10

Deployment Concepts

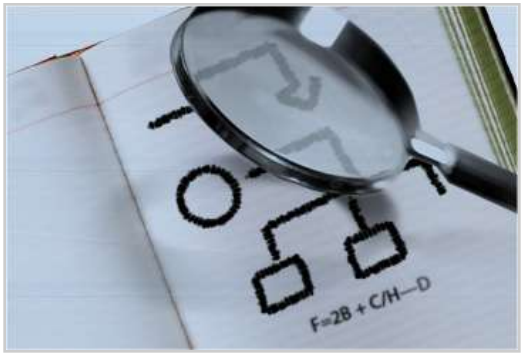
Objectives

After completing this lesson, you should be able to:

- Contrast autodeploy with manual deployment
- Configure and deploy Web applications via the Administration Console, command line, and WLST
- Configure deployment descriptors
- Test deployed applications
- Describe the role of Web servers
- Trace a typical Web interaction flow
- Contrast static and dynamic content and deployment
- Front-end deployed applications with a Web server

Road Map

- Deployment concepts
  - Autodeployment
  - Console deployment
  - Command-line deployment
- Developer deployment
- Front-end with a Web server



Overview of Deployment

Two views of deployment:

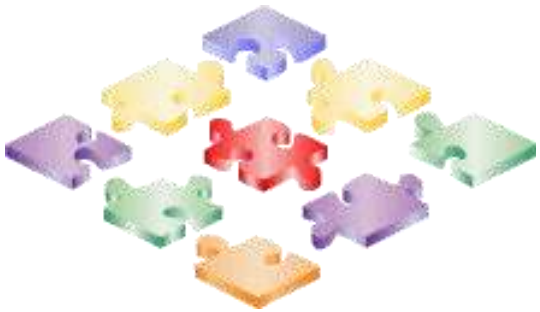
- Developers
  - Development environment
  - Single stand-alone machine
  - Deploy over and over again at will during the testing phase

- Administrators
  - Production environment
  - Multiple WebLogic Server instances or clusters
  - Deploy infrequently during maintenance schedules

What Is Deployed?

Deploy Java EE application in:

- Exploded form
  - Directory structure very important
  - Easier to update individual pieces
  - Harder to keep track of the whole collection



- Archive form
  - Similar to tar or zip, can be maintained with those tools
  - Type: .jar, .war, .ear, .rar, and so on
  - Contains code, metacode, descriptors (xml), directories

Deployment Process

Deploying an application involves the following tasks:

- **Preparing:** Choosing whether to package the application as an archived file or keep it in an exploded directory
- **Configuring:** Creating a deployment plan to maintain the configuration changes without changing the deployment descriptors
- **Deploying:** Targeting and distributing the application to servers in an Oracle WebLogic Server domain
  - Install or deploy
  - Update or redeploy
  - Delete or undeploy

Deployment Methods

- WLS supports three deployment methods:
  - Console deployment
  - Command-line deployment
  - Autodeployment
- Applications and EJBs can be deployed in an:
  - Archived file (`.ear`, `.war`, `.jar`)
  - Exploded (open) directory format

Deployment Tools

Several methods are available to deploy applications and shared libraries to the Oracle WebLogic Server, including:

- Administration Console
- WebLogic Scripting Tool (WLST)
- `weblogic.Deployer` Java class
- `wldeploy` Ant task
- Autodeployment folder

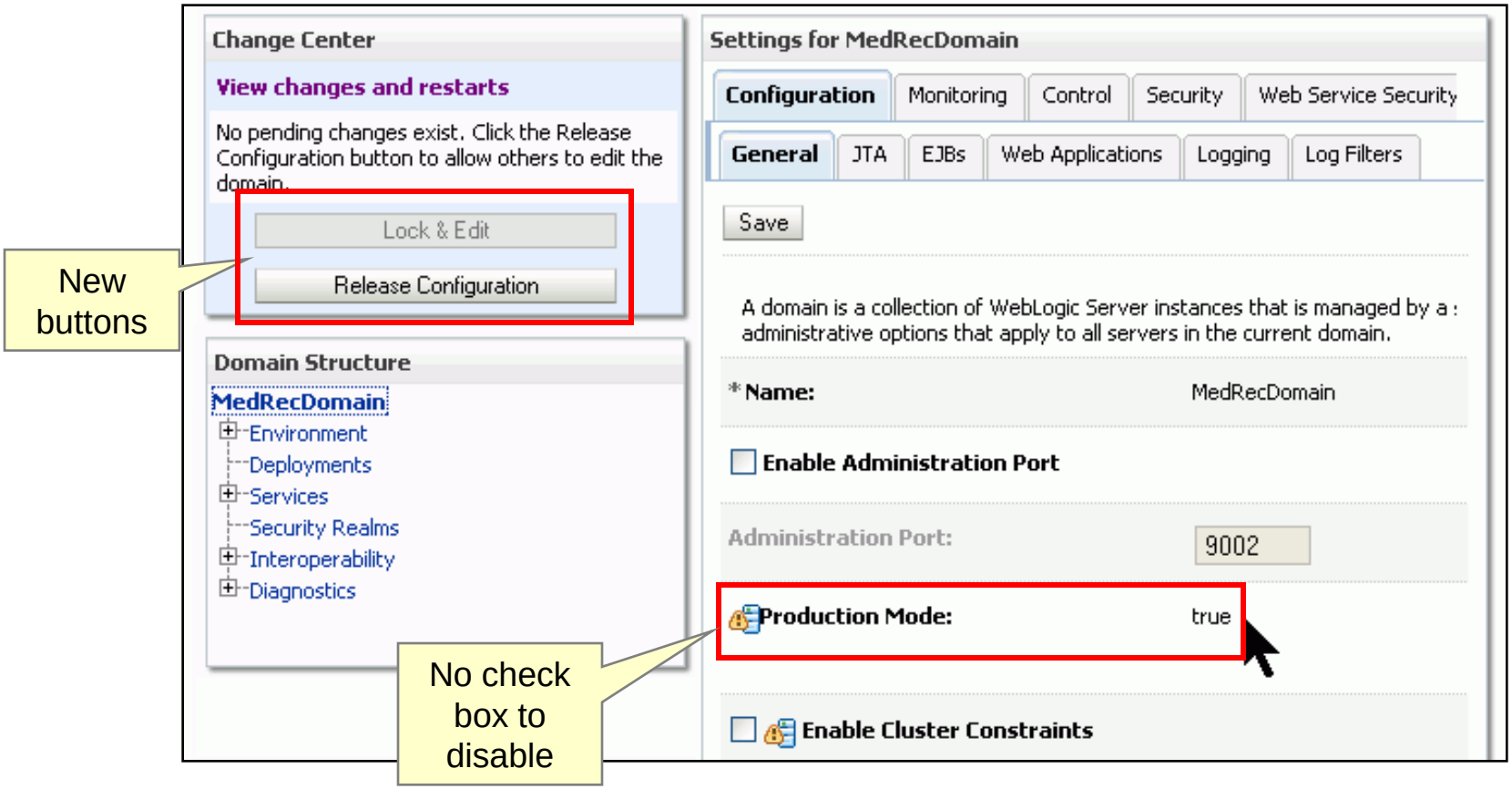
Console Deployment Method

Deploying with the console allows full administrator control:

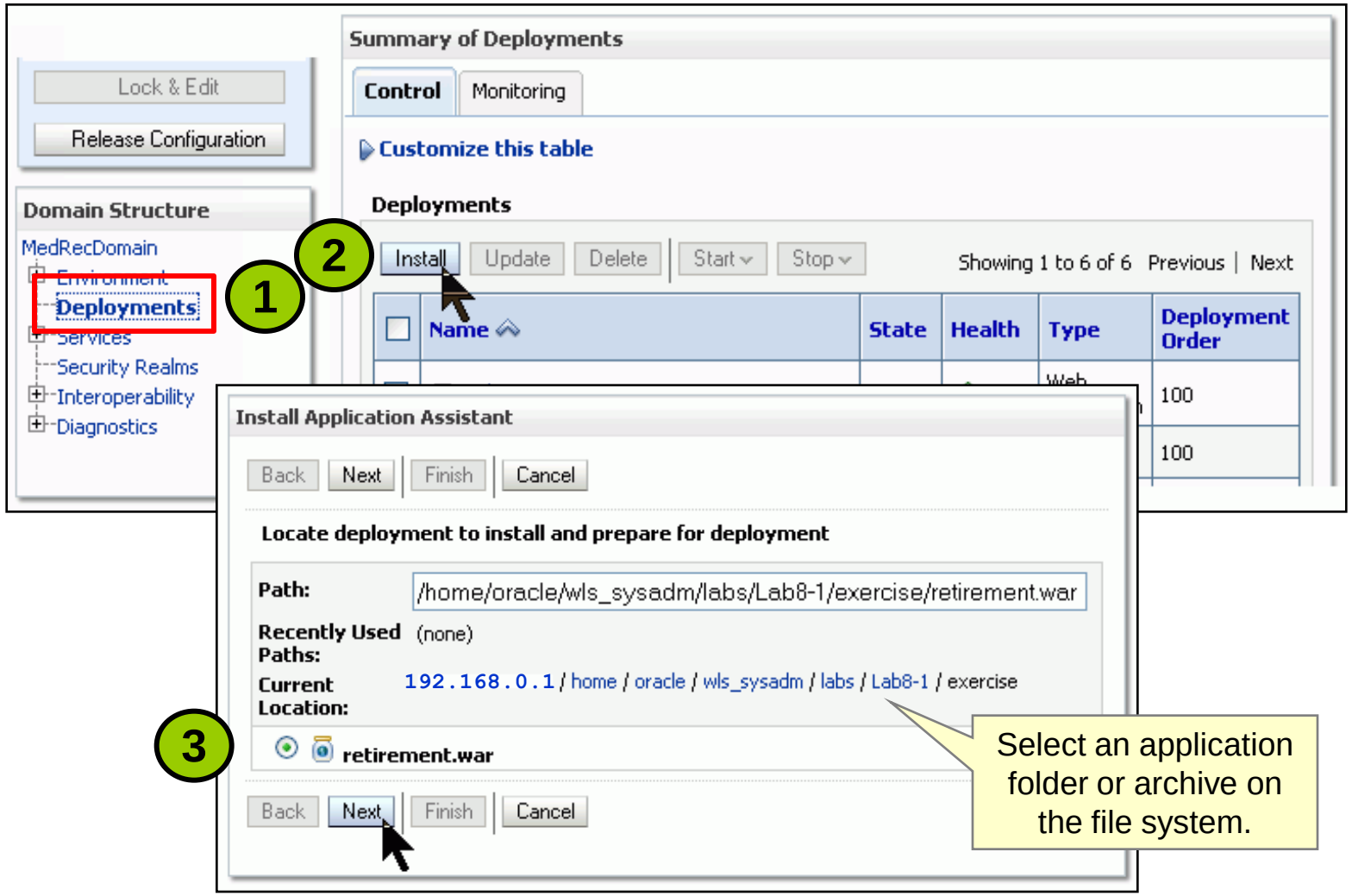
- Installation of an application from a location of your choice
- Manual configuration of the application name
- Targeting the application to individual servers or clusters, or both
- Configuring the application without targeting it
- Activating deployment when desired

Console Deployment Production Mode

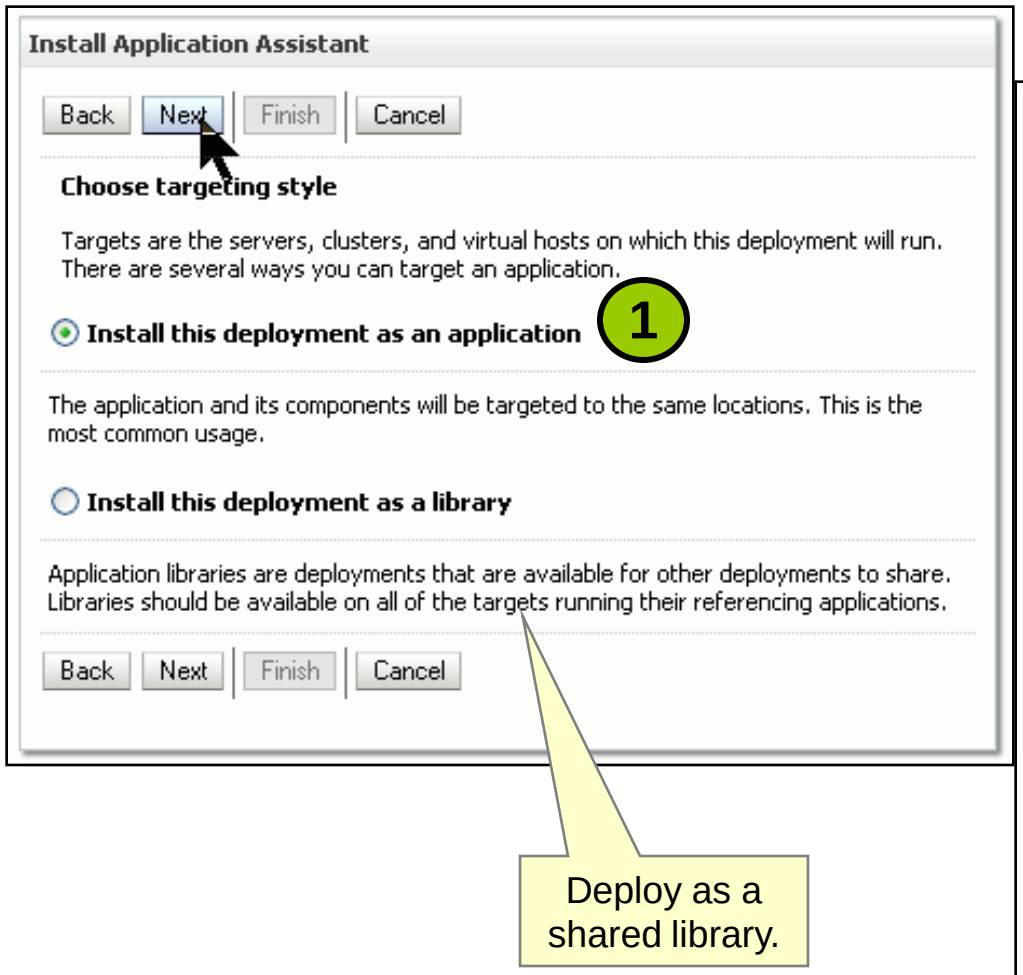
Best used with Production mode:



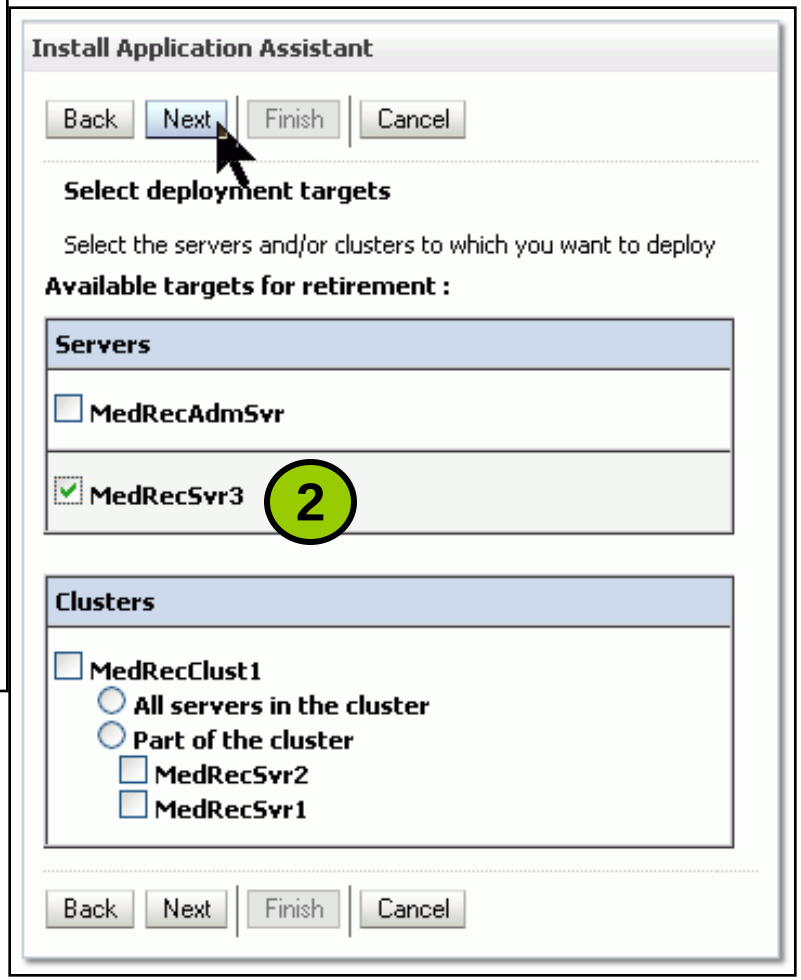
Preparing a New Application



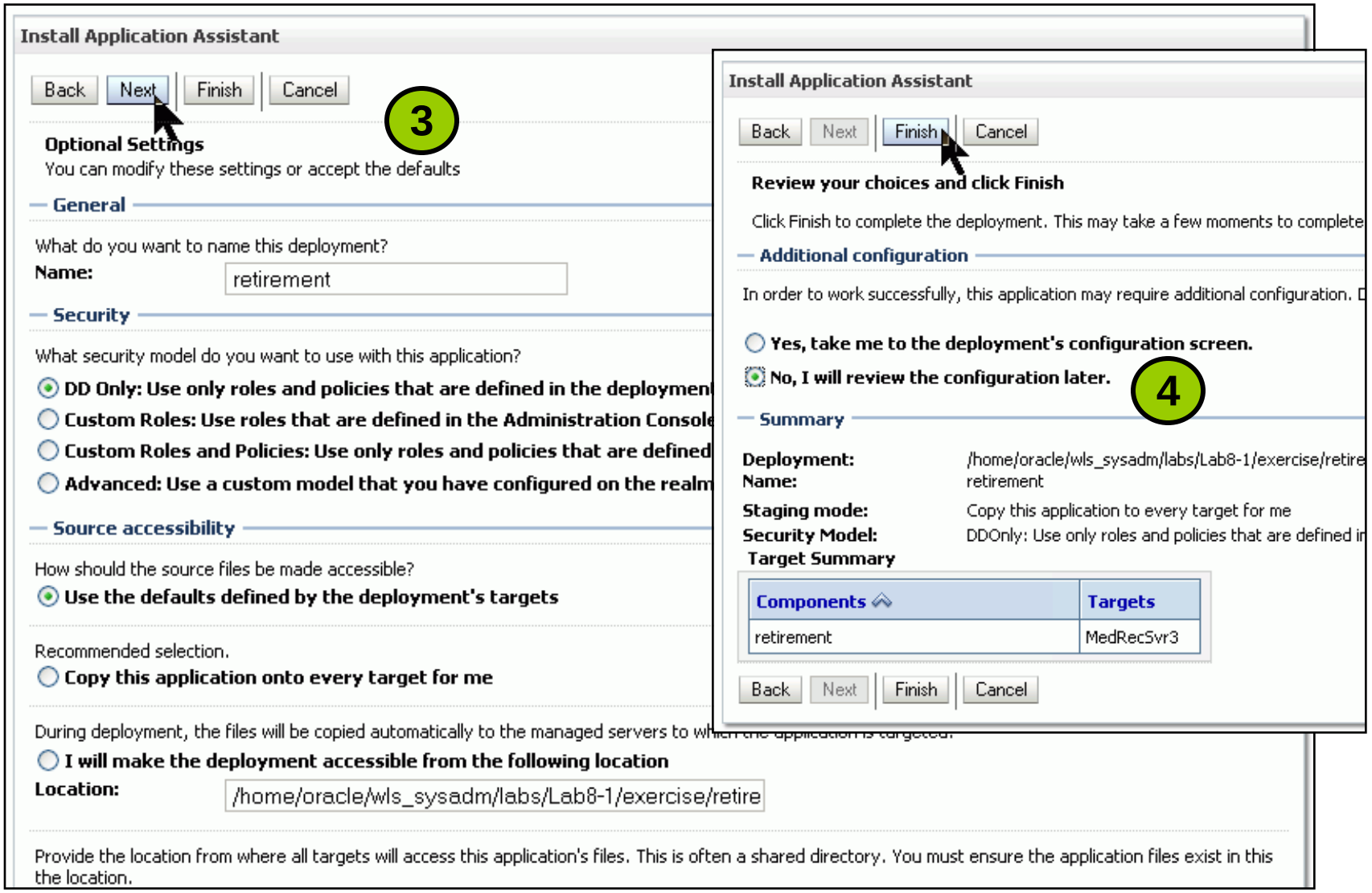
Preparing a New Application: Targeting



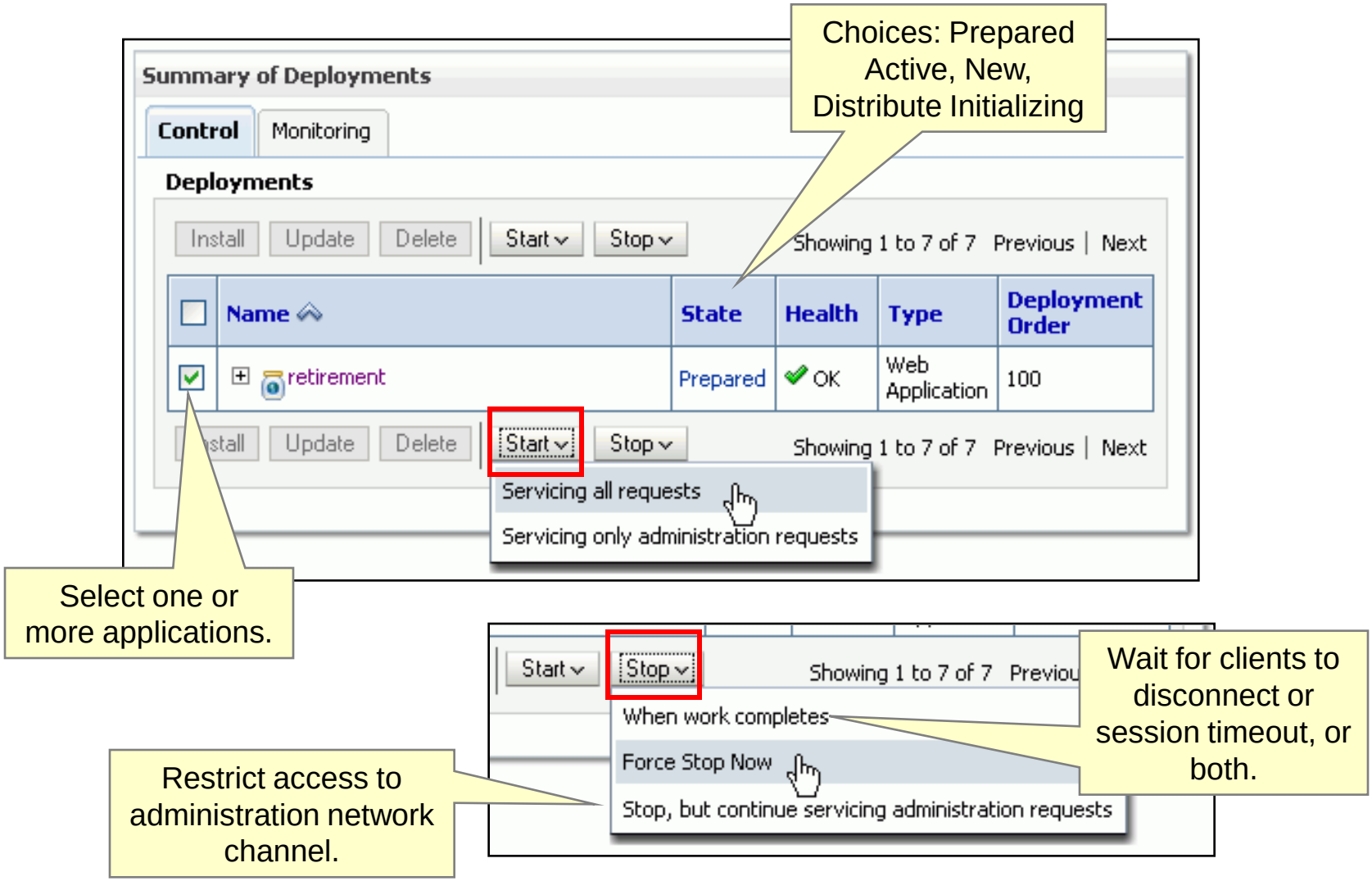
Deploy as a shared library.



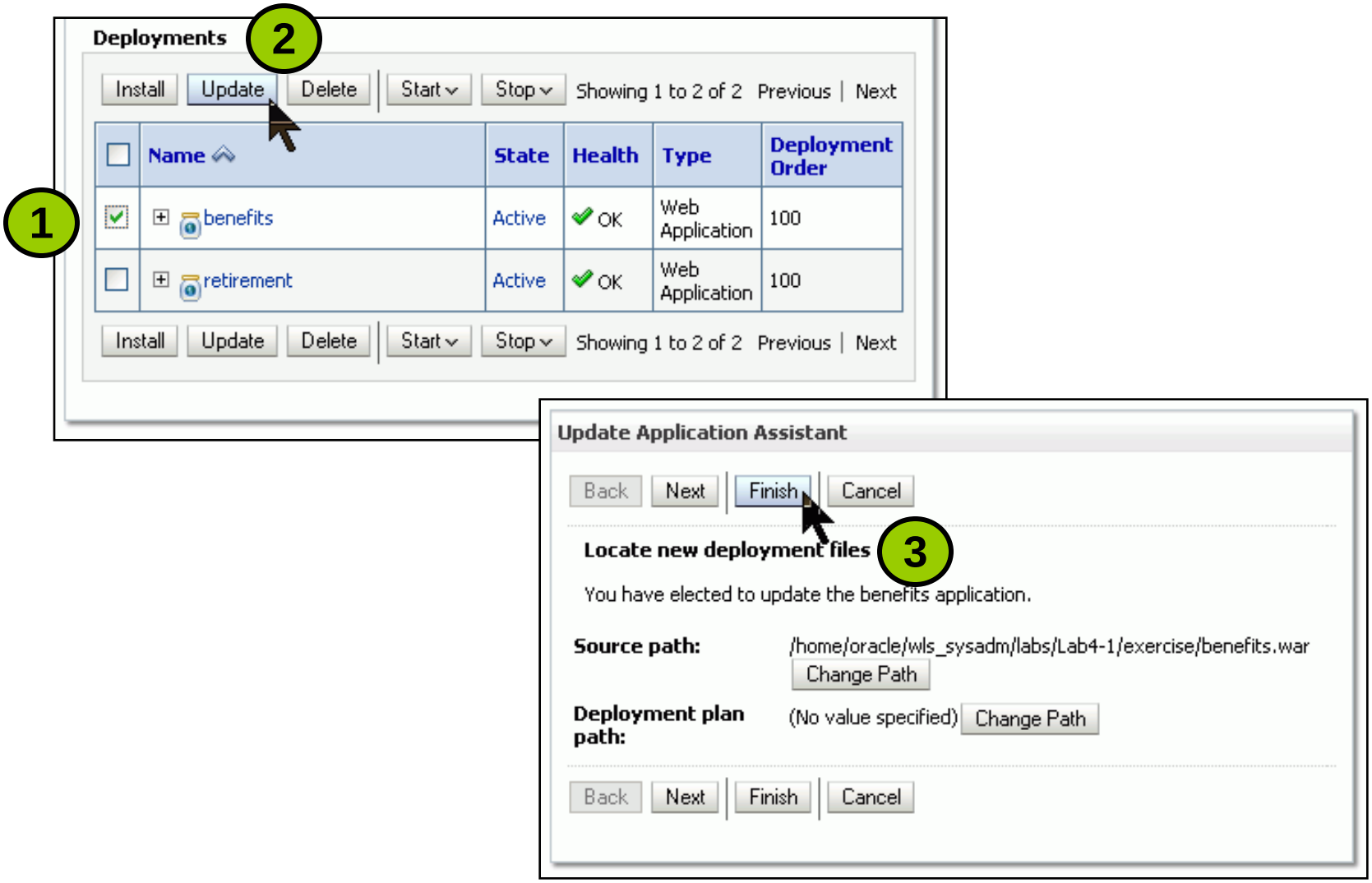
Preparing a New Application: Settings



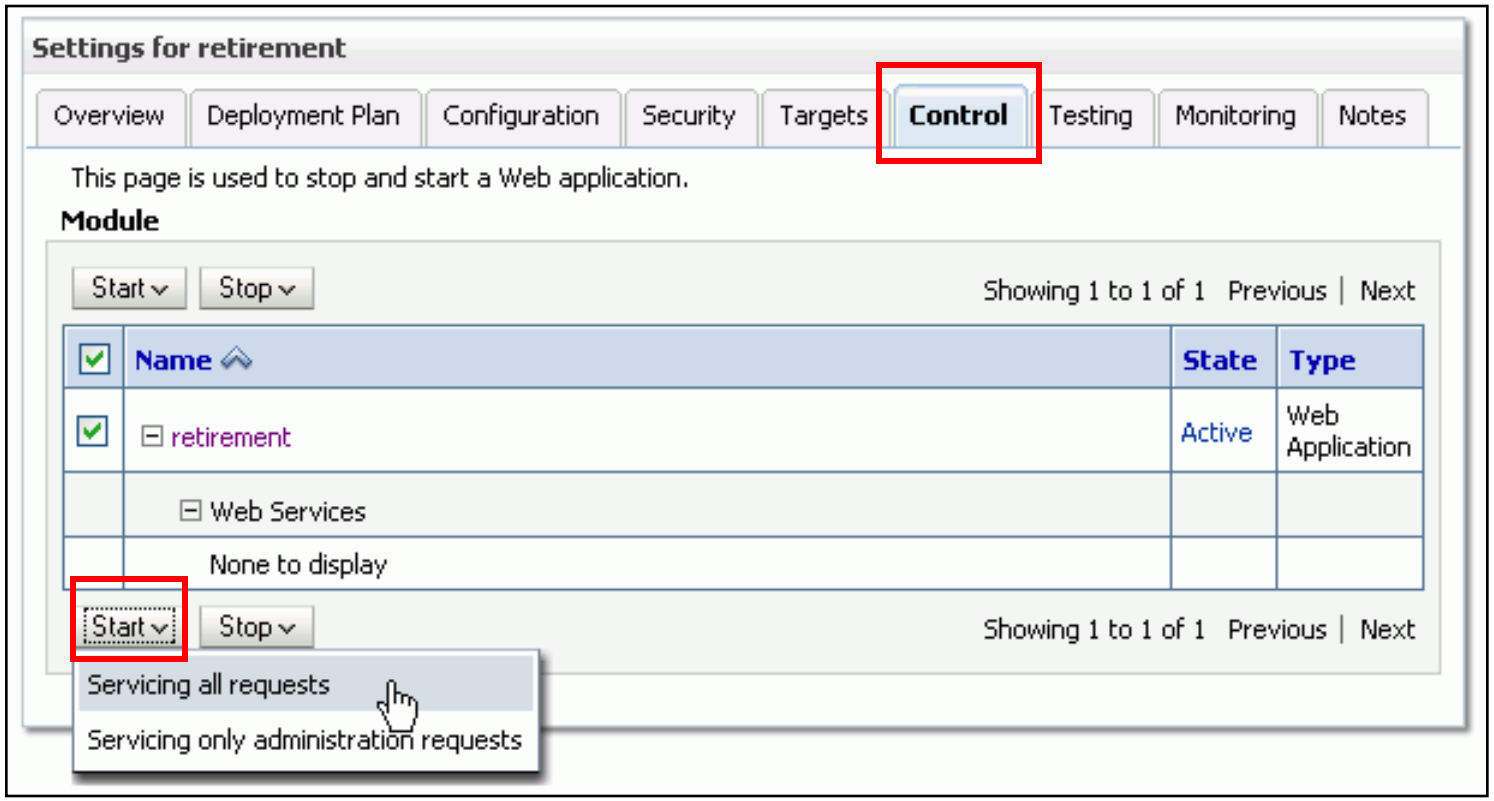
Deploying or Undeploying Applications



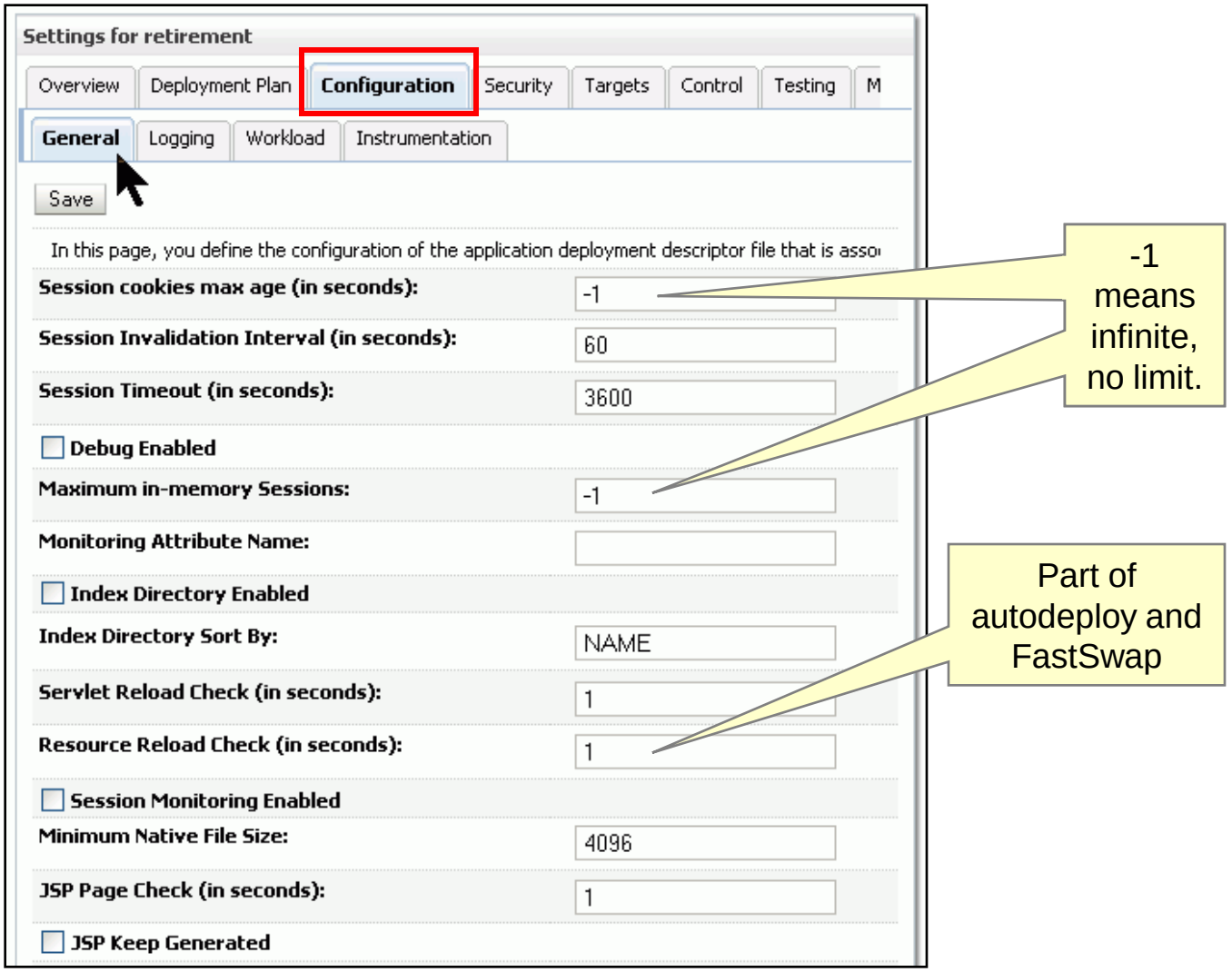
Redeploying an Application



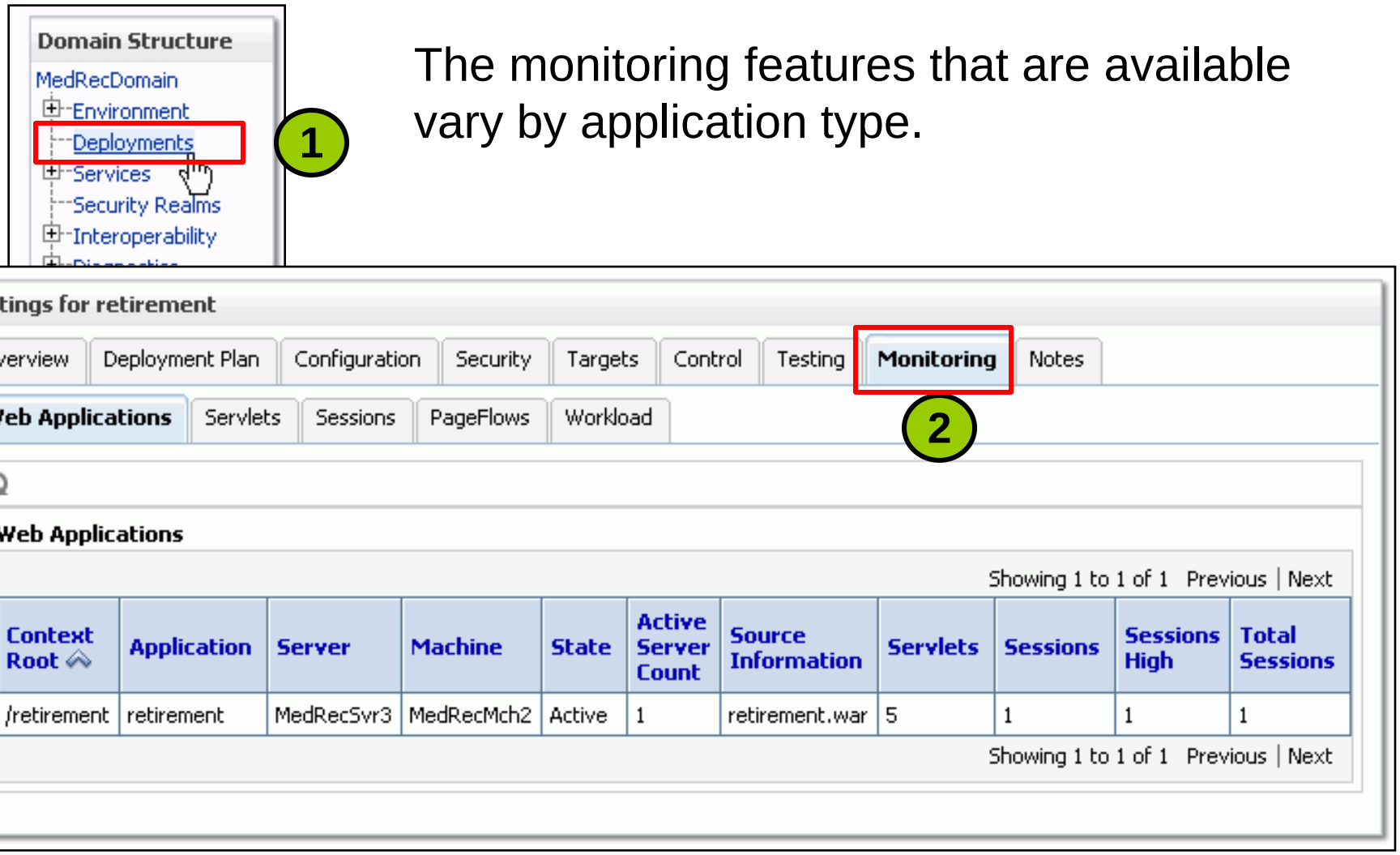
Starting and Stopping an Application



Editing Deployment Descriptors



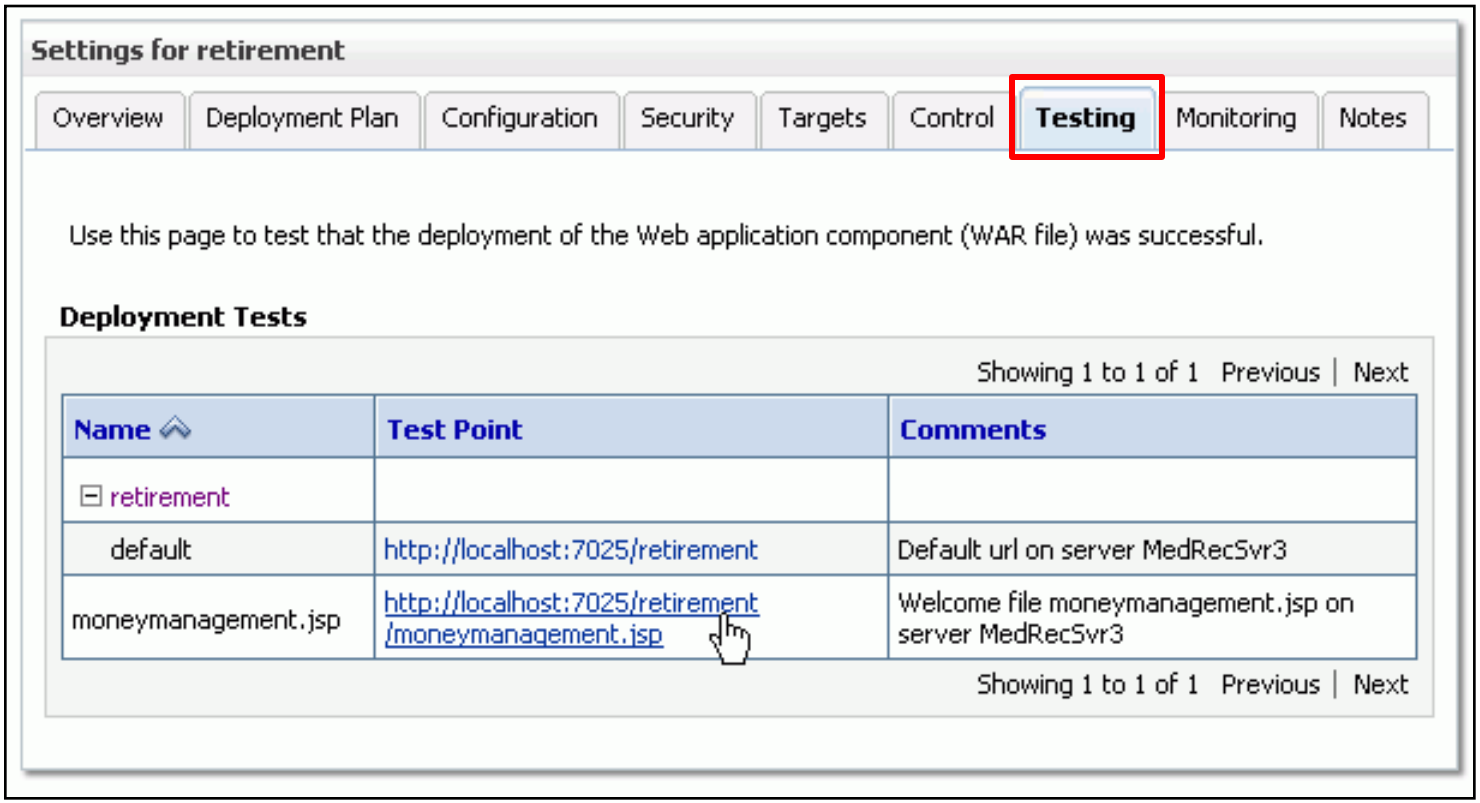
Monitoring an Application



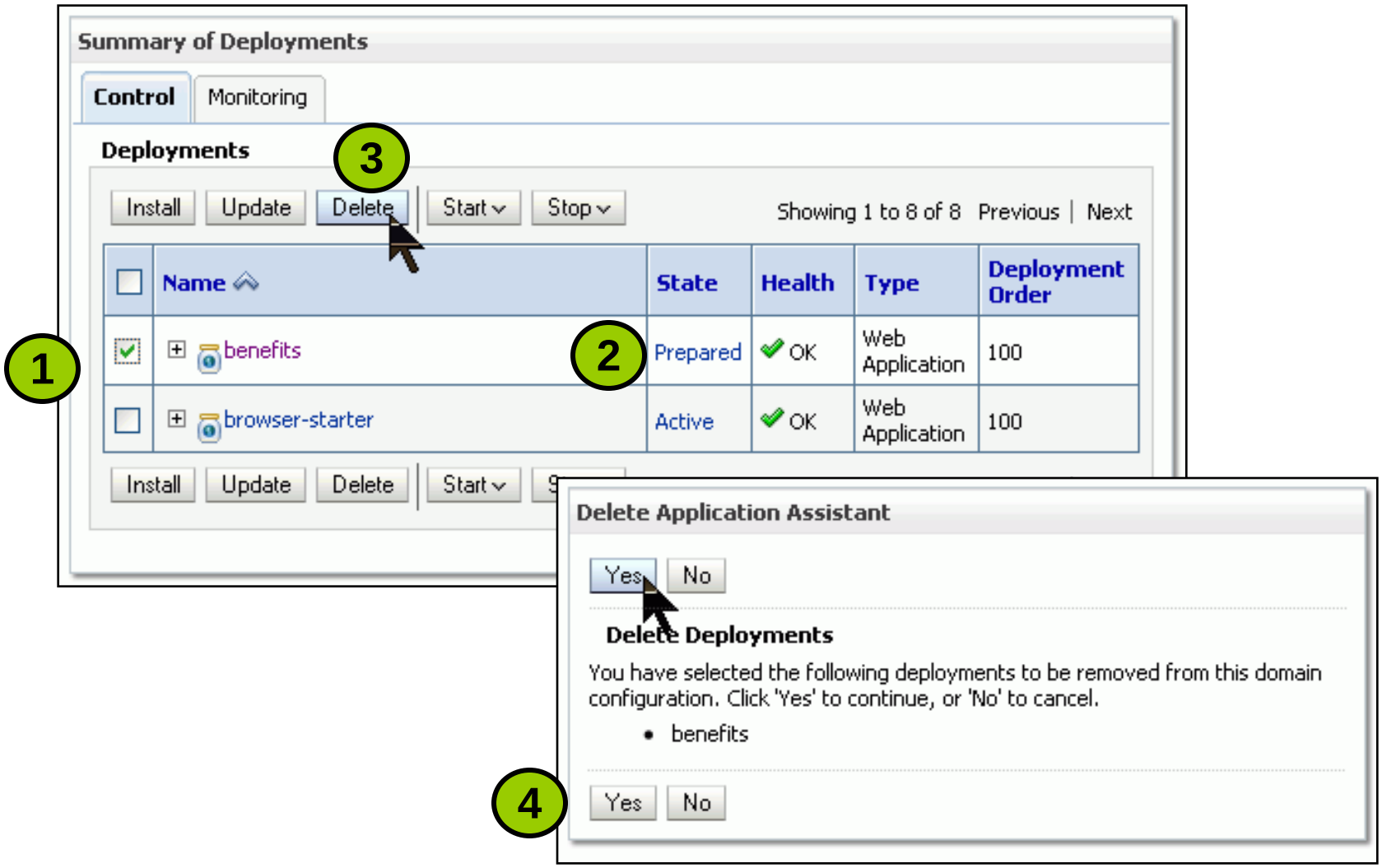
The monitoring features that are available vary by application type.

Application Testing

You can test a deployed application using the Administration Console.



Deleting Applications



Command-Line Deployment

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

`weblogic.Deployer` Syntax:

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

Deployment with `weblogic.Deployer`

- Prepare and deploy a new application:

```
java weblogic.Deployer -adminurl t3://adminserver:7001
   -username myuser -password welcome1 -name HRServices
   -source /usr/HRServices.ear -targets serverA -deploy
```

- Redeploy an application:

```
java weblogic.Deployer -adminurl t3://adminserver:7001
   -username myuser -password welcome1 -name HRServices
   -redeploy
```

- Undeploy an application:

```
java weblogic.Deployer -adminurl t3://adminserver:7001
   -username myuser -password welcome1 -name HRServices
   -undeploy
```

- To list all deployed applications:

```
java weblogic.Deployer -adminurl t3://localhost:7001
   -username myuser -password welcome1 -listapps
```

More `weblogic.Deployer` Examples

- To list all deployment tasks:

```
java weblogic.Deployer -adminurl t3://localhost:7001
   -username system –password welcome1 -listtask
```

- To cancel a deployment task:

```
java weblogic.Deployer -adminurl t3://localhost:7001
   -username system –password welcome1 -cancel -id tag
```

```
[oracle@wls-sysadm]$ java weblogic.Deployer
    -adminurl t3://localhost:7020
    -username weblogic -password Welcome1 –listapps
weblogic.Developer invoked with options:
-adminurl t3://localhost:7020 –username weblogic -listapps
 jsf [LibSpecVersion=1.2,LibImplVersion=1.2.9.0] <ACTIVE VERSION>
 jstl [LibSpecVersion=1.2,LibImplVersion=1.2.0.1] <ACTIVE VERSION>
 medrec
Number of Applications Found : 3
[oracle@wls-sysadm]$
```

Deploying Applications with WLST

WLST provides a number of deployment commands. You can use these commands to:

- Perform life-cycle operations on applications and stand-alone modules in an Oracle WebLogic Server instance
  - Deploy
  - Undeploy
  - Redeploy
- Update an existing deployment plan
- Start and stop a deployed application

Deploying an Application with WLST

Deploy an application (`deployapp.py`):

```
##
# WLST script for Deploying Java EE Application #
##

# Connect to the server
print 'Connecting to server  .... '
connect('weblogic','welcome1','t3://localhost:7001')

appname = "mbeanlister"
applocation = "c:/domains/MedRecDomain/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()
```

Deployment with WLST

Prepare and deploy a new application, or redeploy an existing one:

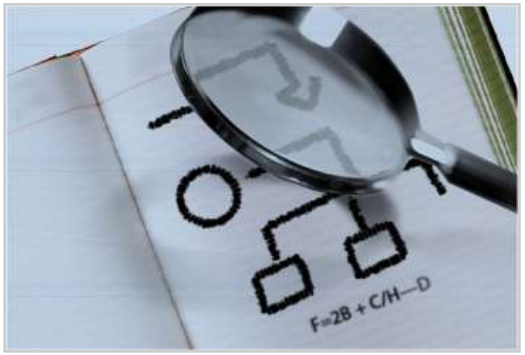
```
connect('myuser','mypass1','t3://adminserver:7001')
name = "HRServices"
location = "/usr/myapplications/HRServices.ear"

deploy(name, location, targets='serverA')
```

Other WLST deployment commands:

```
distributeApplication(location, targets='serverA')
startApplication(name)
redeploy(name)
stopApplication(name)
listApplications()
```

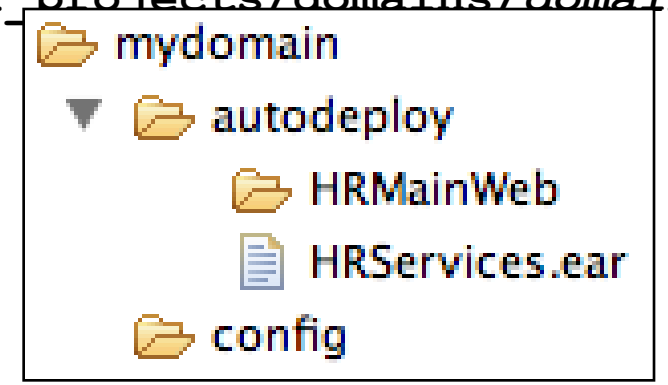
Road Map

- Deployment concepts
- Development deployment
  - Autodeployment
  - FastSwap
- Front-end with a Web server



Autodeployment

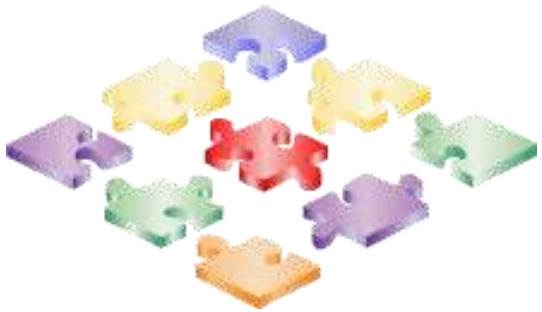
- By default, the autodeployment feature is enabled only if the domain is *not* running in production mode.
- When enabled:
  - The administration server monitors its “autodeploy” folder for new, updated, or removed applications
  - Applications are targeted only to the administration server
  - Developers can quickly test or experiment with an application
- **<WL_HOME>/user_projects/domains/*domain*/autodeploy**



Autodeploying Using an Expanded Directory

If the following conditions are true, you are a candidate for autodeploy. Consider autodeploy if the application is:

- In the development phase
- Being updated frequently
- Deploying to a single machine (for example, only the administration server)



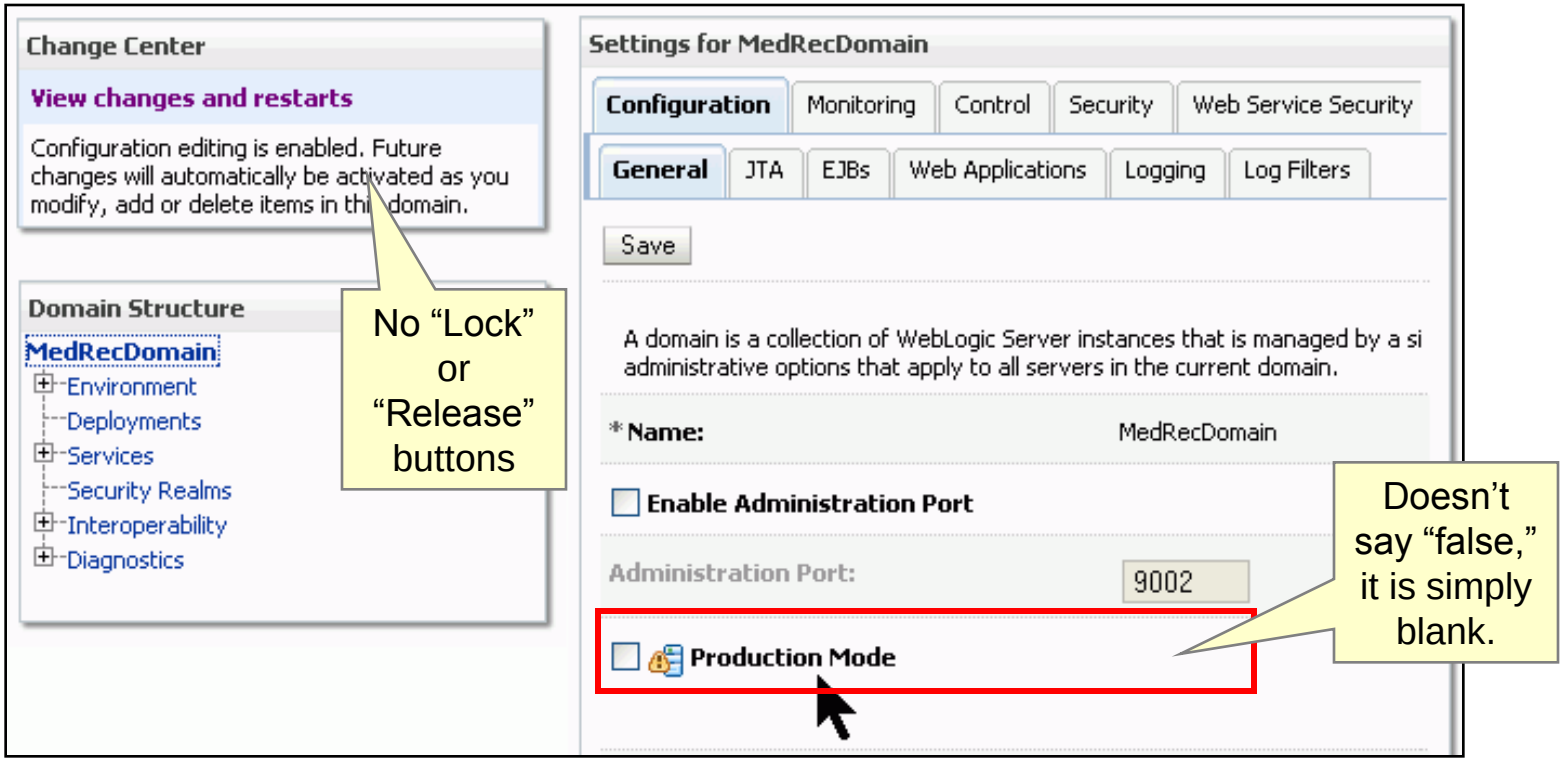
FastSwap and On-Demand Deployment

- WebLogic's FastSwap feature is:
  - Enabled using the WebLogic deployment descriptors
  - Available only if the domain is *not* running in production mode
  - Applicable only to Web applications that are *not* archived
- When enabled:
  - WebLogic automatically reloads the modified Java class files within applications
  - Developers can perform iterative development without an explicit redeployment
- On-demand deployment:
  - **`weblogic.xml:`**
    `<fast-swap>true</fast-swap>`

Production Mode Flag

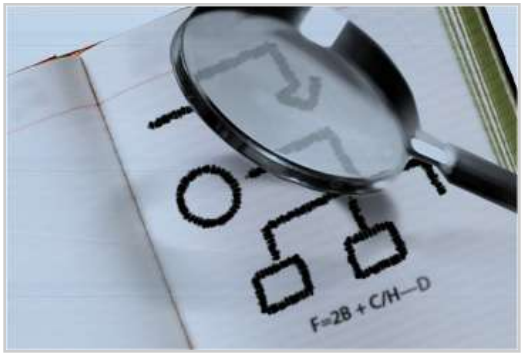
When Production Mode is disabled, applications can be dynamically deployed.

- An application poller is enabled in development mode.



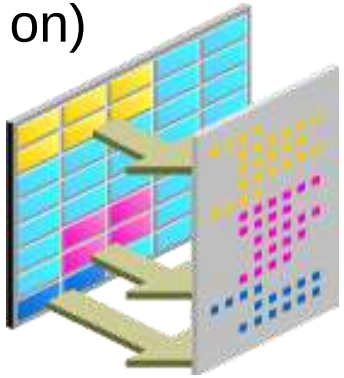
Road Map

- Deployment concepts
- Development deployment
- Front-end with a Web server
  - Web servers defined
  - HTTP
  - Static and dynamic content
  - Redirection

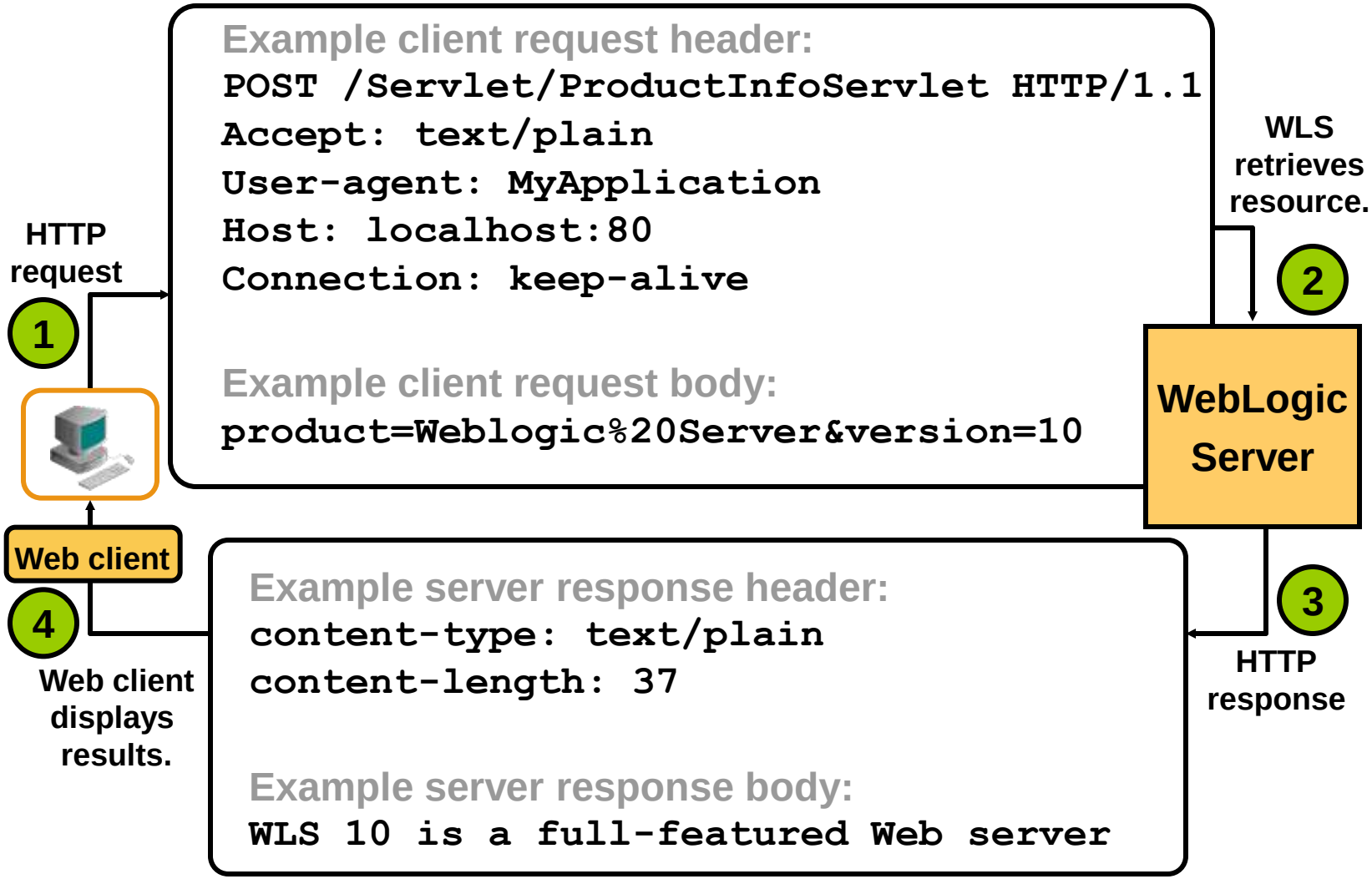


Role of Web Servers

- Web servers are responsible for handling HTTP requests from clients.
- Web servers typically return:
  - Static content (HTML pages, graphics, media, and so on)
  - Dynamic content (generated by servlets, JSPs, CGIs, JSF, Struts, and so on)



A Typical Web Interaction



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 on 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

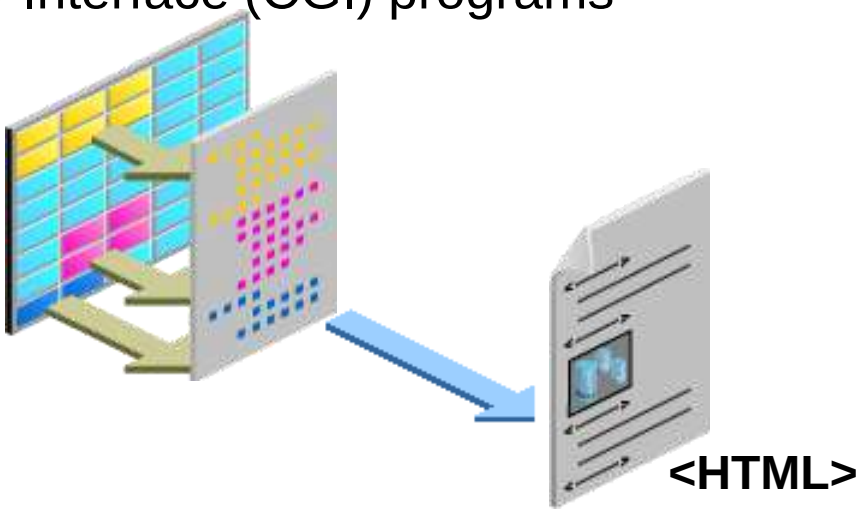
- HTTP status codes indicate to the client whether or not the request was successful, and if not, provide the client a reason for a failed request.
- They are used by clients to provide alternate behavior.
- Some representative codes are:
  - Indicating success:
    - The default status code is 200, 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 prompts the user for an ID and password to log in.

Static Content

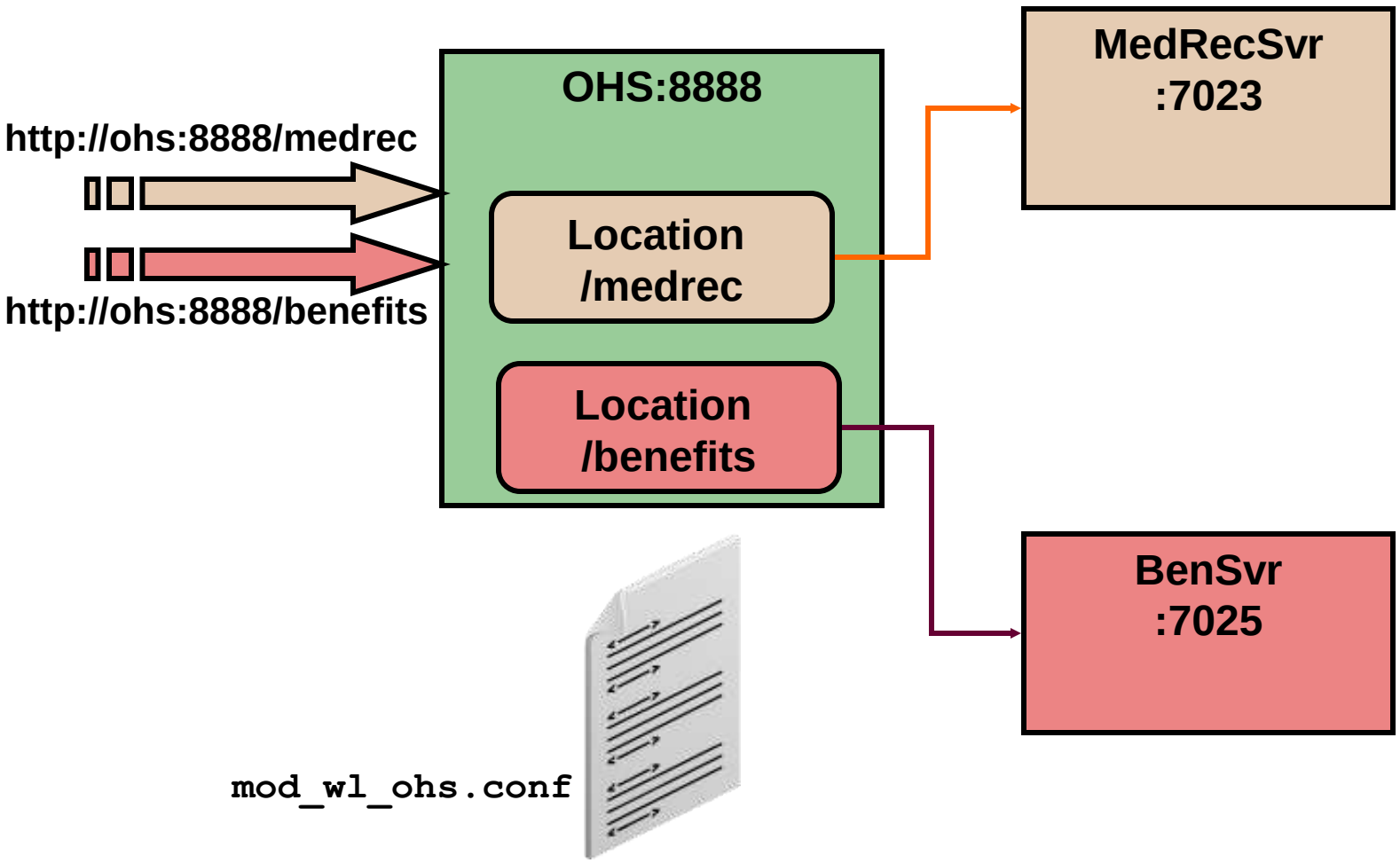
- Static content documents are predefined on the server and do not change.
- Oracle WebLogic Server can be used to serve static content such as:
  - HTML documents
  - Images
  - Media
  - PDF documents
- Oracle WebLogic Server can serve static documents:
  - Over standard HTTP
  - Through SSL using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

Dynamic Content

- Dynamic content documents may change based on the client's request.
- Dynamic content often involves a database query.
- HTML documents can be created using various means including:
  - Servlets
  - Common Gateway Interface (CGI) programs
  - JSPs
  - JSF
  - Struts



Configuring Oracle HTTP Server to Serve Multiple WebLogic Servers



mod_wl_ohs.conf

The main sections of `mod_wl_ohs.conf` are:

- `LoadModule`: Is enabled by default to load the weblogic_module when OHS starts
- `IfModule`: Specifies the host and port details of the WLS server or cluster. For example:

```
<IfModule mod_weblogic.c>
  WebLogicCluster wls-sysadm:7023,wls-
  sysadm:7025
</IfModule>
```

- `Location`: Specifies the root context of the application and advises OHS that WLS will handle requests for that application. For example:

```
<Location /medrec>
  SetHandler weblogic-handler
</Location>
```

Verifying Ports Used by OHS

If OHS is running, you can verify ports using the `opmnctl status` command with the `-l` option:

```
/u01/app/oracle/product/fmw/11.1.0/webtier/instances/instance2/bin
[oracle@edvmr1p0 bin]$ ./opmnctl status -l

Processes in Instance: instance2
-------------+--------------+------+--------+-----------+---------+----------+----
ias-component | process-type |  pid | status |       uid | memused |   uptime |
     ports
-------------+--------------+------+--------+-----------+---------+----------+----
ohs2          | OHS          | 4253 | Alive  | 559158019 |  358996 | 18:11:13 |
     https:8889,https:4443,http:8888

[oracle@edvmr1p0 bin]$
```

Quiz

Which environment supports FastSwap?

1. Production mode, archived files
2. Production mode, expanded files
3. Development mode, archived files
4. Development mode, expanded files
5. All of the above

Quiz

What is the `web.xml` tag indicating the test point for testing applications?

1. `<welcome-file>`
2. `<test-point>`
3. `<deploy>`
4. `<monitor>`
5. `<debug>`

Quiz

It is possible to deploy an exploded directory that contains JAR files.

1. True
2. False

Summary

In this lesson, you should have learned how to:

- Enable autodeploy with manual deployment
- Configure and deploy Web applications via the Administration Console, command line, and WLST
- Redeploy and remove applications
- Configure deployment descriptors
- Test deployed applications
- Front-end deployed applications with a Web server