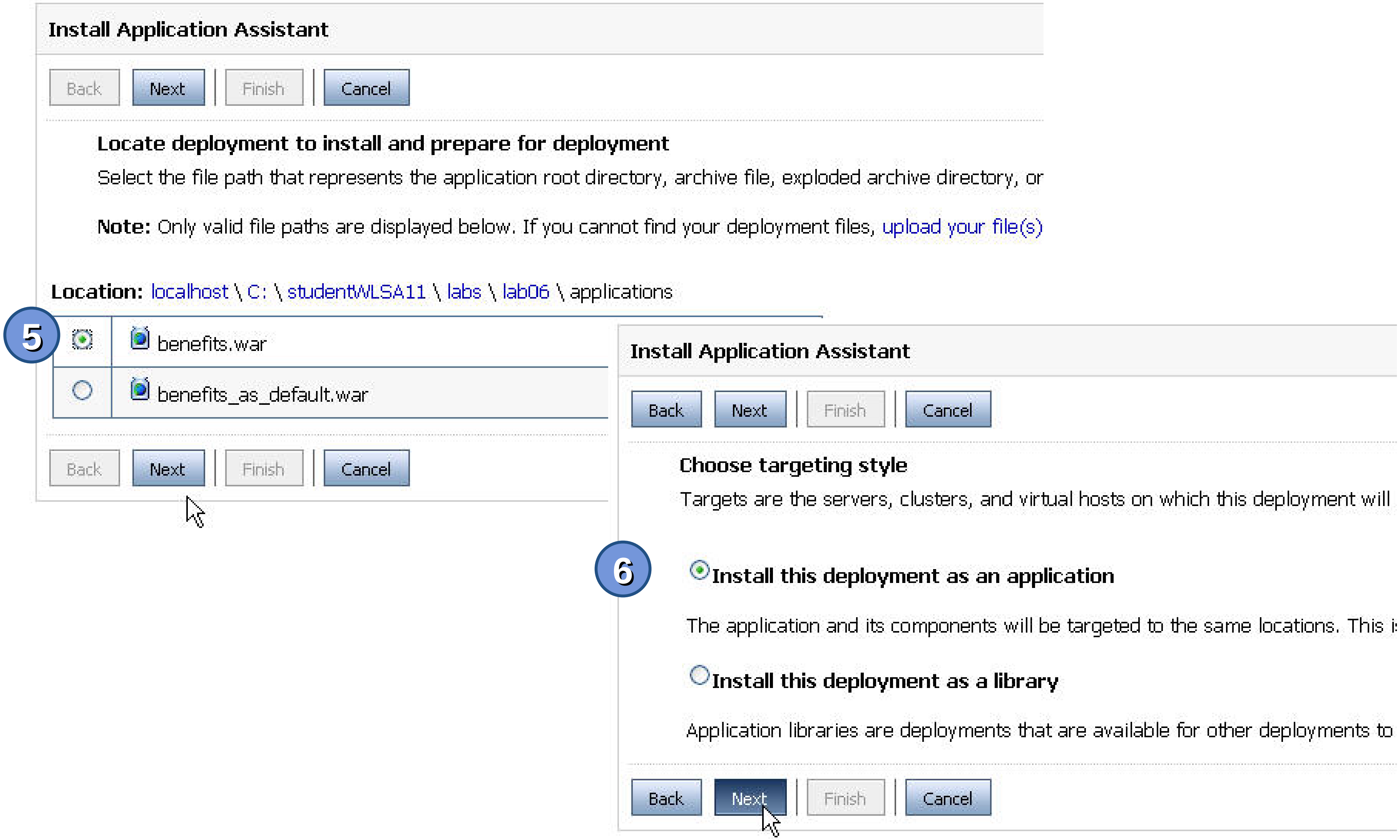
Select the file path that represents the application root directory, archive file, exploded archive directory, or applic

Note: Only valid file paths are displayed below. If you cannot find your deployment files, upload your file(s) and/o

Location: localhost

© 2006 BEA Systems, Inc.

...Console Deployment...



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

Note: Only valid file paths are displayed below. If you cannot find your deployment files, upload your file(s)

Location: localhost \ C: \ studentWLSA11 \ labs \ lab06 \ applications

● benefits.war

○ benefits_as_default.war

Install Application Assistant

Choose targeting style

Targets are the servers, clusters, and virtual hosts on which this deployment will

● **Install this deployment as an application**

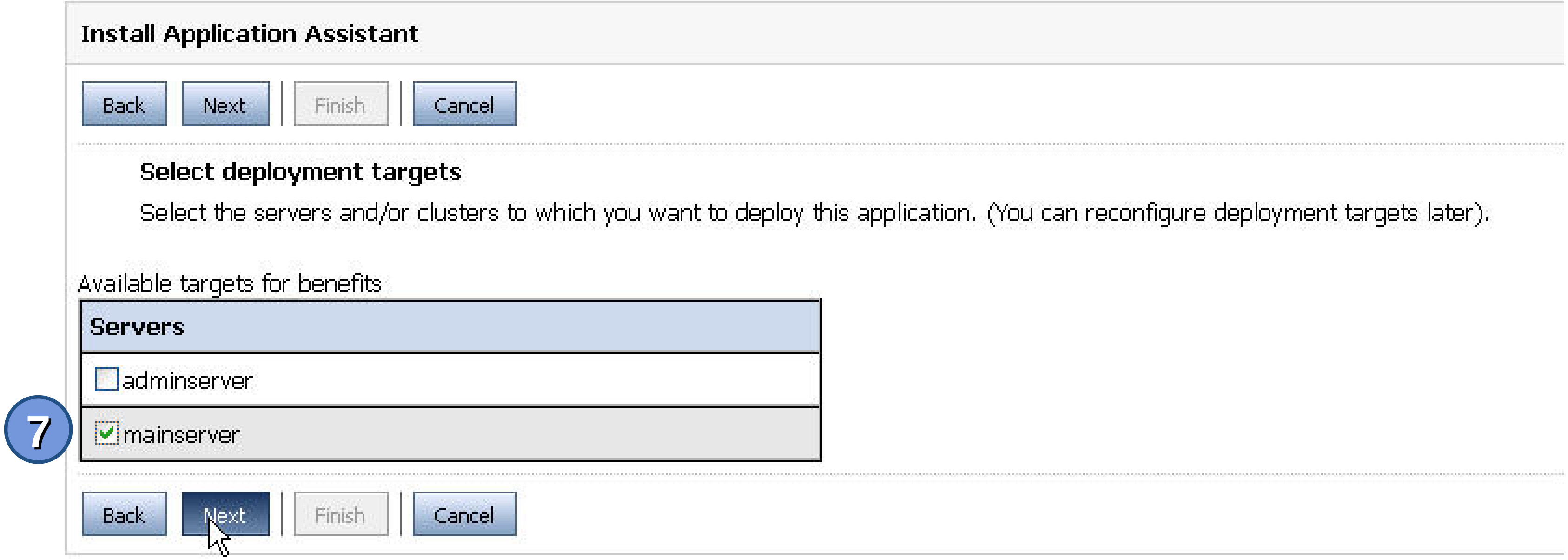
The application and its components will be targeted to the same locations. This i

○ **Install this deployment as a library**

Application libraries are deployments that are available for other deployments to

© 2006 BEA Systems, Inc.

…Console Deployment…



Install Application Assistant

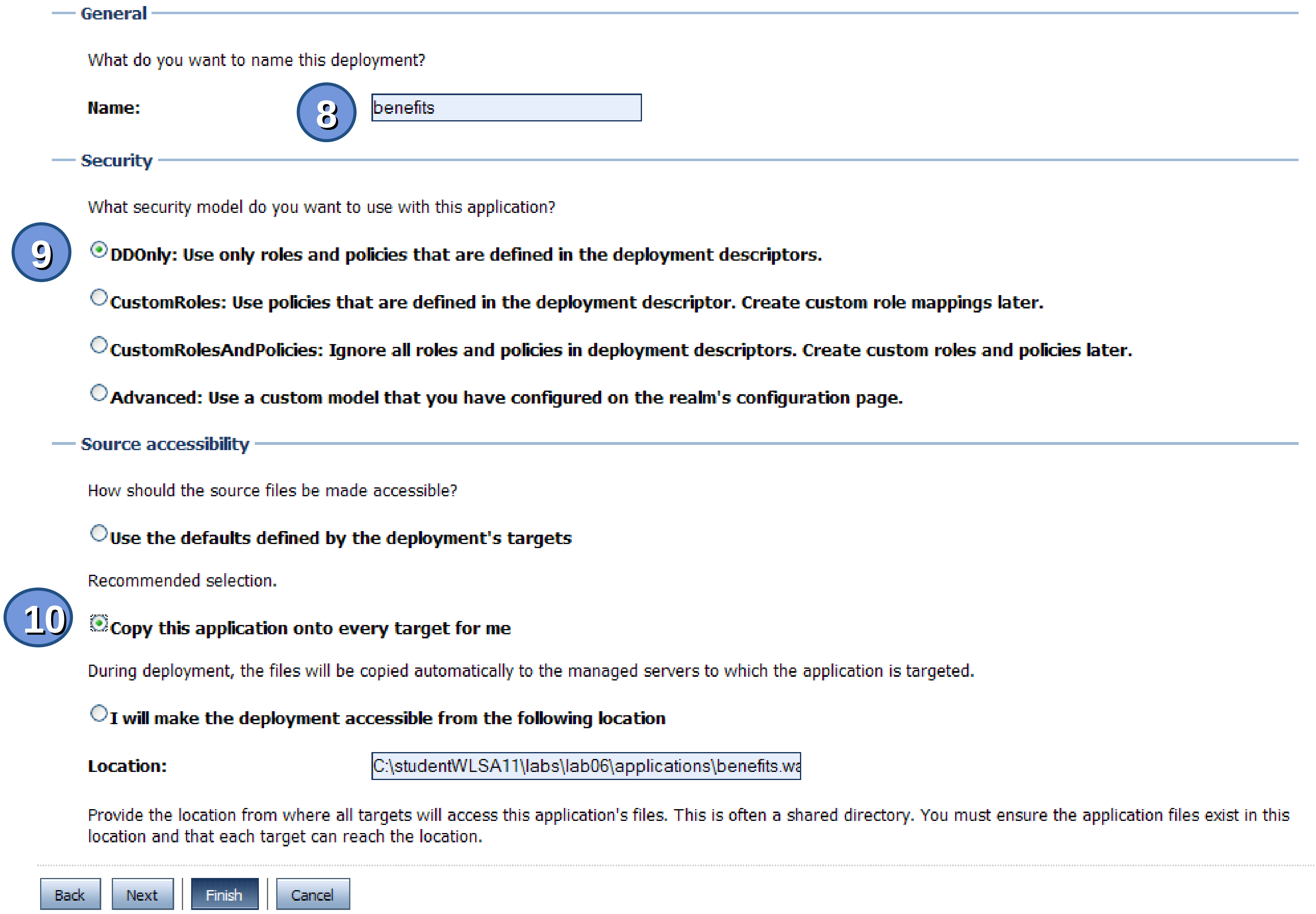
Select deployment targets

Select the servers and/or clusters to which you want to deploy this application. (You can reconfigure deployment targets later).

Available targets for benefits

Servers
□ adminserver
☑ mainserver

...Console Deployment...



General

What do you want to name this deployment?

Name: benefits

Security

What security model do you want to use with this application?

● **DDOnly: Use only roles and policies that are defined in the deployment descriptors.**

○ **CustomRoles: Use policies that are defined in the deployment descriptor. Create custom role mappings later.**

○ **CustomRolesAndPolicies: Ignore all roles and policies in deployment descriptors. Create custom roles and policies later.**

○ **Advanced: Use a custom model that you have configured on the realm's configuration page.**

Source accessibility

How should the source files be made accessible?

○ **Use the defaults defined by the deployment's targets**

Recommended selection.

● **Copy this application onto every target for me**

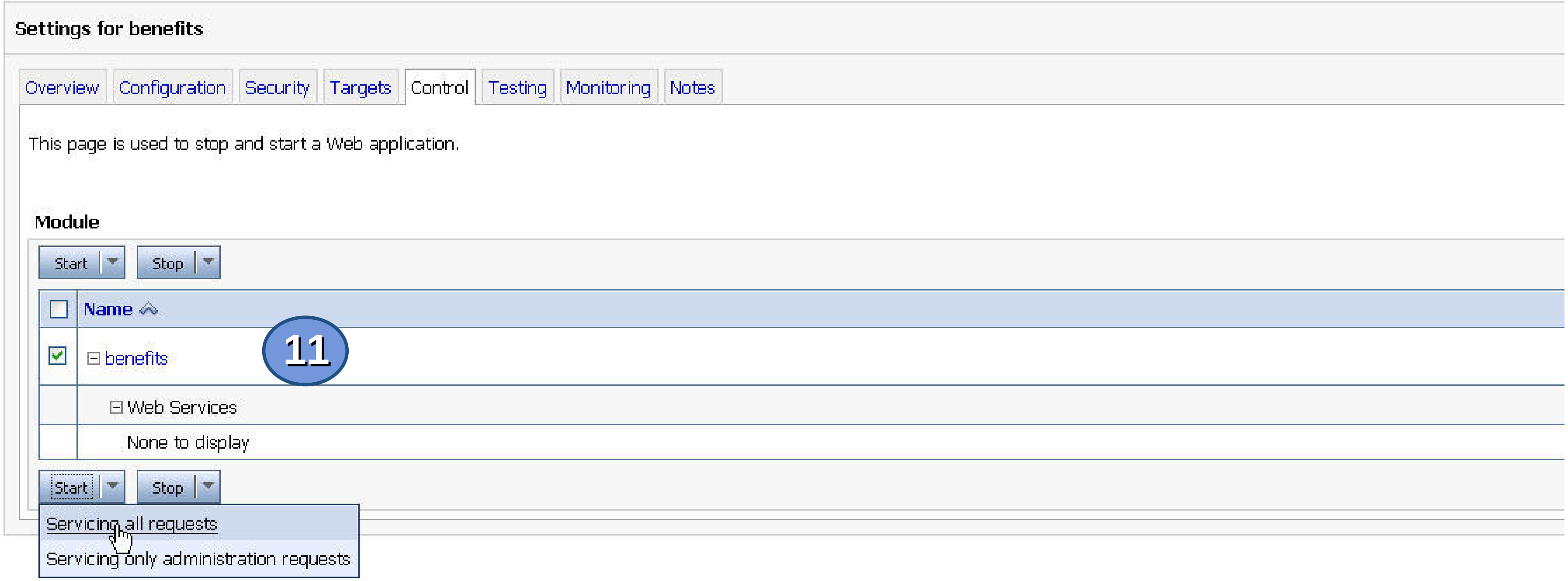
During deployment, the files will be copied automatically to the managed servers to which the application is targeted.

○ **I will make the deployment accessible from the following location**

Location: C:\studentWLSA11\labs\lab06\applications\benefits.wa

Provide the location from where all targets will access this application's files. This is often a shared directory. You must ensure the application files exist in this location and that each target can reach the location.

…Console Deployment



Settings for benefits

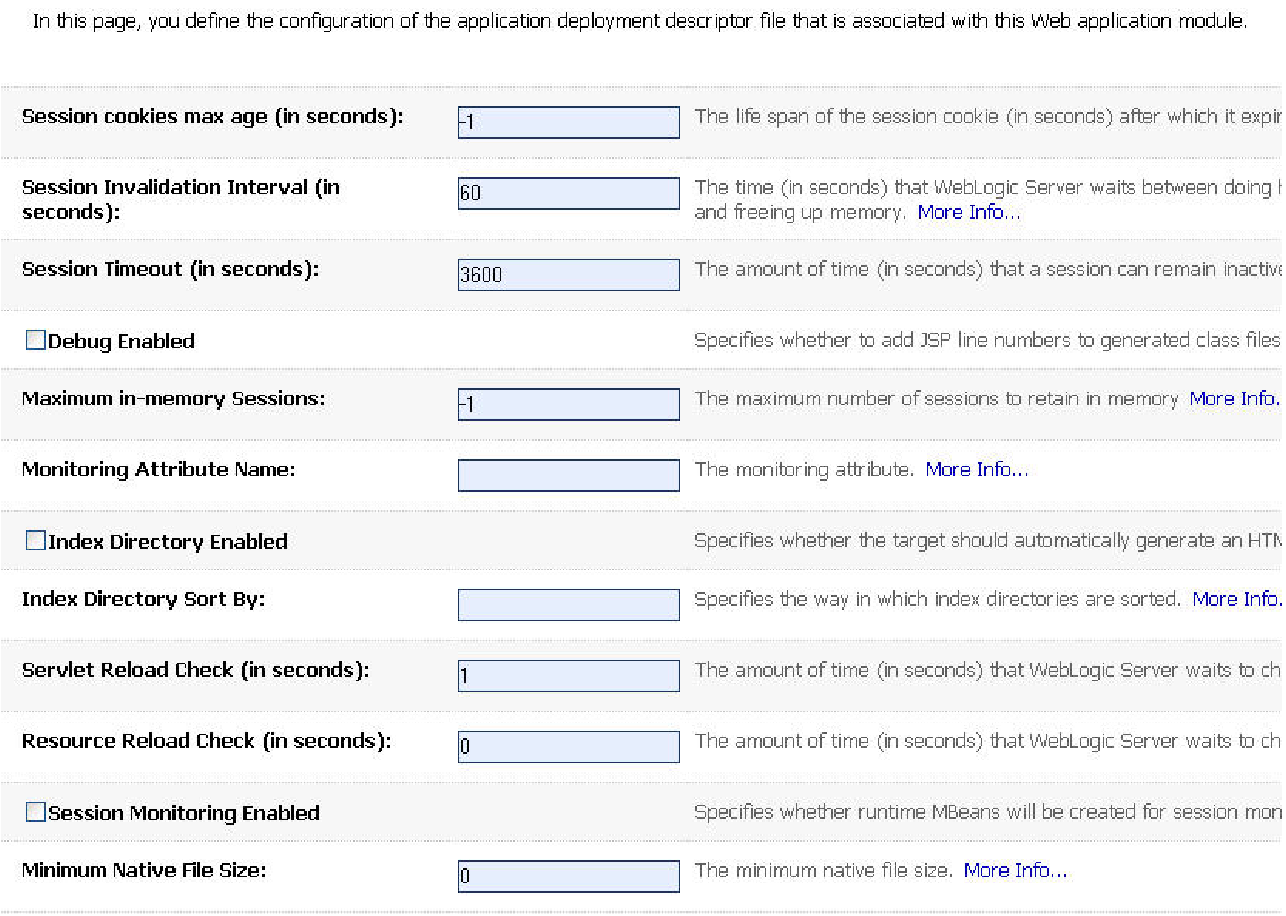
This page is used to stop and start a Web application.

Module

☐	Name
☑	⊟ benefits
	⊟ Web Services
	None to display

DD Editing

- Some deployment descriptor elements are editable via the console

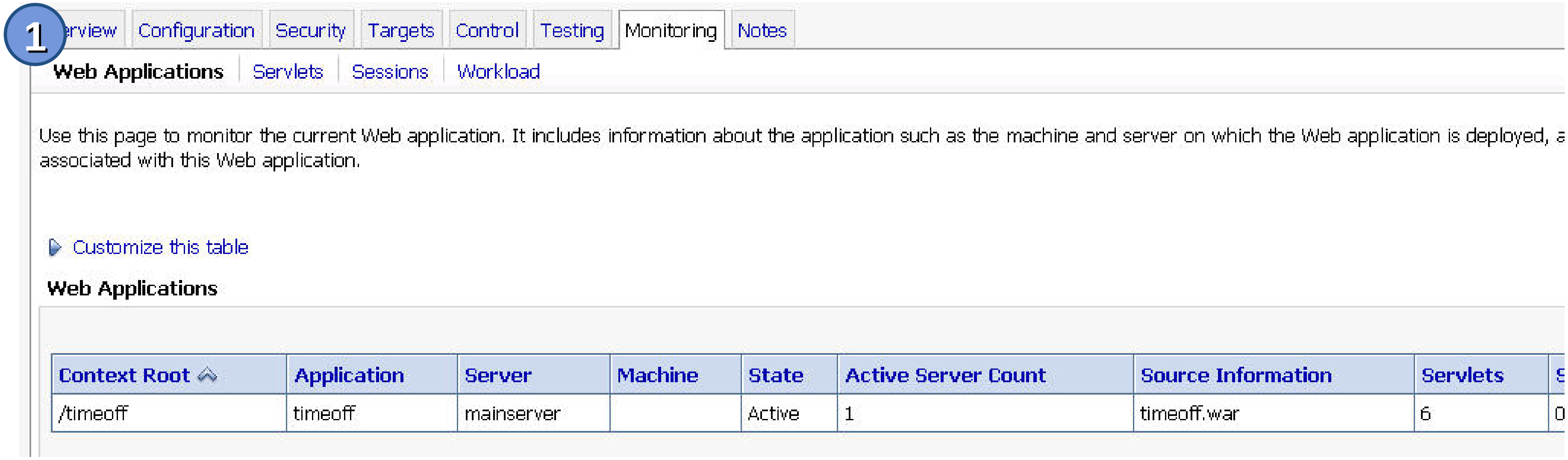


In this page, you define the configuration of the application deployment descriptor file that is associated with this Web application module.

Session cookies max age (in seconds):	-1	The life span of the session cookie (in seconds) after which it expir
Session Invalidation Interval (in seconds):	60	The time (in seconds) that WebLogic Server waits between doing h and freeing up memory. More Info...
Session Timeout (in seconds):	3600	The amount of time (in seconds) that a session can remain inactive
□ Debug Enabled		Specifies whether to add JSP line numbers to generated class files
Maximum in-memory Sessions:	-1	The maximum number of sessions to retain in memory More Info.
Monitoring Attribute Name:		The monitoring attribute. More Info...
□ Index Directory Enabled		Specifies whether the target should automatically generate an HTM
Index Directory Sort By:		Specifies the way in which index directories are sorted. More Info.
Servlet Reload Check (in seconds):	1	The amount of time (in seconds) that WebLogic Server waits to ch
Resource Reload Check (in seconds):	0	The amount of time (in seconds) that WebLogic Server waits to ch
□ Session Monitoring Enabled		Specifies whether runtime MBeans will be created for session mon
Minimum Native File Size:	0	The minimum native file size. More Info...

© 2006 BEA Systems, Inc.

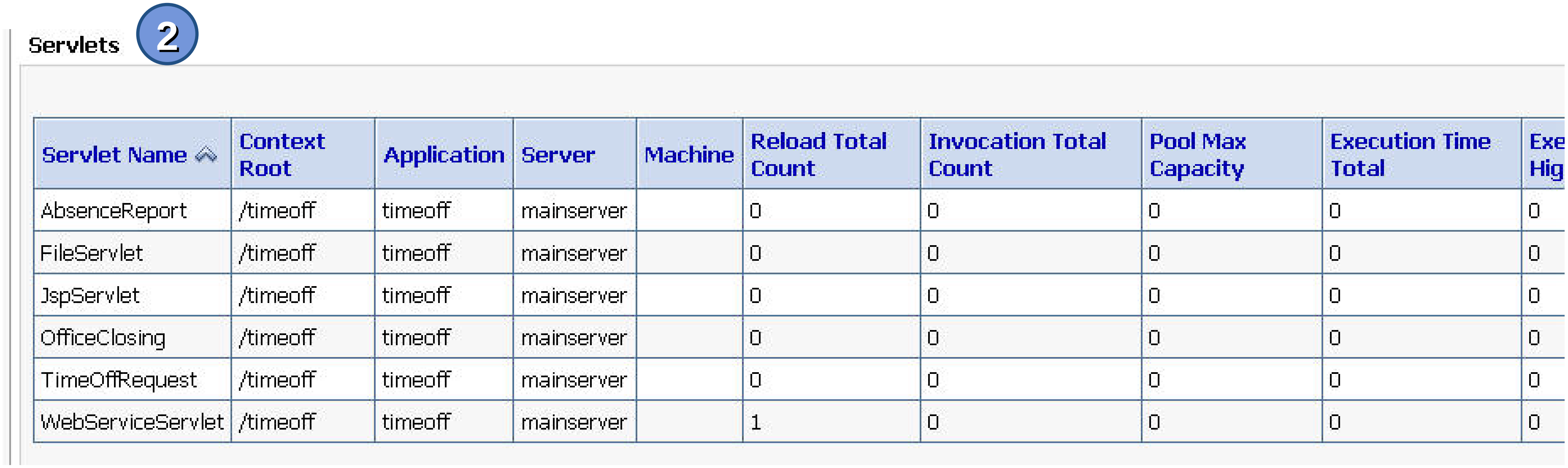
Application Monitoring



Use this page to monitor the current Web application. It includes information about the application such as the machine and server on which the Web application is deployed, a associated with this Web application.

Web Applications

Context Root	Application	Server	Machine	State	Active Server Count	Source Information	Servlets	S
/timeoff	timeoff	mainserver		Active	1	timeoff.war	6	0



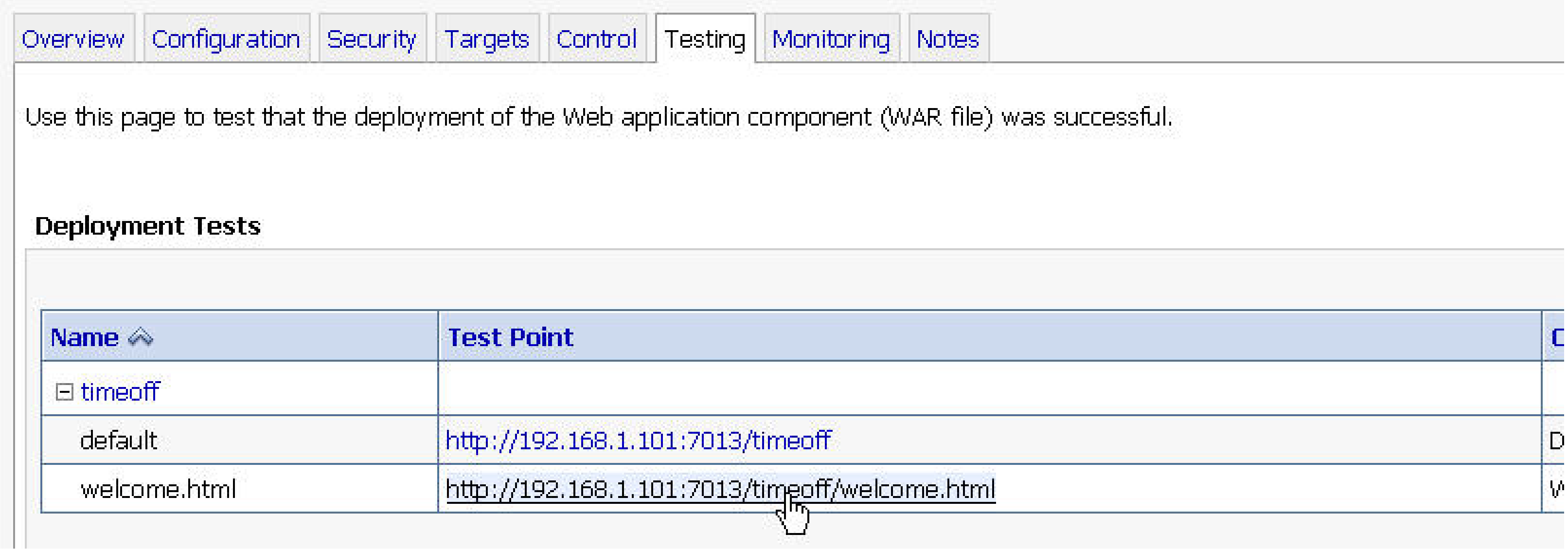
Servlets

Servlet Name	Context Root	Application	Server	Machine	Reload Total Count	Invocation Total Count	Pool Max Capacity	Execution Time Total	Exe Hig
AbsenceReport	/timeoff	timeoff	mainserver		0	0	0	0	0
FileServlet	/timeoff	timeoff	mainserver		0	0	0	0	0
JspServlet	/timeoff	timeoff	mainserver		0	0	0	0	0
OfficeClosing	/timeoff	timeoff	mainserver		0	0	0	0	0
TimeOffRequest	/timeoff	timeoff	mainserver		0	0	0	0	0
WebServiceServlet	/timeoff	timeoff	mainserver		1	0	0	0	0

© 2006 BEA Systems, Inc.

Application Testing

- You can test a deployed application using the administration console.



Use this page to test that the deployment of the Web application component (WAR file) was successful.

Deployment Tests

Name	Test Point	C
⊟ timeoff		
default	http://192.168.1.101:7013/timeoff	D
welcome.html	http://192.168.1.101:7013/timeoff/welcome.html	W

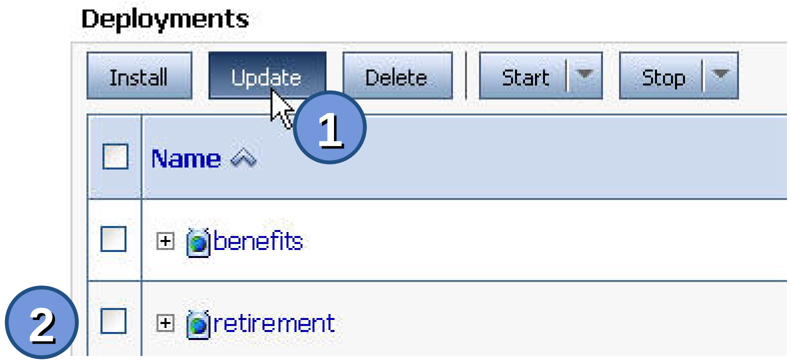
© 2006 BEA Systems, Inc.

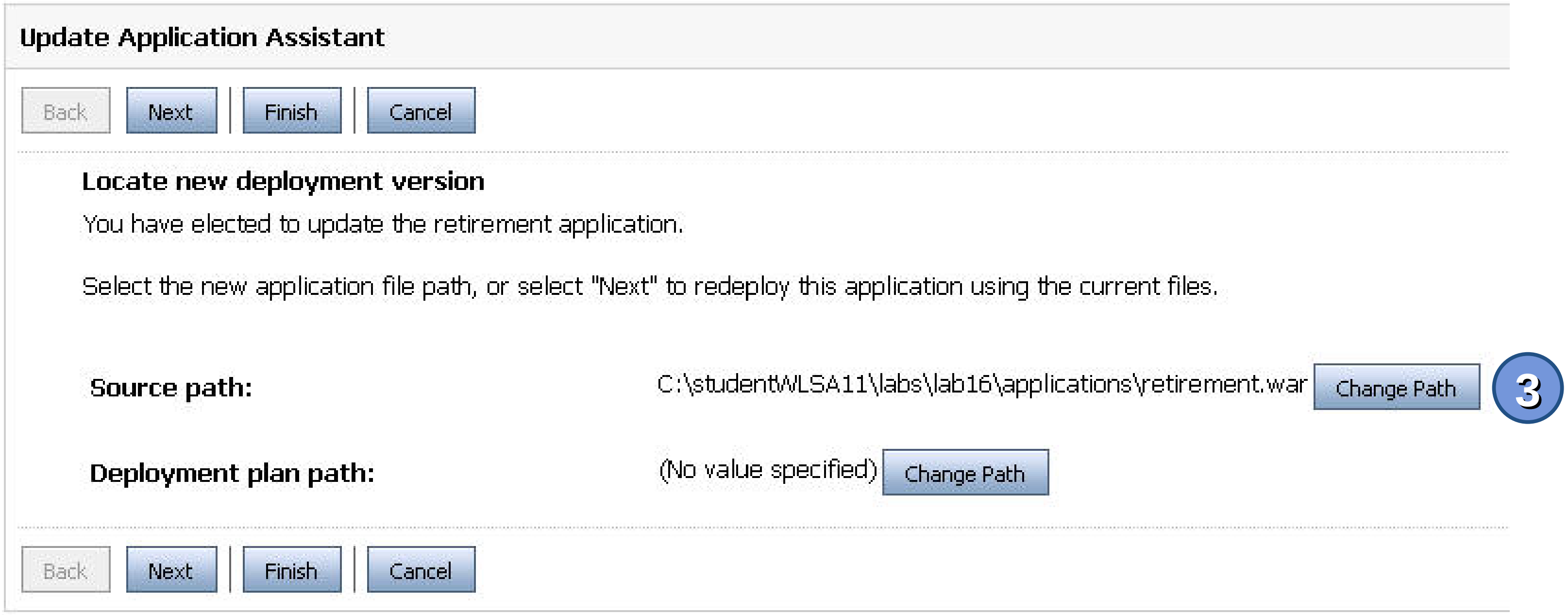
Application Update and Delete…

- Using the console, applications can be *updated (redeployed)* after configuration or component changes, or *deleted (undeployed).*
- All concurrent deployment activity is tracked by the Administration server in a series of tasks:
  - Task progress and outcome can be queried for each application
  - Reasons for failure are logged

© 2006 BEA Systems, Inc.

…Application Update and Delete…





Update Application Assistant

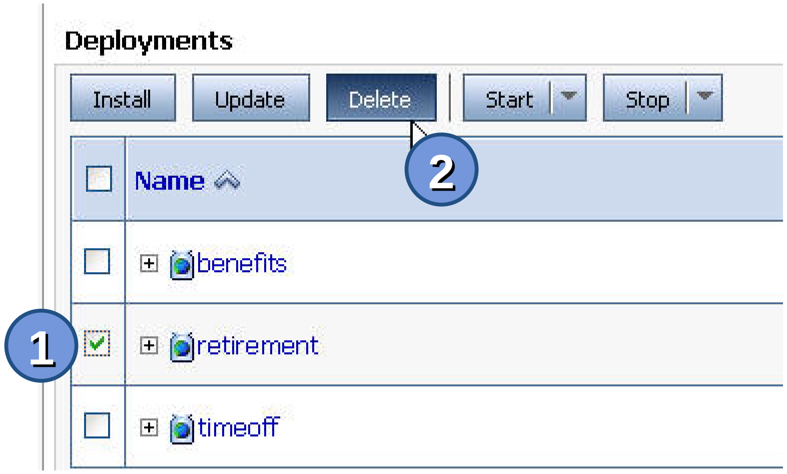
Locate new deployment version

You have elected to update the retirement application.

Select the new application file path, or select "Next" to redeploy this application using the current files.

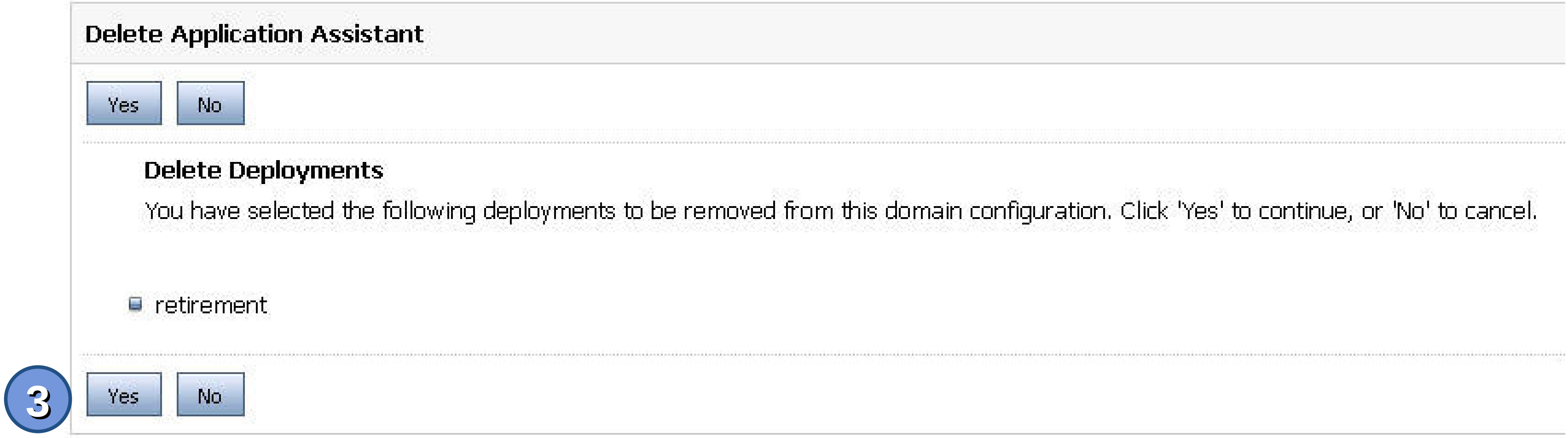
Source path:	C:\studentWLSA11\labs\lab16\applications\retirement.war
Deployment plan path:	(No value specified)

…Application Update and Delete



Deployments

□	Name
□	⊞ benefits
☑	⊞ retirement
□	⊞ timeoff



Delete Application Assistant

Delete Deployments

You have selected the following deployments to be removed from this domain configuration. Click 'Yes' to continue, or 'No' to cancel.

- retirement

Command-Line Deployment

- The `weblogic.Deployer` utility allows you to do deployment operations similar to those available in the console.
- `weblogic.Deployer` actions can be also be scripted with the ant task `wldeploy`

`weblogic.Deployer` Syntax:

```
% java weblogic.Deployer [options]
       [-deploy|-undeploy|-redeploy|-start|-stop|-listapps]
              [file(s)]
```

weblogic.Deployer Examples…

weblogic.Deployer Examples:

To deploy a new application:

```
java weblogic.Deployer -adminurl t3://localhost:7001
     -username system -password weblogic
     -name app -source /myapp/app.ear
     -targets server1,server2 -deploy
```

To redeploy an application:

```
java weblogic.Deployer -adminurl t3://localhost:7001
     -username system -password weblogic -name app –redeploy
```

To redeploy part of an application:

```
java weblogic.Deployer -adminurl t3://localhost:7001
     -username system -password weblogic
     -targets server1,server2 -redeploy jsps/*.jsp
```

To undeploy an application:

```
java weblogic.Deployer -adminurl t3://localhost:7001
     -username system -password weblogic -undeploy
     -name myapp –targets server1,server2
```

…weblogic.Deployer Examples

More weblogic.Deployer Examples:

To list all deployed applications:

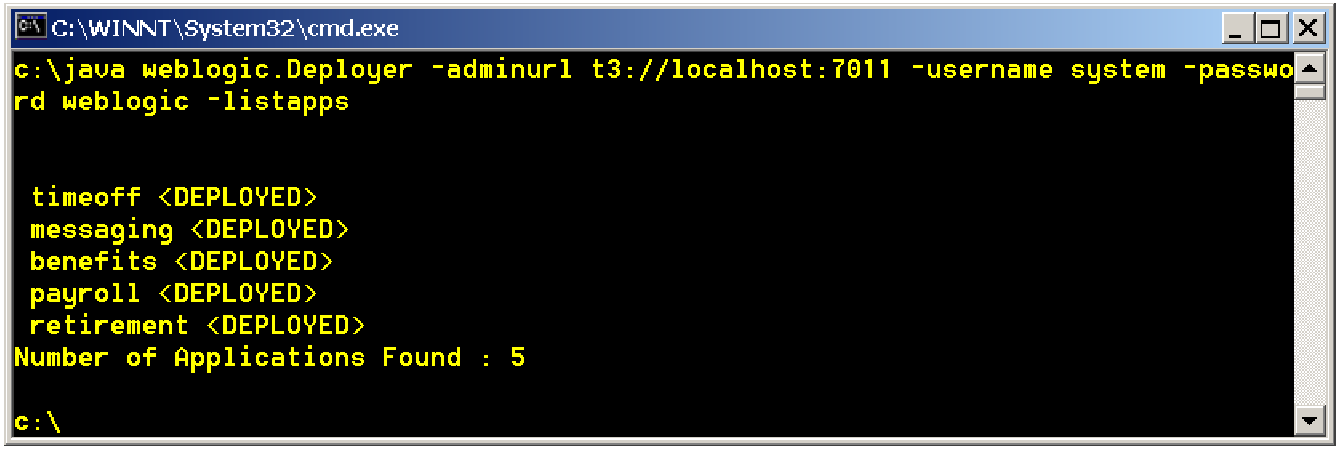
```
java weblogic.Deployer -adminurl t3://localhost:7001
     -username system -password weblogic -listapps
```

To list all deployment tasks:

```
java weblogic.Deployer -adminurl http://localhost:7001
     -username system –password weblogic -listtask
```

To cancel a deployment task:

```
java weblogic.Deployer -adminurl http://localhost:7001
     -username system –password weblogic -cancel -id tag
```



```
c:\java weblogic.Deployer -adminurl t3://localhost:7011 -username system -passwo
rd weblogic -listapps


 timeoff <DEPLOYED>
 messaging <DEPLOYED>
 benefits <DEPLOYED>
 payroll <DEPLOYED>
 retirement <DEPLOYED>
Number of Applications Found : 5

c:\
```

Deploying Applications with WLST

- WLST provides a number of deployment commands
- You can use these commands to:
  - Deploy, undeploy, and redeploy applications and standalone modules to a WebLogic Server instance
  - Update an existing deployment plan
  - Start and stop a deployed application

© 2006 BEA Systems, Inc.

Deploying an Application with WLST

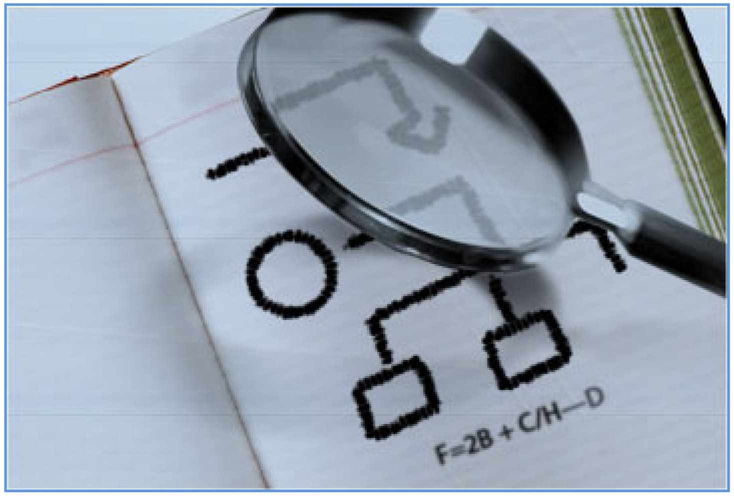
Deploy an application (deployapp.py):

```
#                                                                   #
# WLST script for Deploying J2EE Application                         #
#                                                                   #

# Connect to the server
print ’Connecting to server  .... ’
connect(’system’,’weblogic’,’t3://localhost:7001’)

appname = "mbeanlister"
applocation = "c:/domains/dizzyworld/apps/mbeanlister"

# Start deploy
print ’Deploying application ’ + appname
deploy(appname, applocation, targets=’myserver’,
   planPath=’c:/myapps/plan/plan.xml’)
print ’Done Deploying the application ’+ appname
exit()
```

Section Review

In this section we discussed:

- ✓ Auto-deployment
- ✓ Console deployment
- ✓ Command-line deployment



Exercise

Deploying & Undeploying Web Applications

- In this lab you will learn about deploying and Undeploying Web Applications.
- Ask the instructor for any clarification.
- The instructor will determine the stop time.

Lab Exercise

© 2006 BEA Systems, Inc.

Road Map

1. Web Servers
2. Web Applications
3. EJB Applications
4. Enterprise Applications
5. Deployment
6. **Advanced Deployment**
   - Deployment Plans
   - Staged Deployment
   - Side-by-Side Deployment

© 2006 BEA Systems, Inc.

What Is a Deployment Plan?

- It is an optional XML document that resides outside an application archive.
- It configures an application for deployment to a specific WLS environment.
- It is created and owned by Administrators or developers for a particular environment.

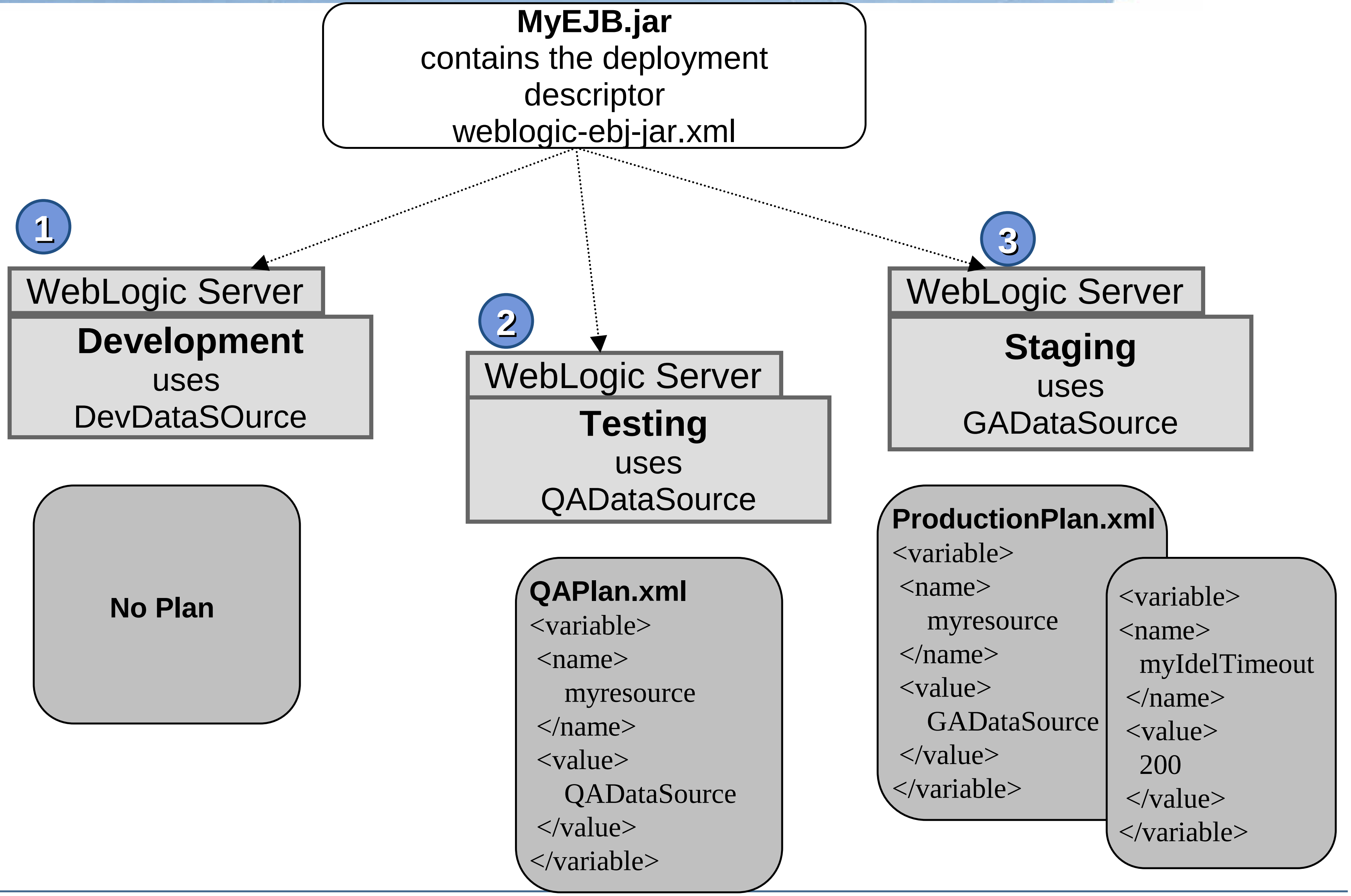
© 2006 BEA Systems, Inc.

Advantages of Deployment Plan

- Works by setting/overriding deployment property values defined in application’s WLS deployment descriptor.
- Helps easily modify an application’s WLS configuration for deployment into different multiple WLS environments without modifying the deployment descriptor files included in the application archive.
- Enables an application to be deployed to multiple domains or to multiple target servers and clusters that have different configuration within the same domain.

© 2006 BEA Systems, Inc.

Configuring an Application for Multiple Deployment Environments



MyEJB.jar
contains the deployment descriptor
weblogic-ebj-jar.xml

1

WebLogic Server
Development
uses
DevDataSOurce

No Plan

2

WebLogic Server
Testing
uses
QADataSource

QAPlan.xml

```
<variable>
 <name>
    myresource
 </name>
 <value>
    QADataSource
 </value>
</variable>
```

3

WebLogic Server
Staging
uses
GADataSource

ProductionPlan.xml

```
<variable>
 <name>
    myresource
 </name>
 <value>
    GADataSource
 </value>
</variable>
```

```
<variable>
<name>
  myIdelTimeout
 </name>
 <value>
  200
 </value>
</variable>
```

© 2006 BEA Systems, Inc.

Sample Deployment Plan

```
<deployment-plan xmlns="http://www.bea.com/ns/weblogic/90">
        <application-name>sample_root</application-name>
          <variable-definition>
          <variable>      <name>SessionDescriptor_InvalidationIntervalSecs_11029744771850</name>
            <value>80</value>
          </variable>
          <variable>
            <name>SessionDescriptor_TimeoutSecs_11029744772011</name>
            <value>8000</value>
          </variable>
          </variable-definition>
        <module-override>
          <module-name>jspExpressionEar.ear</module-name>
          <module-type>ear</module-type>
          <module-descriptor external="false">
            <root-element>weblogic-application</root-element>
            <uri>META-INF/weblogic-application.xml</uri>
          </module-descriptor>
          <module-descriptor external="false">
            <root-element>application</root-element>
            <uri>META-INF/application.xml</uri>
          </module-descriptor>
        </module-override>
        <module-override>
          <module-name>jspExpressionWar</module-name>
          <module-type>war</module-type>
          <module-descriptor external="false">

      …
```

Creating a Deployment Plan

- Tools for creating a deployment plan
  - weblogic.PlanGenerator
  - Administration Console
- Goals for creating a deployment plan
  - To expose the external resource requirements of the application as variables in the deployment plan
  - To expose additional configurable properties, such as tuning parameters as variables in the deployment plan

© 2006 BEA Systems, Inc.

weblogic.PlanGenerator

- Java-based deployment configuration tool.
- It is primarily intended for developers who want to export portions of a WebLogic Server deployment configuration into an XML deployment plan.
- Enables you to generate a basic WebLogic Server configuration for applications that have only J2EE deployment descriptors.

© 2006 BEA Systems, Inc.

Using the Admin Console to Generate a Deployment Plan

- Administration Console automatically generates or updates the deployment plan.
- Generating a deployment plan using the Administration Console involves the following steps:
  - Preparing the deployment files
  - Installing the application archive
  - Saving configuration changes to a deployment plan

© 2006 BEA Systems, Inc.

Using an Existing Deployment Plan to Configure an Application

1. Prepare application.
2. Place the existing deployment plan in the plan subdirectory of the application root.
3. Install the application.
4. Administration console validates deployment plan configuration against the target servers and clusters selected during installation.
5. Use Administration console or the weblogic.Deployer utility to identify the application and plan to use for deployment.

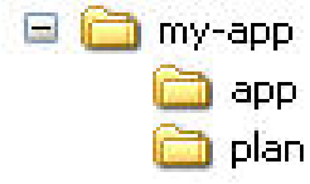
© 2006 BEA Systems, Inc.

Directory Structure for Easier Production Deployment…

- The application directory structure separates generated configuration files from core application files.
- This allows configuration files to be easily changed or replaced without disturbing the application itself.
- Applications can be deployed simply by specifying the installation root.

© 2006 BEA Systems, Inc.

…Directory Structure for Easier Production Deployment

Directory Tree	Description
 my-app	Application root
app	Application deployment files (archive or exploded)
plan	`plan.xml`

- This allows deployment configuration files to be located in a well-known location.

© 2006 BEA Systems, Inc.

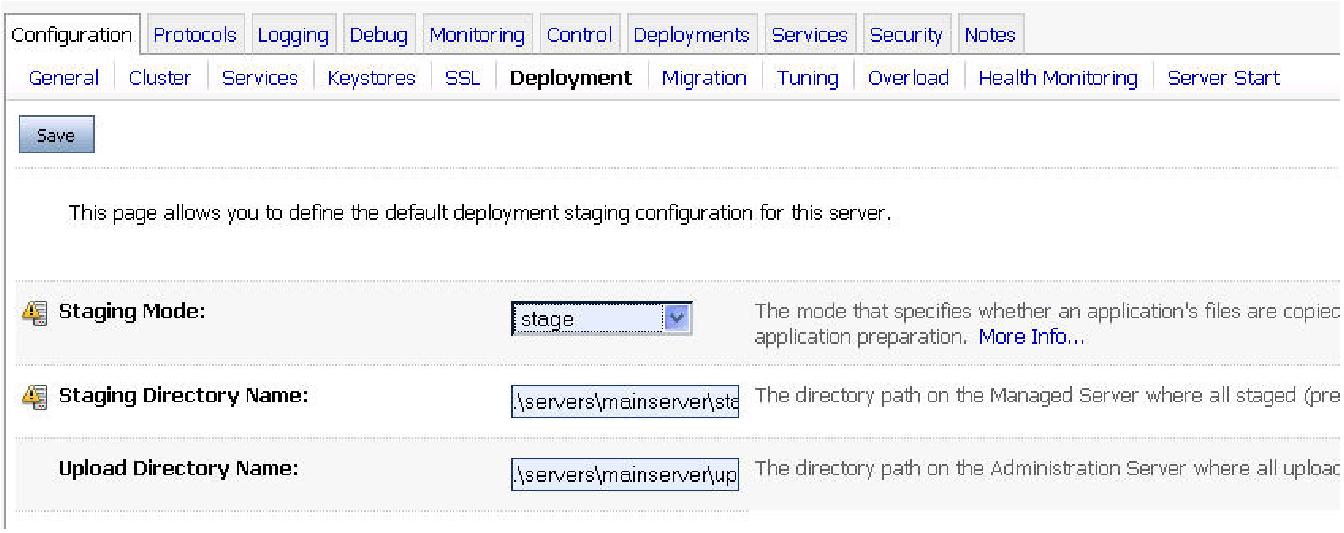
Sanity Checking in Production Without Disruption to Clients

- Using Administration mode, administrators can deploy an application into a production environment without exposing the application to external clients.
- Access to the application is restricted to a configured Administration channel.
- A final (“sanity”) check can be performed on the application directly in the production environment without disruption to clients.

© 2006 BEA Systems, Inc.

Staged Deployment

- Deployment can be configured per server or for each application as:
  - *staged* (default)—files copied to preconfigured staging directory for preparation and activation
  - *no-stage*—files deployed from static location
  - *external_stage*—files copied by user or third-party tool prior to deployment



This page allows you to define the default deployment staging configuration for this server.

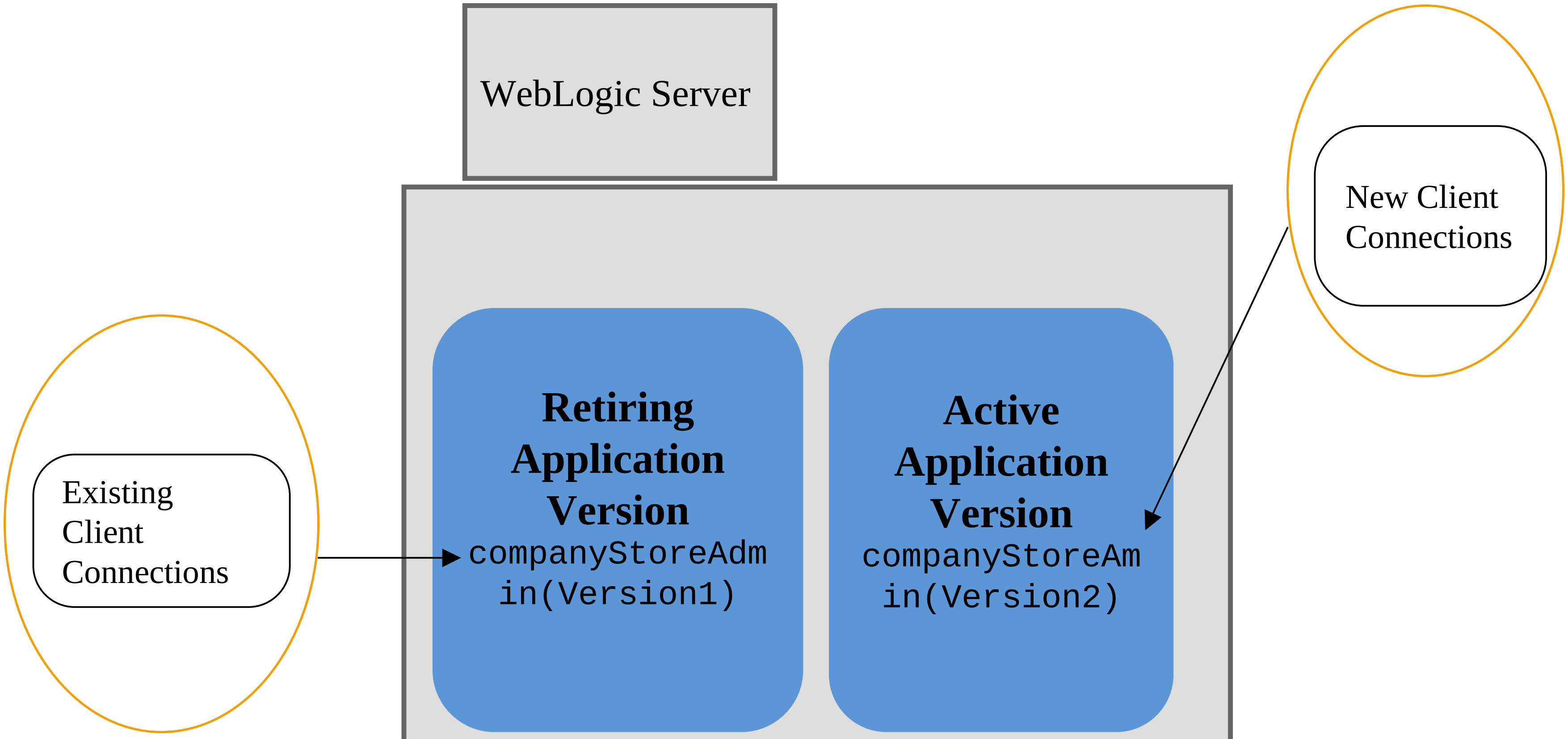
Staging Mode:	stage	The mode that specifies whether an application's files are copiec application preparation. More Info...
Staging Directory Name:	.\servers\mainserver\sta	The directory path on the Managed Server where all staged (pre
Upload Directory Name:	.\servers\mainserver\up	The directory path on the Administration Server where all uploac

Side-by-Side Deployment…

- A revised version of a production application can be redeployed alongside the older version:
  - Without affecting the existing clients to the application
  - Without interrupting the availability of the application to the new client request
- WebLogic Server automatically manages client connections so that:
  - Existing clients continue to use the older application
  - New client requests are directed to the newer application
- The older version is undeployed after all current clients complete their work.

© 2006 BEA Systems, Inc.

...Side-by-Side Deployment...


WebLogic Server
New Client Connections
Existing Client Connections
Retiring Application Version
companyStoreAdmin(Version1)
Active Application Version
companyStoreAmin(Version2)

… Side-by-Side Deployment

- To support the production redeployment strategy, WebLogic Server now recognizes a unique version string entry in the Enterprise `MANIFEST` file.
- When a redeployment operation is requested, WebLogic Server checks the version string to determine whether to deploy a new version of the application.
- Side-by-Side deployment is performed automatically if:
  - An application supports production redeployment
  - Its deployment configuration is updated with changes to resource bindings
- This occurs even if no version string is specified in the application's manifest file.

© 2006 BEA Systems, Inc.

Advantages of Side-by-Side Deployment

Saves the trouble of:

1. Scheduling application downtime
2. Setting up redundant servers to host new application versions
3. Managing client access to multiple application versions manually
4. Retiring older versions of an application manually

© 2006 BEA Systems, Inc.

Requirements and Restrictions for Side-by-Side Deployment

Production redeployment is supported for:

1. Stand-alone Web Application (WAR) modules.
2. Enterprise Applications (EARs) whose client access the application via a Web Application (HTTP)

Production redeployment is not supported for:

1. EJB or RAR modules
2. Stand-alone or embedded Web Service modules
3. Applications that use JTS drivers
4. Applications that obtain JDBC data sources via the DriverManager API instead of using the JNDI lookup
5. Applications that include EJB 1.1 container-managed persistence (CMP) EJBs

© 2006 BEA Systems, Inc.

Requirements and Restrictions for Side-by-Side Deployment

1. A deployed application must specify a version number.
2. WLS can host a maximum of two different versions of an application at one time.
3. On redeploying a new version of an application the followings features can not change:
    1. Deployment targets
    2. Security model
    3. Persistent store settings

© 2006 BEA Systems, Inc.

Redeploying a New Application Version

1. Verify only one version of the application is currently deployed.
2. Verify the MANIFEST.MF files to ensure both applications have different versions.
3. Copy new version into a suitable directory.
4. Redeploy the new application version and specify updated deployment files.
5. Verify both versions are deployed and new requests are being sent to the new version.

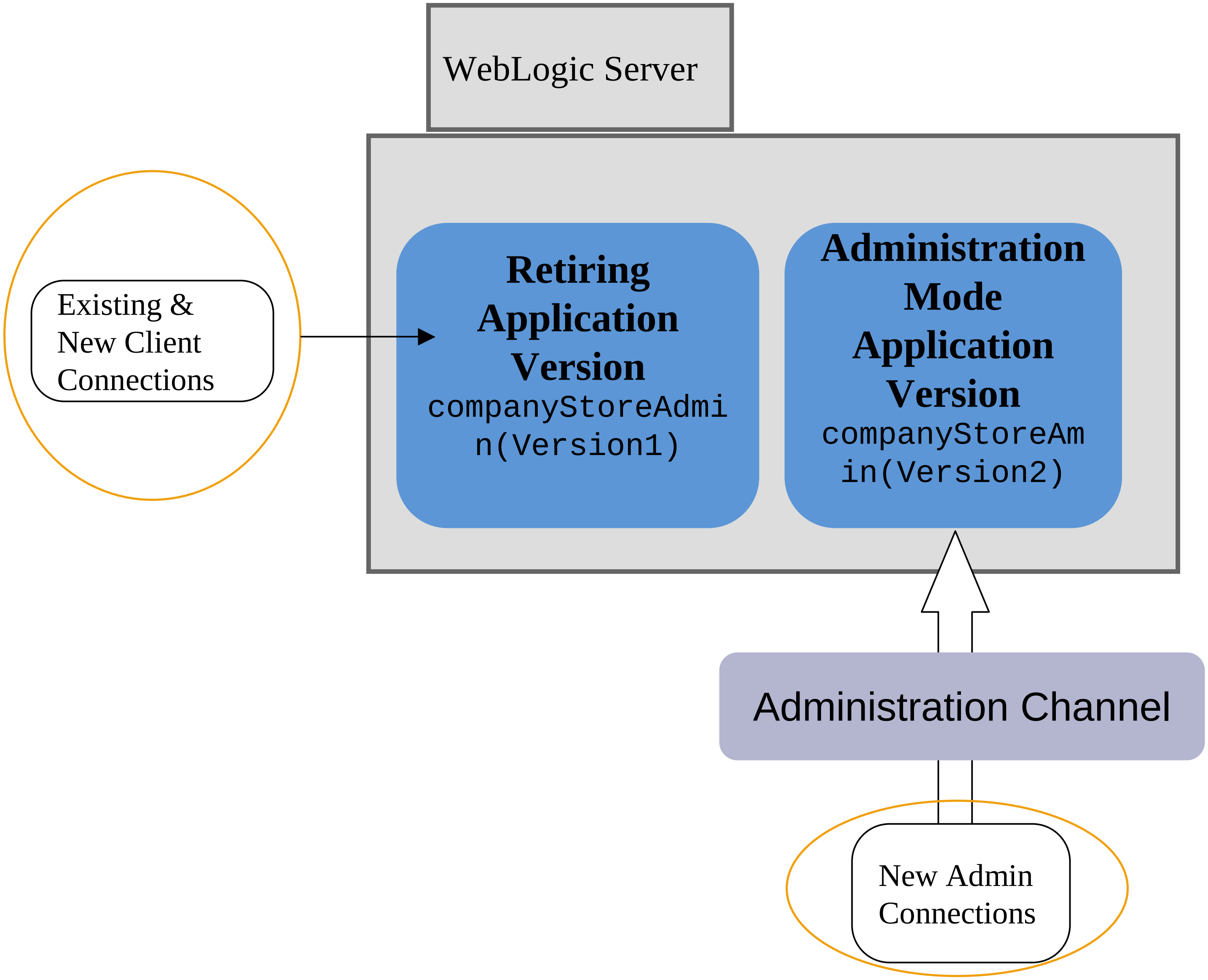
© 2006 BEA Systems, Inc.

Redeploy vs. Distribute

- Distributing is an alternative to deploying an application.
  - Distributing an application prepares it for deployment by copying its files to all target servers and validating it.
  - You can start a distributed application in Administration mode. Access to the application is then restricted to a configured Administration channel.
- Distributing a new version of the application makes it available for testing prior to being released for general consumption.
- Redeploying a new version of an application will place the application immediately into use and will make it available to new client requests.

© 2006 BEA Systems, Inc.

Distributing a New Version of Production Application



WebLogic Server

Existing & New Client Connections

Retiring Application Version

`companyStoreAdmin(Version1)`

Administration Mode Application Version

`companyStoreAmin(Version2)`

Administration Channel

New Admin Connections

© 2006 BEA Systems, Inc.

Distributing a New Application Version

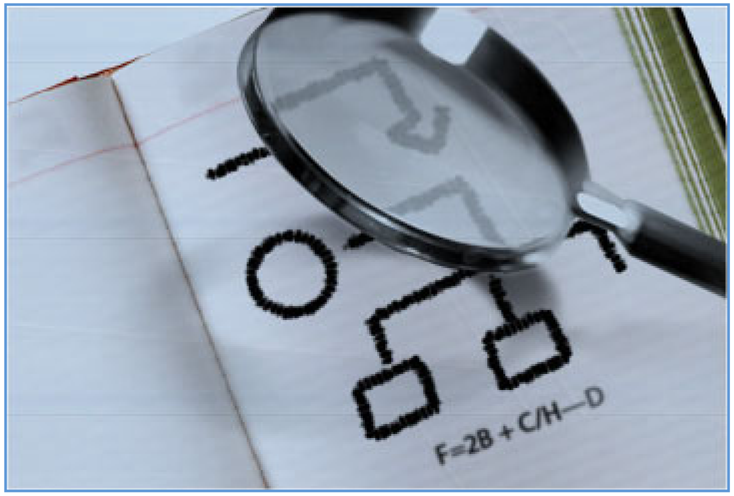
1. Use the weblogic.Deployer `–distribute` command
2. Once the application is distributed, start the application in Administration mode
3. Test the application
4. When ready, start the application (without using –adminmode)
5. Optionally set a retirement timeout for the older version of the application.

© 2006 BEA Systems, Inc.

Section Review

In this section we discussed:

- ✓ Deployment Plans
- ✓ Deployment staging
- ✓ Side-by-Side Deployment



© 2006 BEA Systems, Inc.

Exercise

Using a Deployment Plan

- In this lab you will learn about using a deployment plan for deploying applications.
- Ask the instructor for any clarification.
- The instructor will determine the stop time.

Lab Exercise

© 2006 BEA Systems, Inc.

Exercise

Side-by-Side deployment

- In this lab you will learn about Side-by-Side deployment.
- Ask the instructor for any clarification.
- The instructor will determine the stop time.

Lab Exercise

Module Review

In this module we discussed:

- ✓ Web server and Web Application basics
- ✓ Packaging and deploying Web Applications
- ✓ Enterprise JavaBeans concepts
- ✓ EJB configuration
- ✓ Enterprise Application concepts

© 2006 BEA Systems, Inc.