Configuring a WebLogic Server Environment

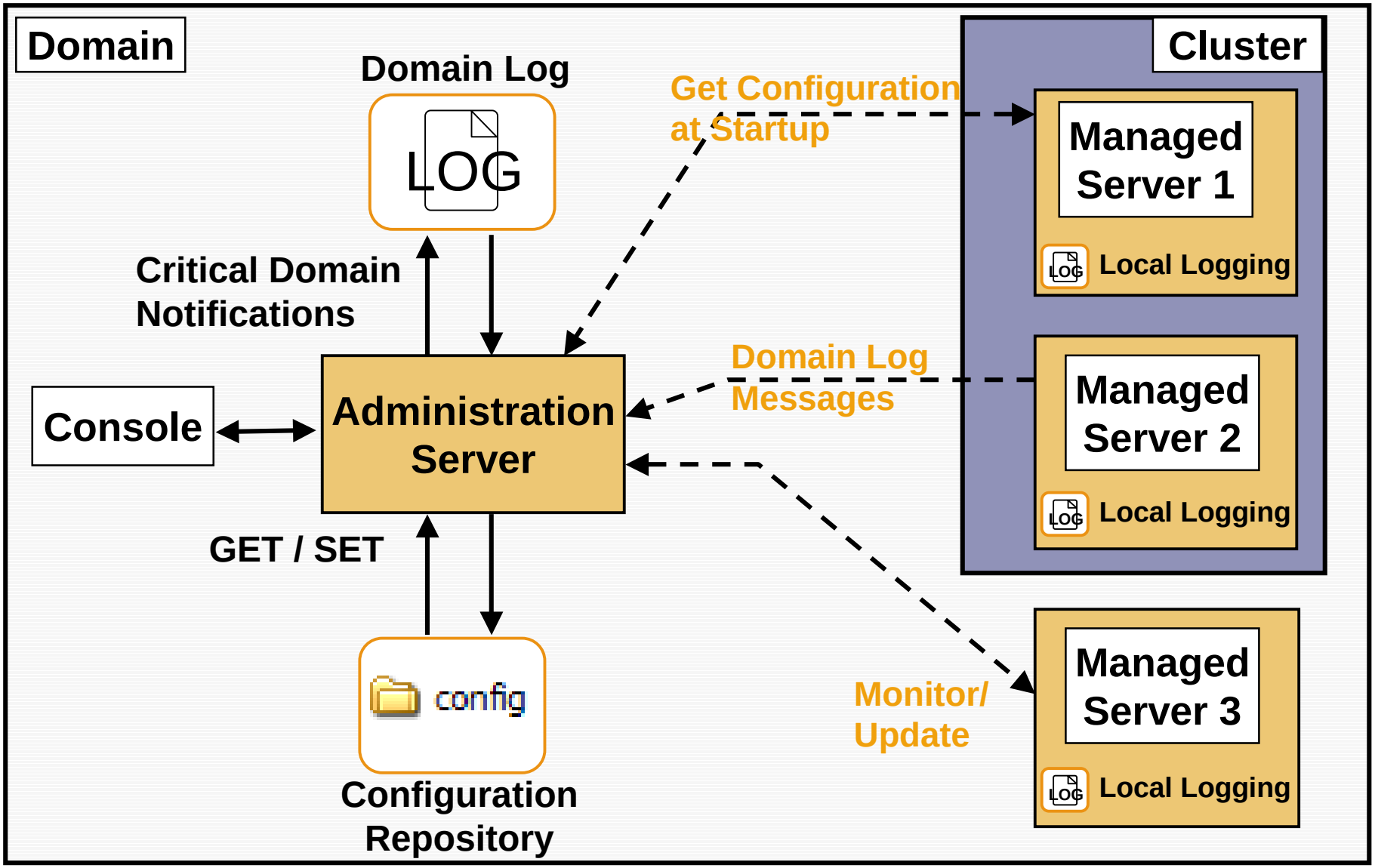
At the end of this module you will be able to:

- ✓ Configure domains, machines, and managed servers
- ✓ Start the WebLogic Server Administration Console
- ✓ Start managed servers at boot time
- ✓ Set Basic properties using the Administration Console
- ✓ Perform basic Administration from the Command Line
- ✓ Administer servers and managed servers

Road Map

1. **Configuring Domains**
   - How WebLogic Server Domain works
   - Domain Directory Structure and Files
   - Creating a Domain
2. Configuring Servers
3. Domain Templates
4. Console Administration
5. Command Line Administration

Domain Overview



Configuring a Domain

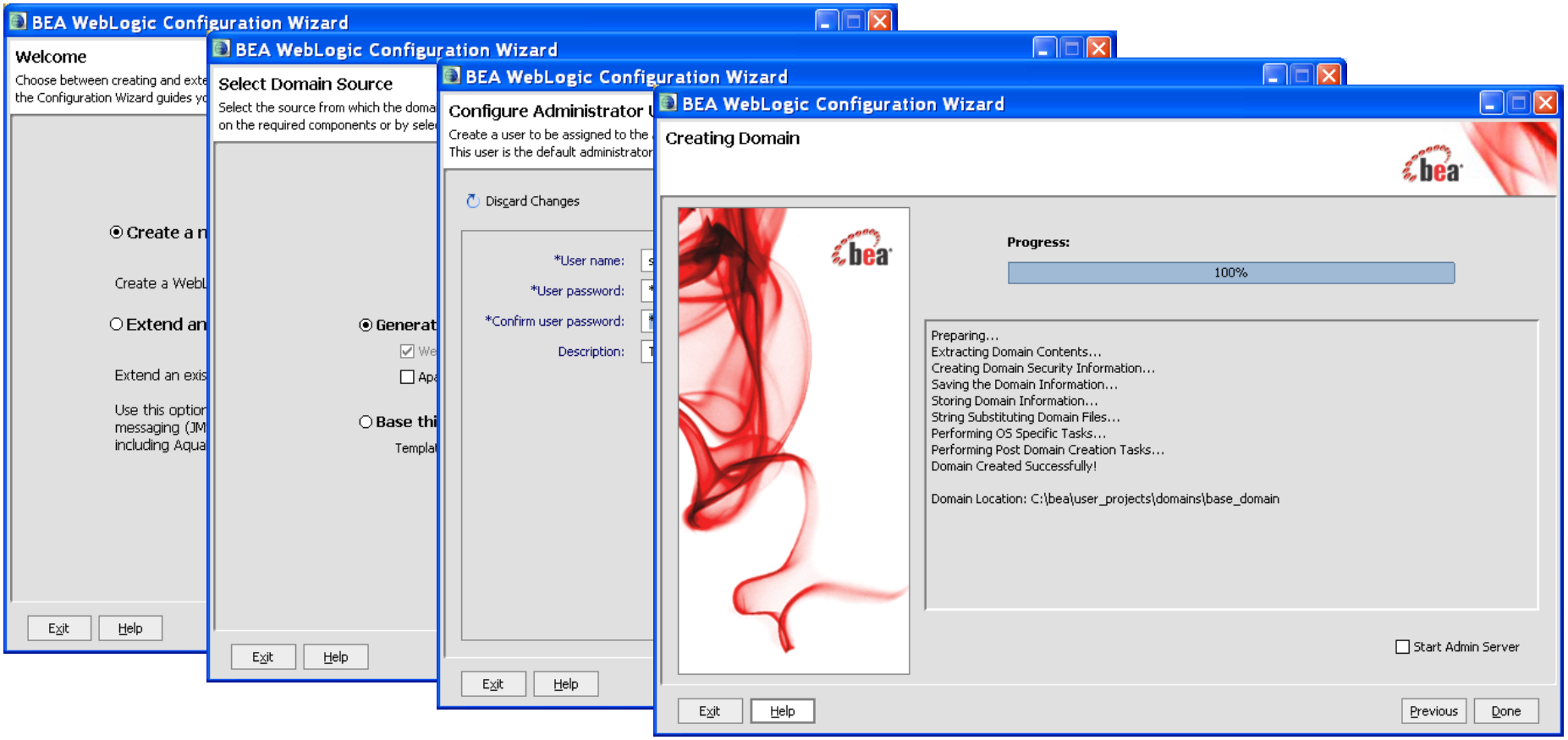
- After installing, configure a WLS domain on which to develop and deploy applications.
- When you create a domain, you define a collection of *resources*, such as:
  - Managed servers
  - Clusters
  - Database connections
  - Security services
  - J2EE applications
- Use the Configuration Wizard to create and configure WLS domains.

Starting Configuration Wizard

- Scripts in `<WEBLOGIC_HOME>/common/bin` directory
- Graphical mode
  - Windows Start menu
  - [Windows] `config.cmd`
  - [Unix/Linux] `config.sh`
- Console mode
  - [Windows] `config.cmd –mode=console`
  - [Unix/Linux] `config.sh –mode=console`
- Note: silent mode is deprecated in WebLogic Server 9.X

Configuration Wizard – Graphical Mode

- The graphical version of the domain configuration wizard walks the user through each step.



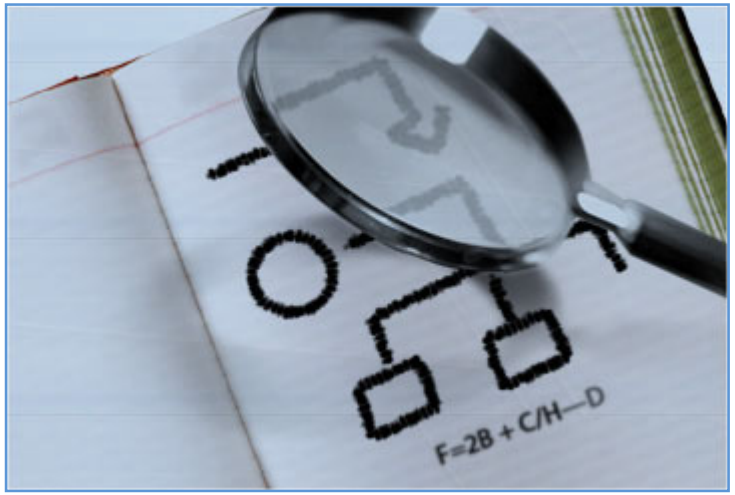
Domain Directory Structure

Directory	Column Head
🗁 *domain-name*	The name of this directory is the name of the domain.
🗀 autodeploy	In development mode, WLS automatically deploys any applications or modules that you place in this directory.
🗀 bin	The scripts for starting and stopping the Administration Server and the Managed Servers in the domain.
🗀 config	The current configuration and deployment state of the domain. **`config.xml.`**
🗀 console-ext	Console extensions.
🗀 init-info	Server initialization information.
🗀 lib	JAR files added to the classpath of each server instance.
🗀 pending	Domain configuration changes that have been requested, but not yet activated.
🗀 security	Domain-wide security-related files.
🗁 server	One subdirectory for each server in the domain.
🗀 *server-name*	The server directory for the WLS instance with the same name.

Section Review

In this section we discussed:

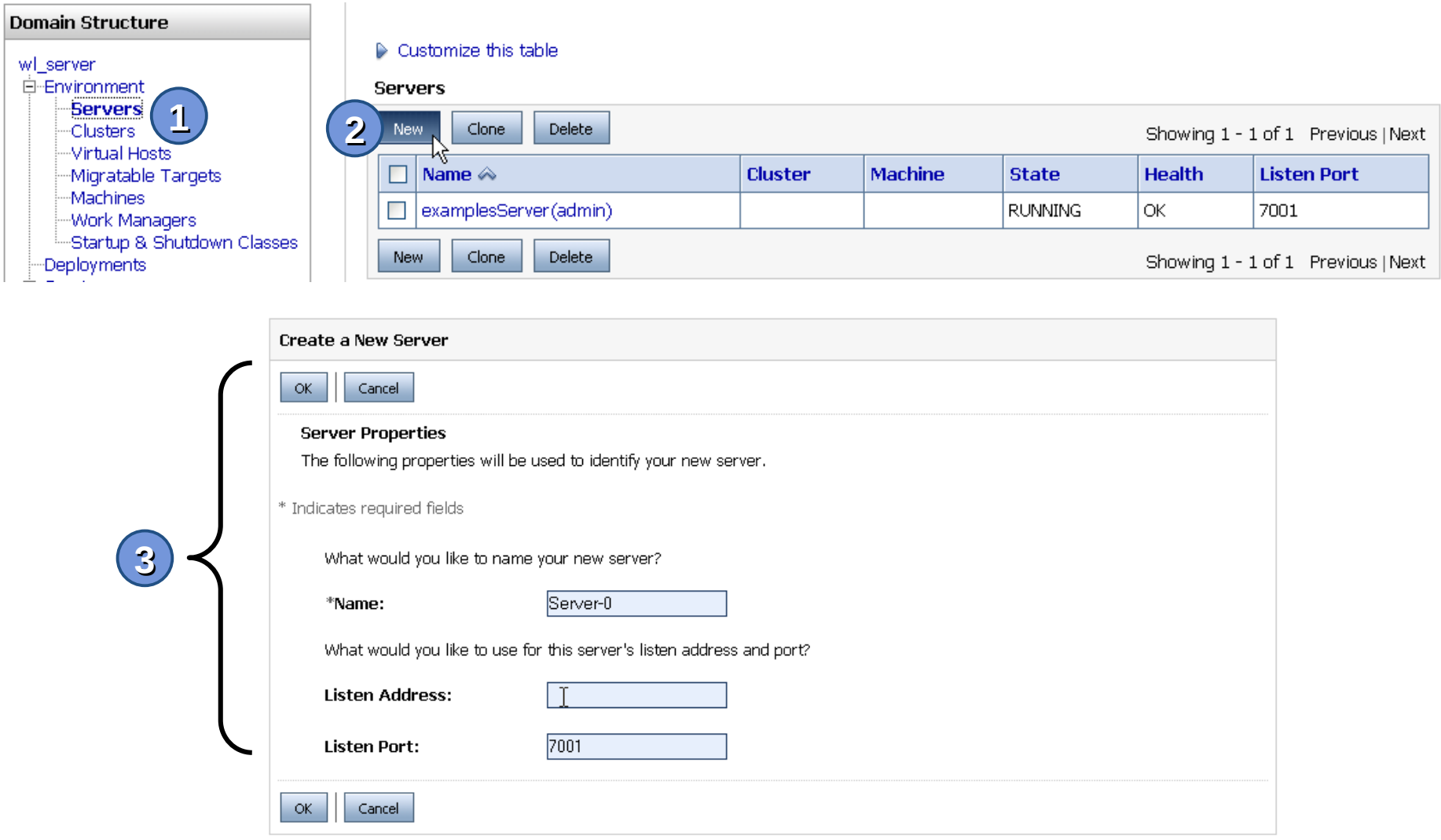
- ✓ How a WebLogic Server domain works
- ✓ The domain directory structure
- ✓ Domain files
- ✓ How to create a domain



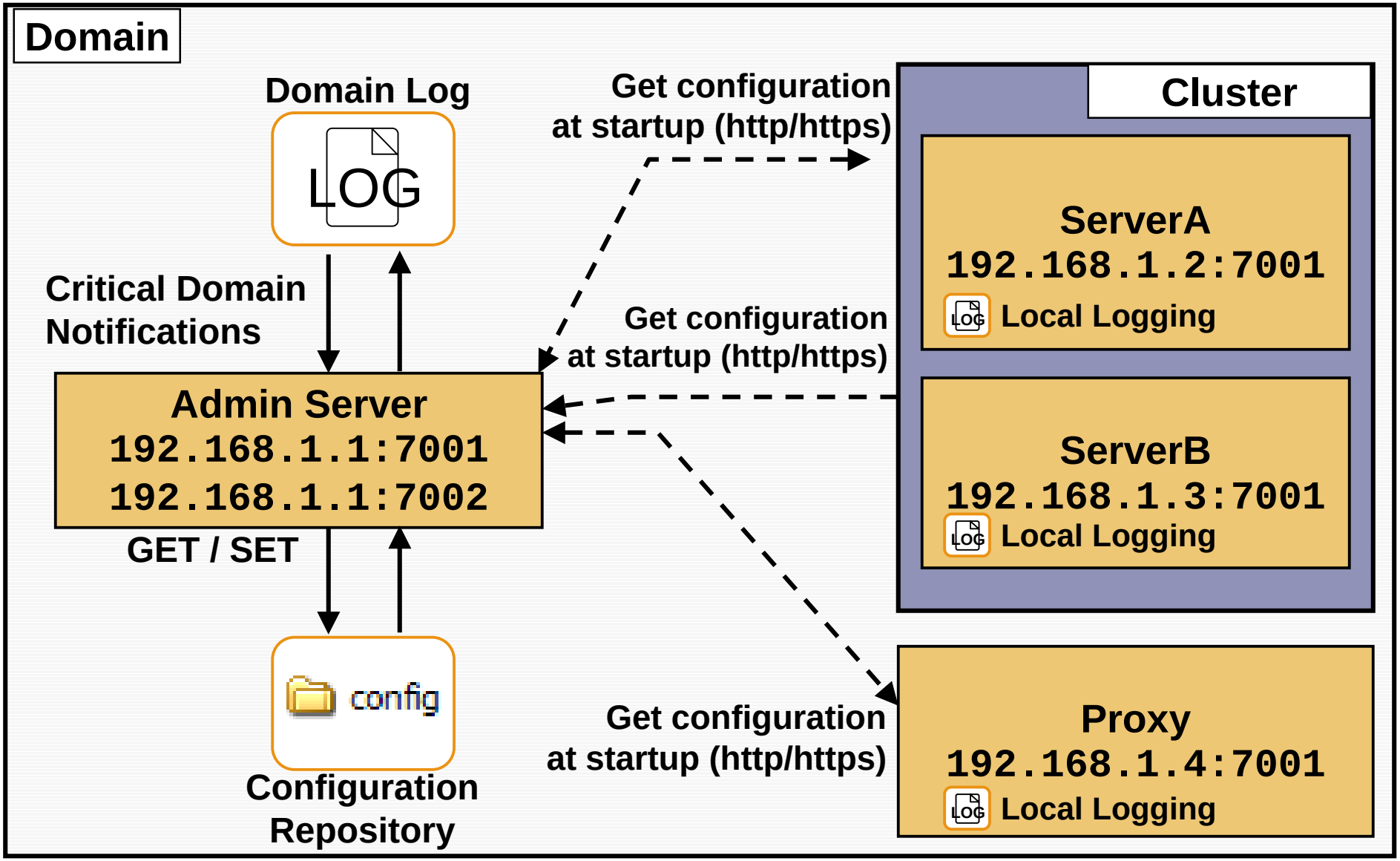
Road Map

1. Configuring Domains
2. **Configuring Servers**
   - Configuring Managed Servers
   - Starting Managed Servers
   - Running Multiple WLS Instances
3. Domain Templates
4. Console Administration
5. Command Line Administration

Configuring Managed Servers



Starting Managed Servers



Starting Managed Servers

- To start a managed server you must:
  - Specify a server name
  - Specify an administration server URL from which to load configuration information

To start a managed server:

```
java -server –Xms256m –Xmx512m
1 -Dweblogic.Name=%SERVER_NAME% -Dplatform.home=C:\bea\weblogic91
  -Dweblogic.management.username=%WLS_USER%
  -Dweblogic.management.password=%WLS_PW%
2 -Dweblogic.management.server=%ADMIN_URL%
  -Dweblogic.ProductionModeEnabled=%STARTMODE%
  -Djava.security.policy=%WL_HOME%\server\lib\weblogic.policy
   weblogic.Server
```

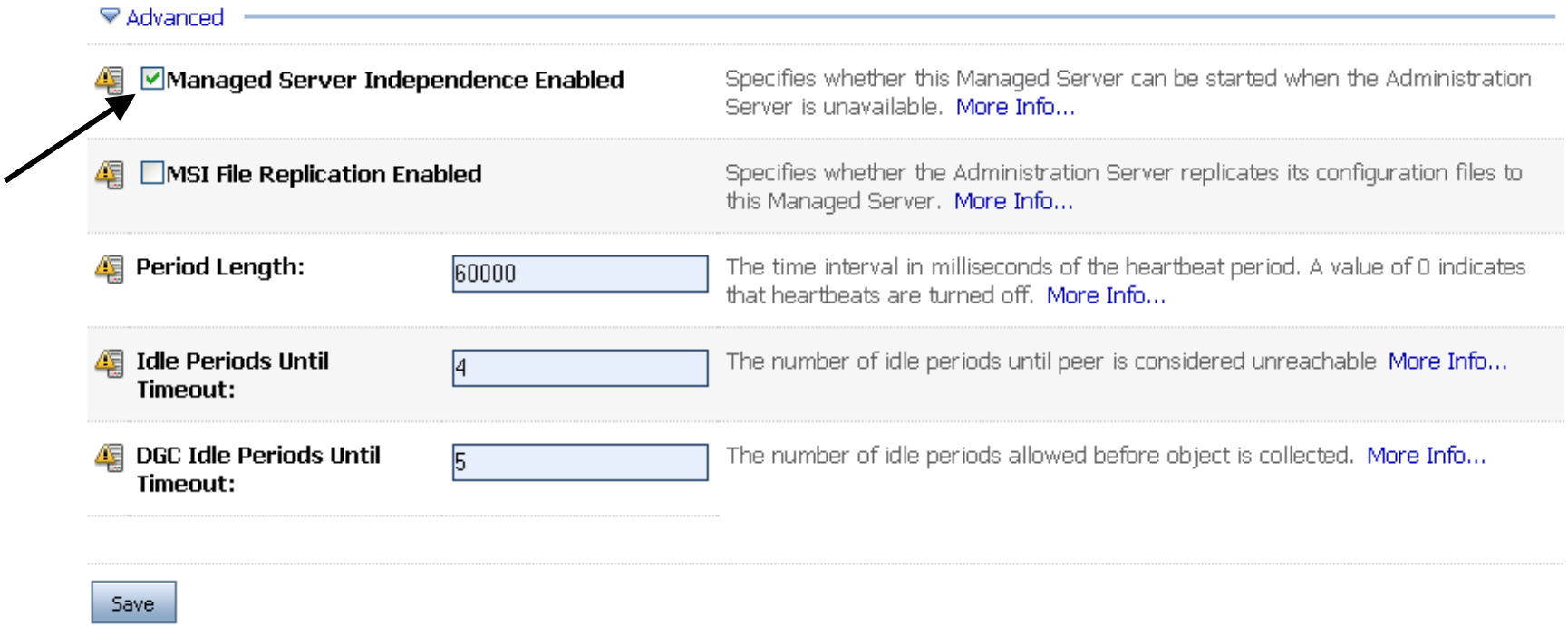
<startManagedWeblogic.cmd>

Creating a Boot Identity File

- Create a file called `boot.properties` located in the domain's root directory containing 2 lines:
  - `username=username`
  - `password=password`
- First time you start the server, the server reads the Boot Identity file and overwrites it with an encrypted version of the username and password.
- Thereafter, the server will remember your identity for the subsequent startup cycles.

Managed Server Independence…

- By default, managed servers can function independently of the administration server.
- Configure Managed Server Independence Mode from the Console:



…Managed Server Independence

- If the administration server is unavailable at boot time, managed servers search for:
  - `config.xml`
  - `SerializedSystemIni.dat`
  - `boot.properties(optional)`
- Each managed server looks in its local config directory for `config.xml` - a replica of the domain's `config.xml`

What If an Admin Server Is Down?

- The administration server:
  - Can go down without affecting the operation of managed servers
  - Can be restarted while managed servers are still running
- When an administration server goes down:
  - Domain log entries are lost while it is down
  - Managed servers can start in independent mode
  - The administration console and management tools are unavailable

Administration Server Backup

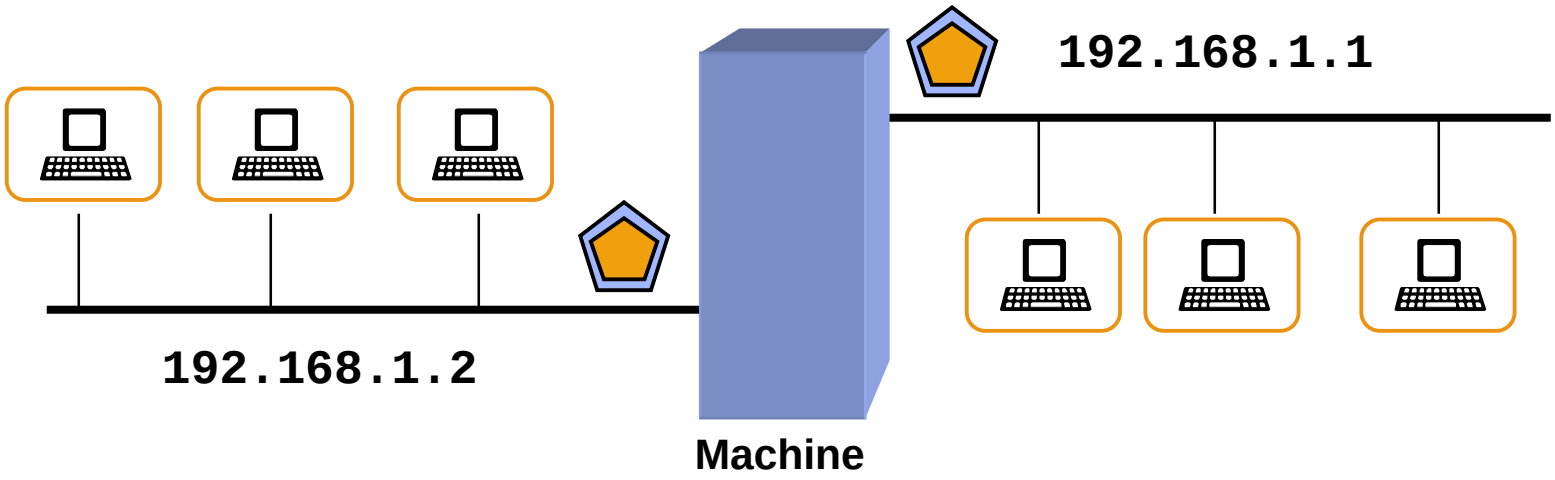
- WLS allows the creation of a backup of the server as follows:
  - Install (if necessary) WLS on backup machine
  - Copy application files to backup machine
  - Copy configuration files to backup machine
  - Restart Administration server on new machine
- The new Administration server will contact managed servers and inform them that it is running on a new IP address.

Running Multiple WLS Instances

- You can run multiple instances of WLS using different configurations on the same physical machine at the same time by:
  - Assigning multiple IP address to a machine (multihoming) and defining each server to use a unique IP address
- or by:
  - Specifying the same IP address but using different listen ports

Multihoming

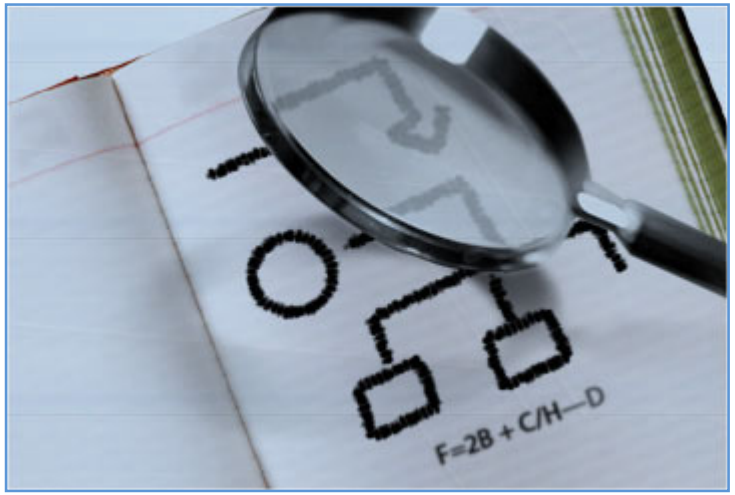
- A multihomed machine:
  - Is a machine with multiple IP addresses
  - Can run a different WLS instance bound to each IP address
  - Can be used to configure a cluster on a single machine



Section Review

In this section we discussed:

- ✓ How to define and start an administration server
- ✓ How to create and start a managed server
- ✓ Managed server independence
- ✓ Administration server backup



Road Map

1. Configuring Domains
2. Configuring Servers
3. **Domain Templates**
   - Creating customized domain templates
4. Console Administration
5. Command Line Administration

Custom Domain Templates

- A domain template defines the full set of resources within a domain.
- Although BEA provides templates for creating any platform domain, you may wish to create your own or customize an existing template.
- The Domain Template Builder lets you define templates:
  - Define a domain and replicate it across multiple projects
  - Distribute a domain packed with an application that has been developed to run in it

Creating a Domain Template

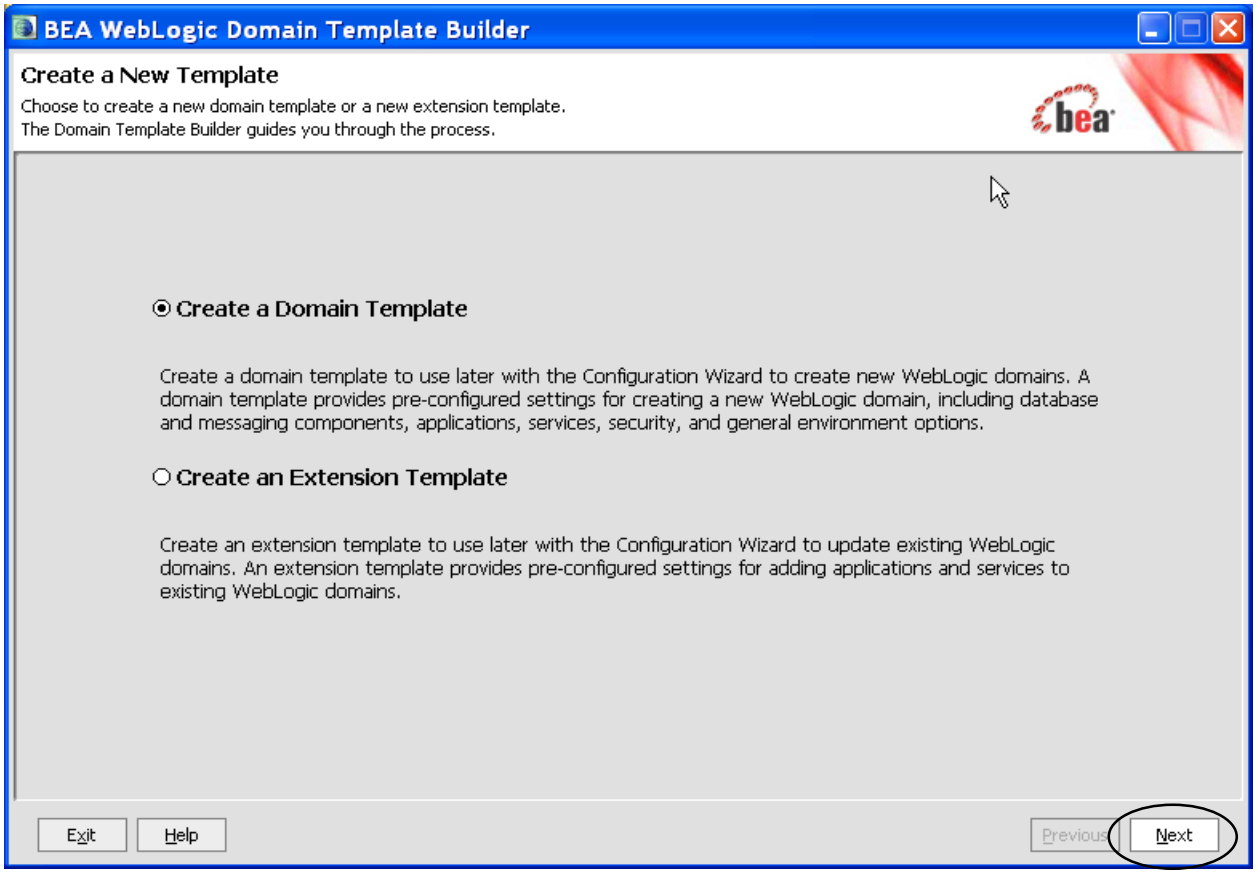
To create a domain template:

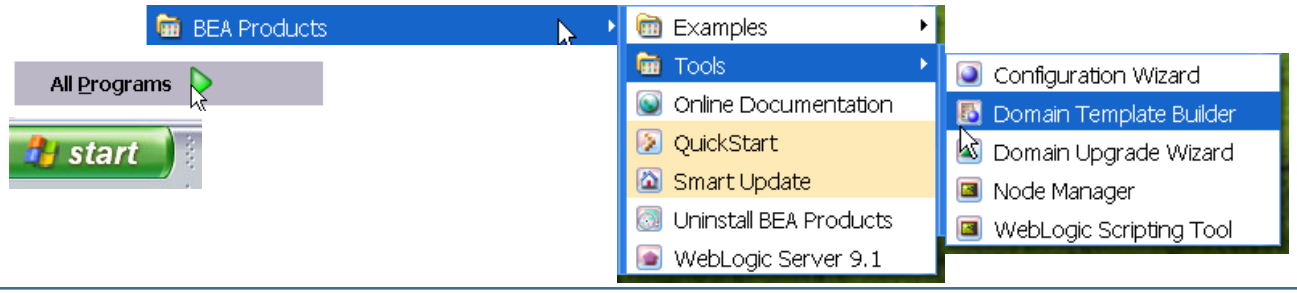
1. Create a new template using Domain Template Builder
2. Select Configuration Template Source
3. Describe the template.
4. Add files to the Template.
5. Add SQL Scripts to the Template
6. Configure the Administration Server, Username and Password.
7. Specify Start Menu entries.
8. Review Domain Template
9. Create Template

Starting Domain Template Builder

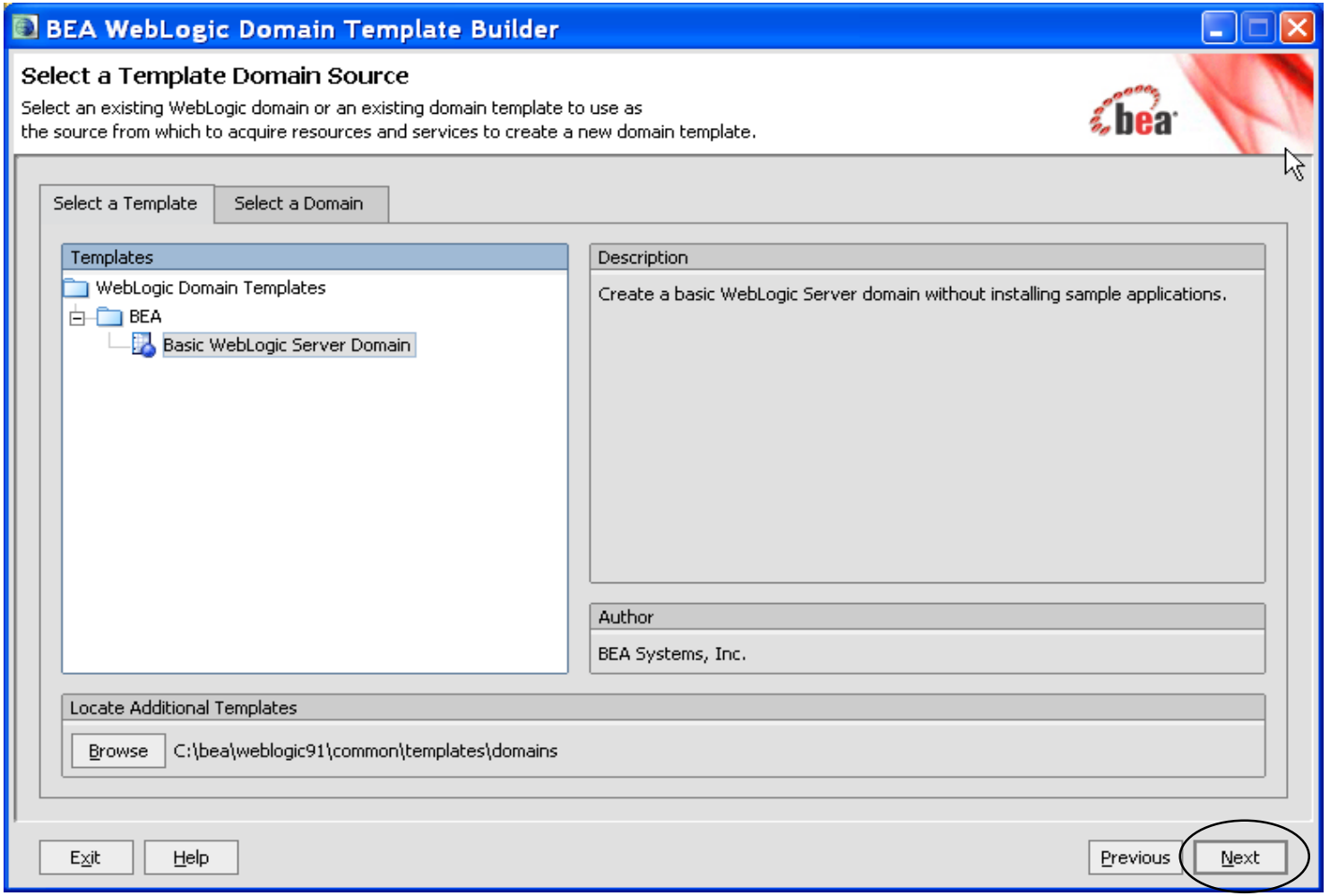
- Using the GUI mode in Windows environment or
- Using script `config_builder.cmd` or `sh config_builder.sh` under `\common\bin directory`

Create a New Template

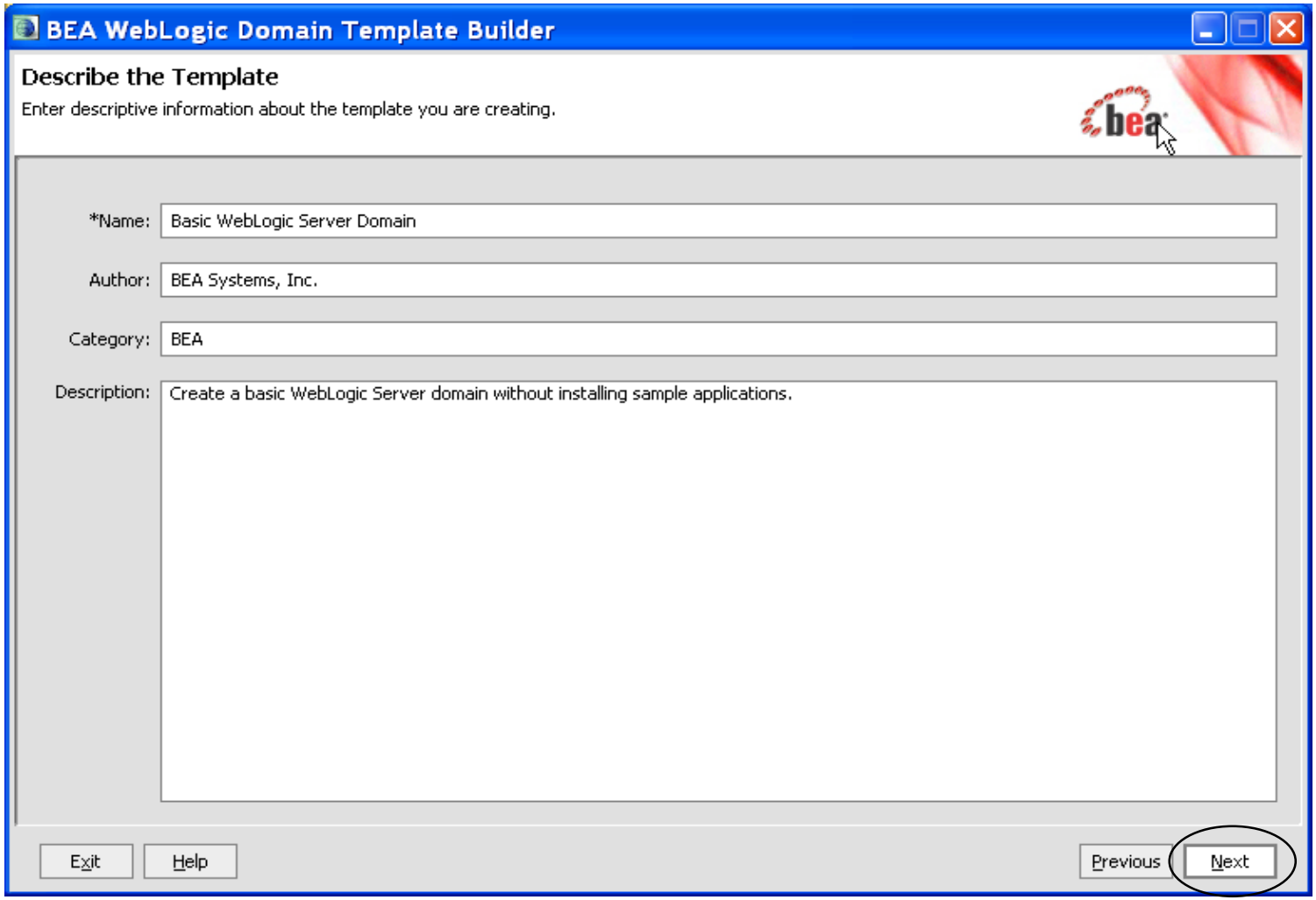




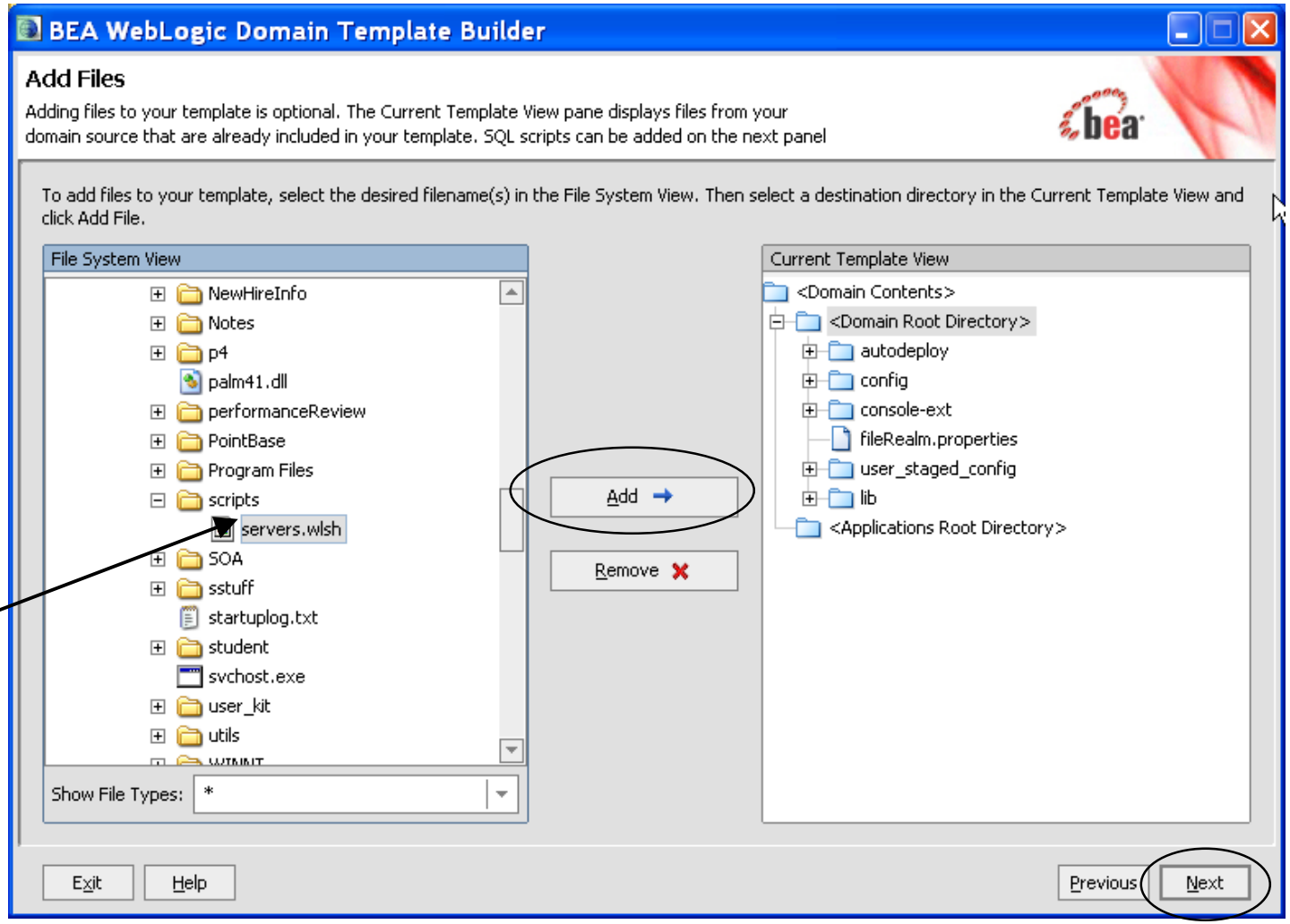
Select Configuration Template Source



Describe the Template

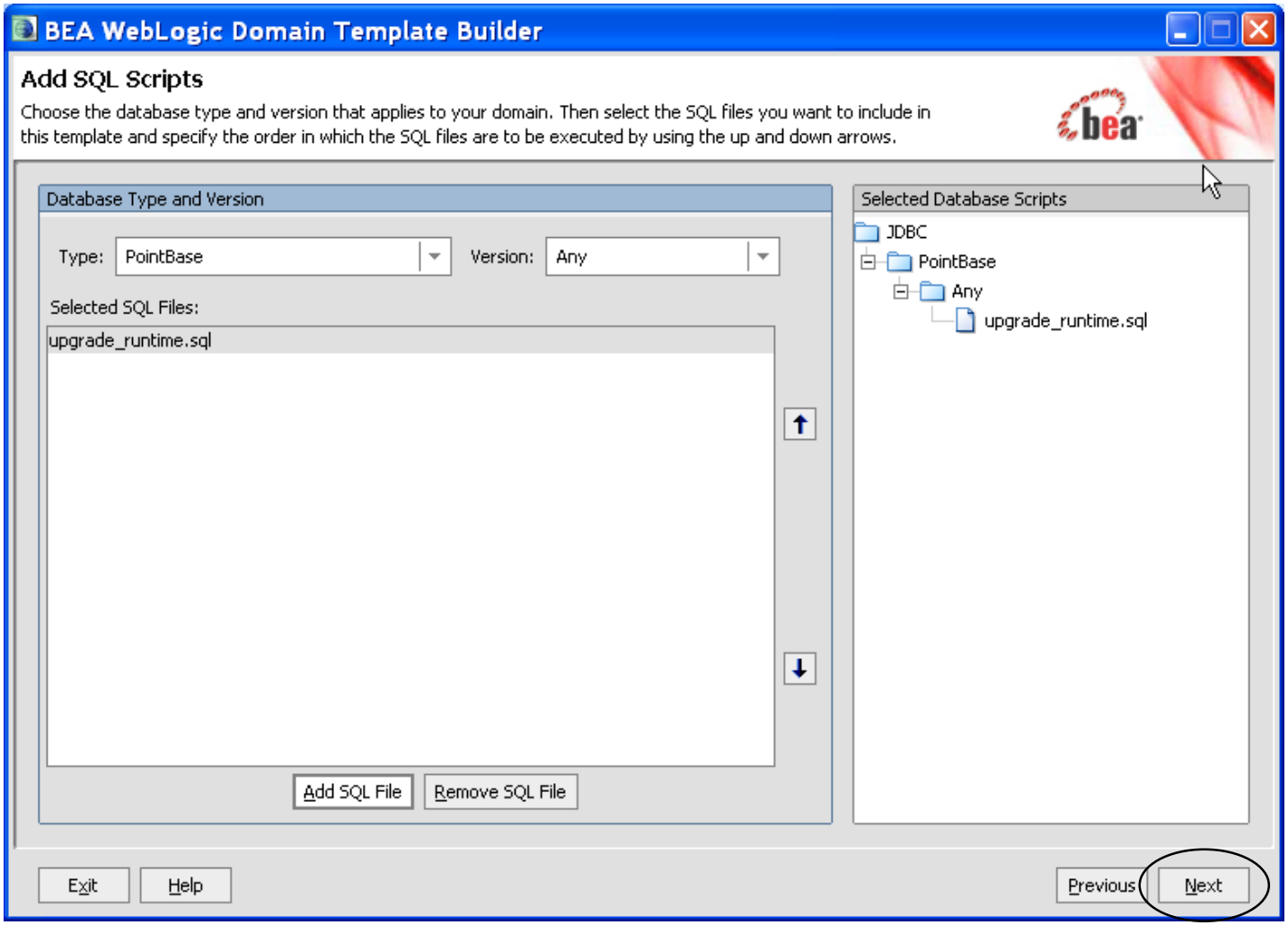


Add Files to the Template

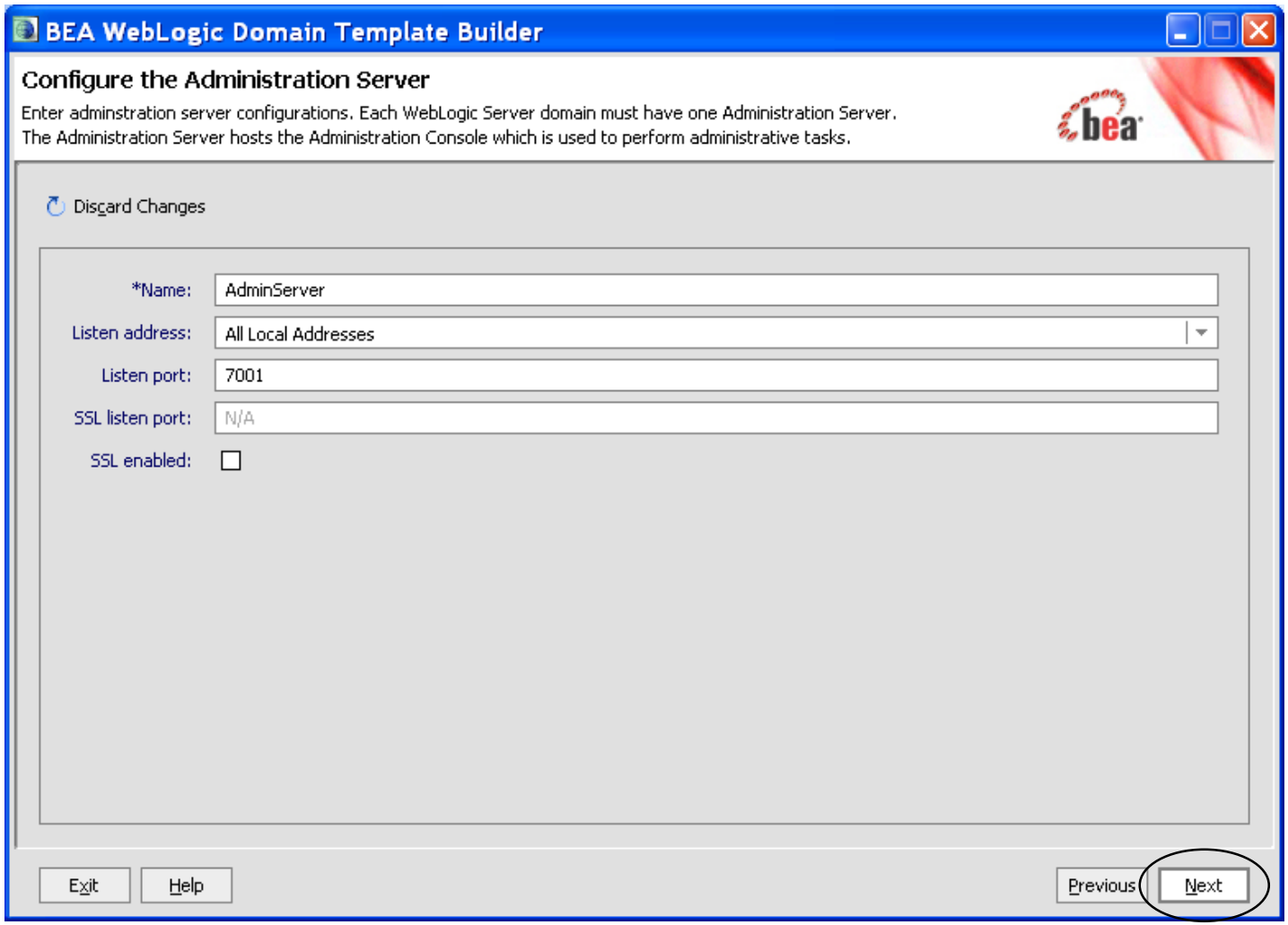


Add script to domain

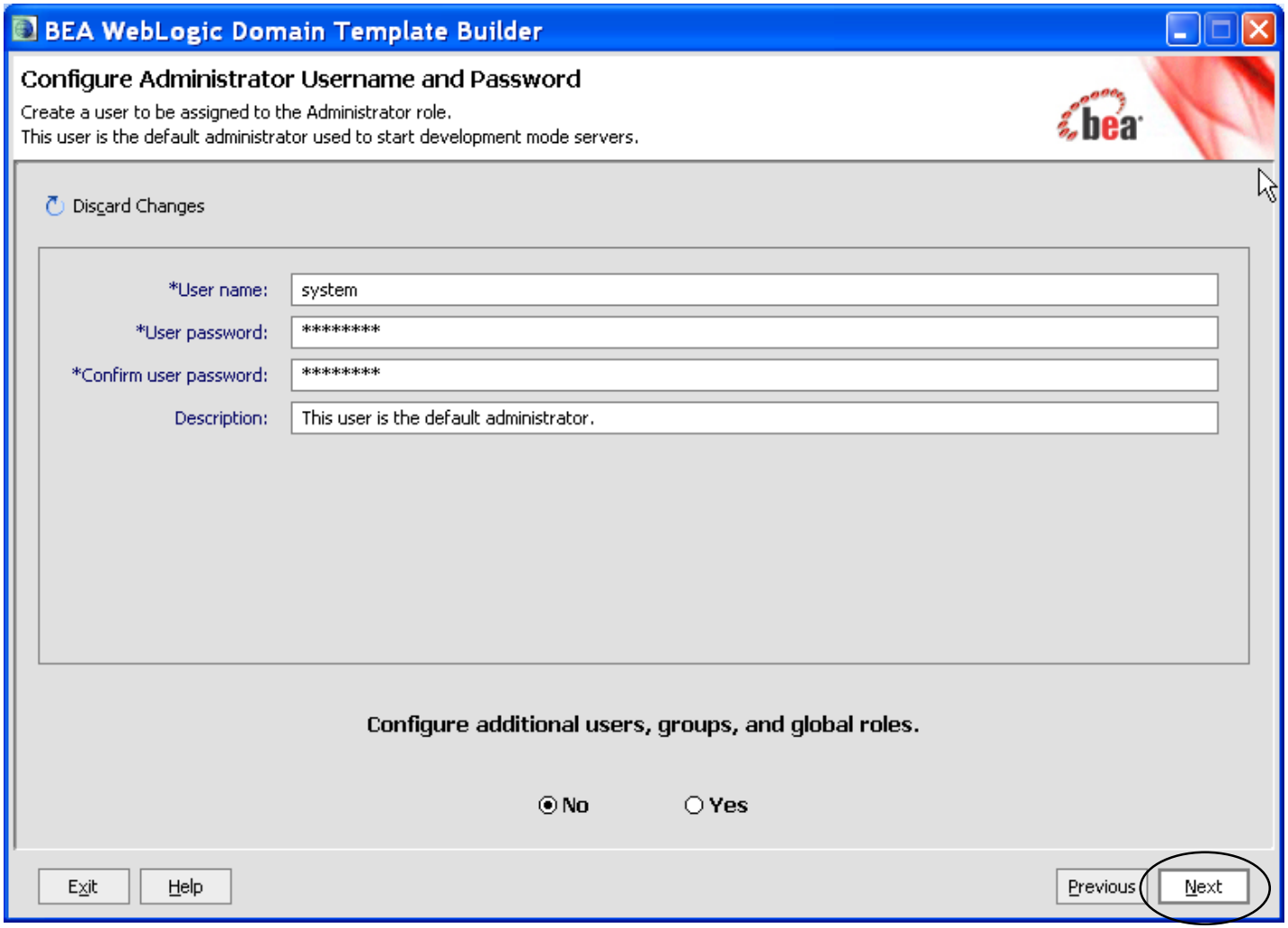
Add SQL Files to the Template



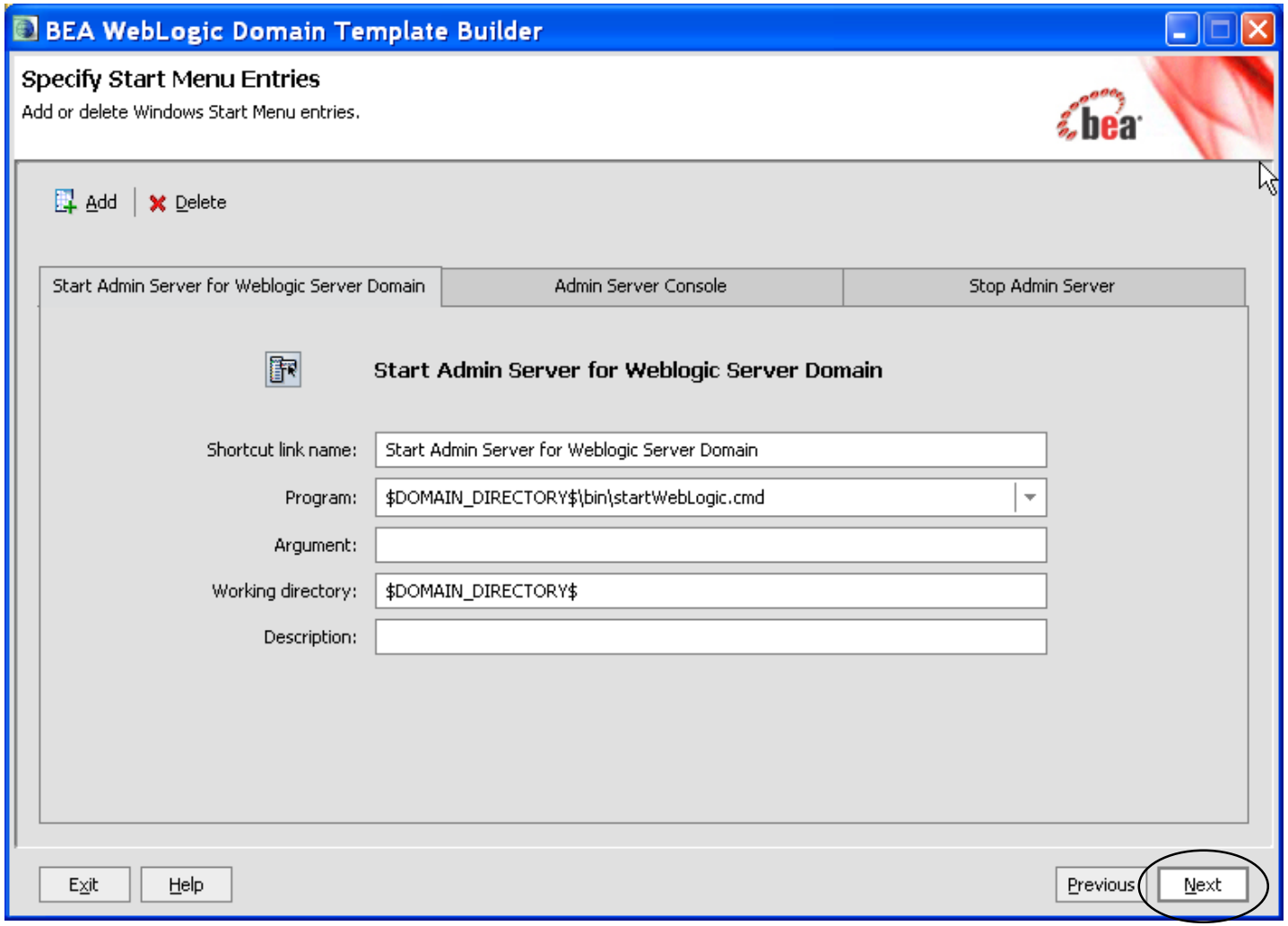
Configure the Administration Server



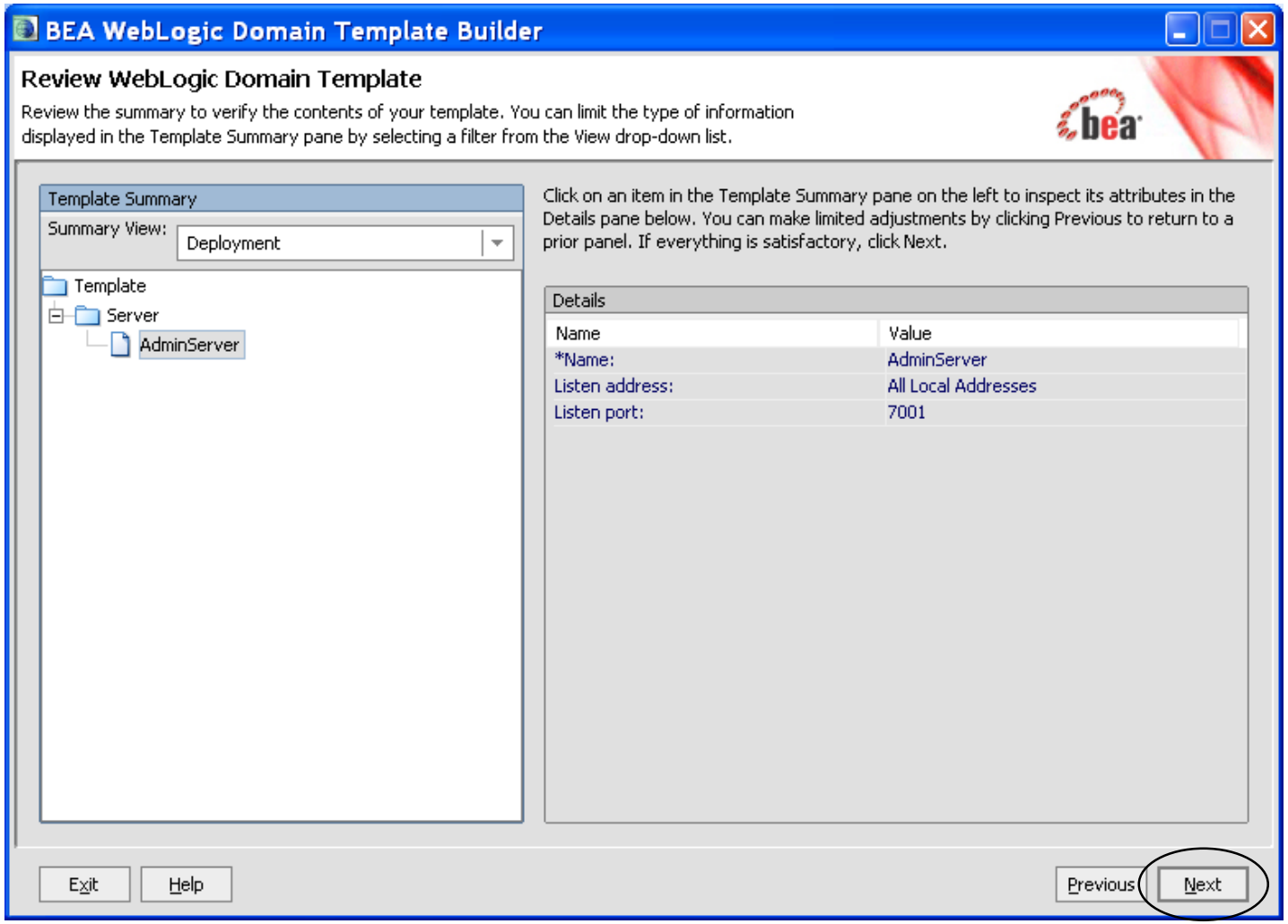
Configure Administrator Username and Password



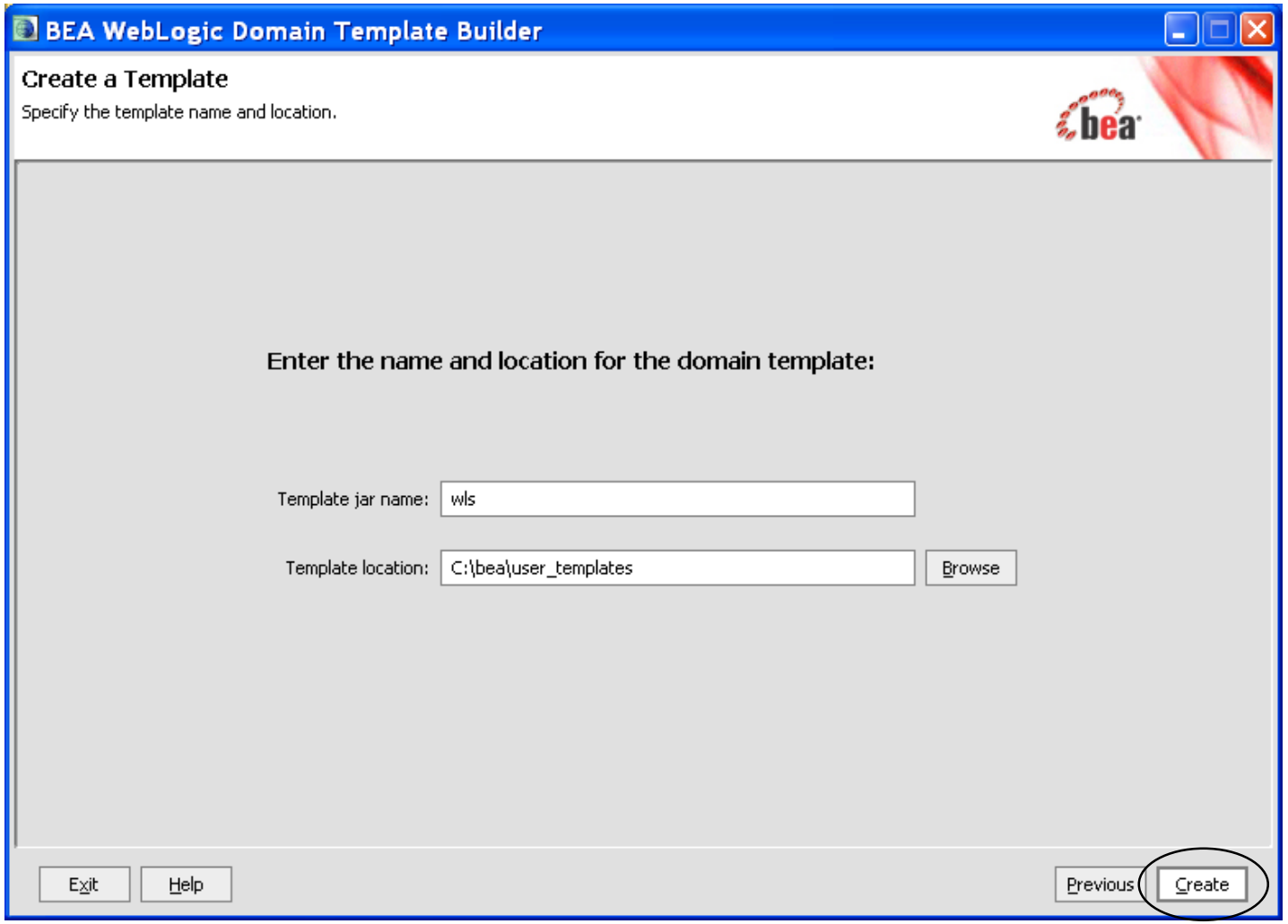
Specify Start Menu Entries



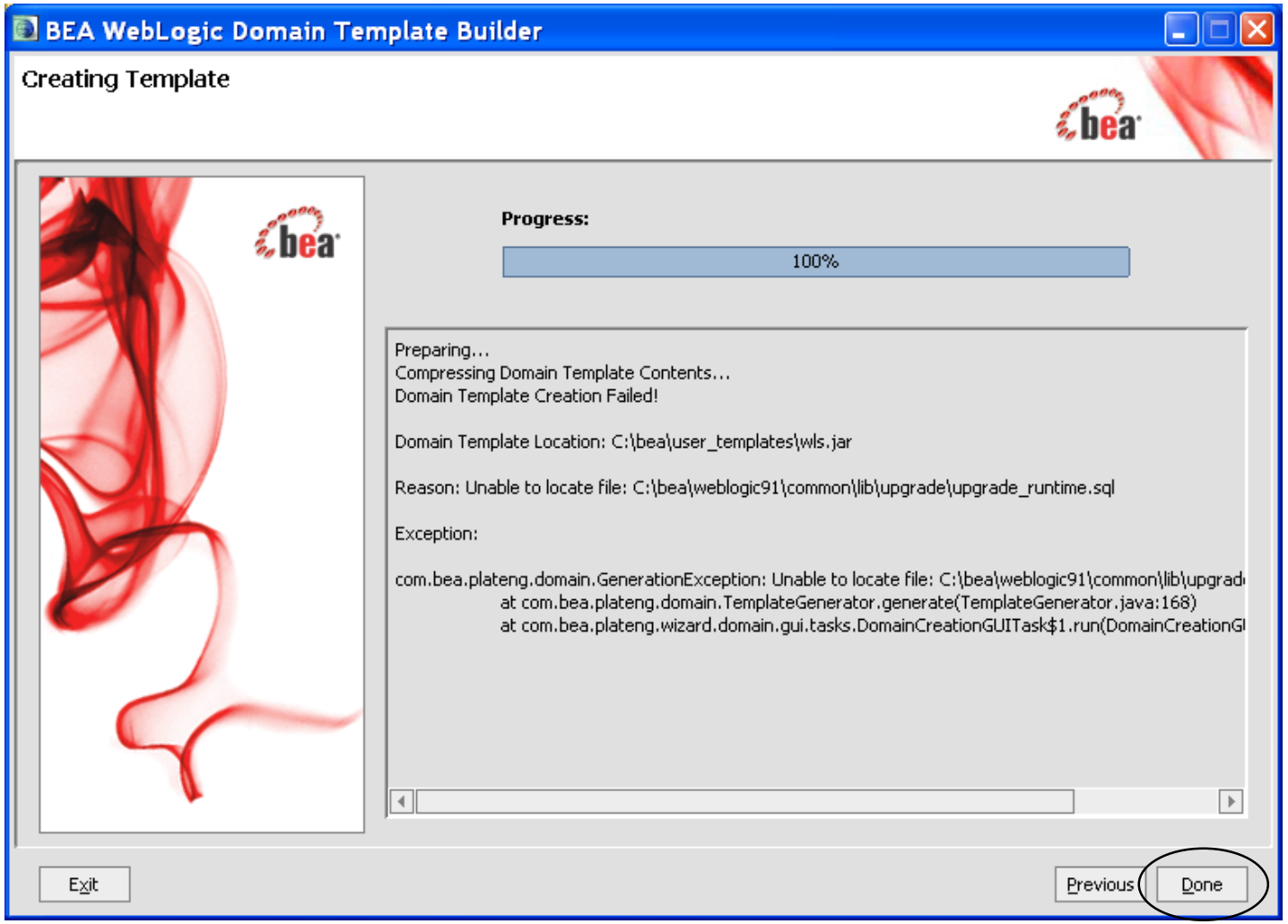
Review Domain Template



Create Template



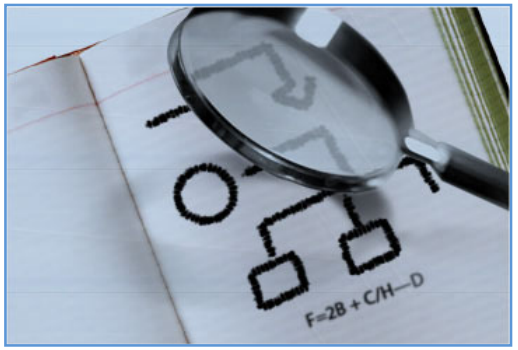
Domain Template Created



Section Review

In this section, we learned how to:

- ✓ Understand the benefits of using a custom domain template
- ✓ Create a custom domain template



Exercise

Create a Domain Template

- For details on the exercise, refer to the Lab Guide.
- If questions arise, ask the instructor.
- The instructor will determine the stop time.

Road Map

1. Configuring Domains
2. Configuring Servers
3. Domain Templates
4. **Console Administration**
   - WebLogic Server Administration Console
   - Setting Basic Properties Via Console
5. Command Line Administration

Administration Using the Console

- Using the Administration Console you can:
  - Configure attributes of resources
  - Deploy applications or components
  - Configure, collect and view diagnostic information
  - Start and shutdown servers, or perform other management actions

Starting the Console

- After starting the administration server, you can start the console in the browser of your choice:

Starting the Administration Console:

`http://hostname:port/console` ◄····· (unsecure)

`https://hostname:secureport/console` ◄····· (secure)

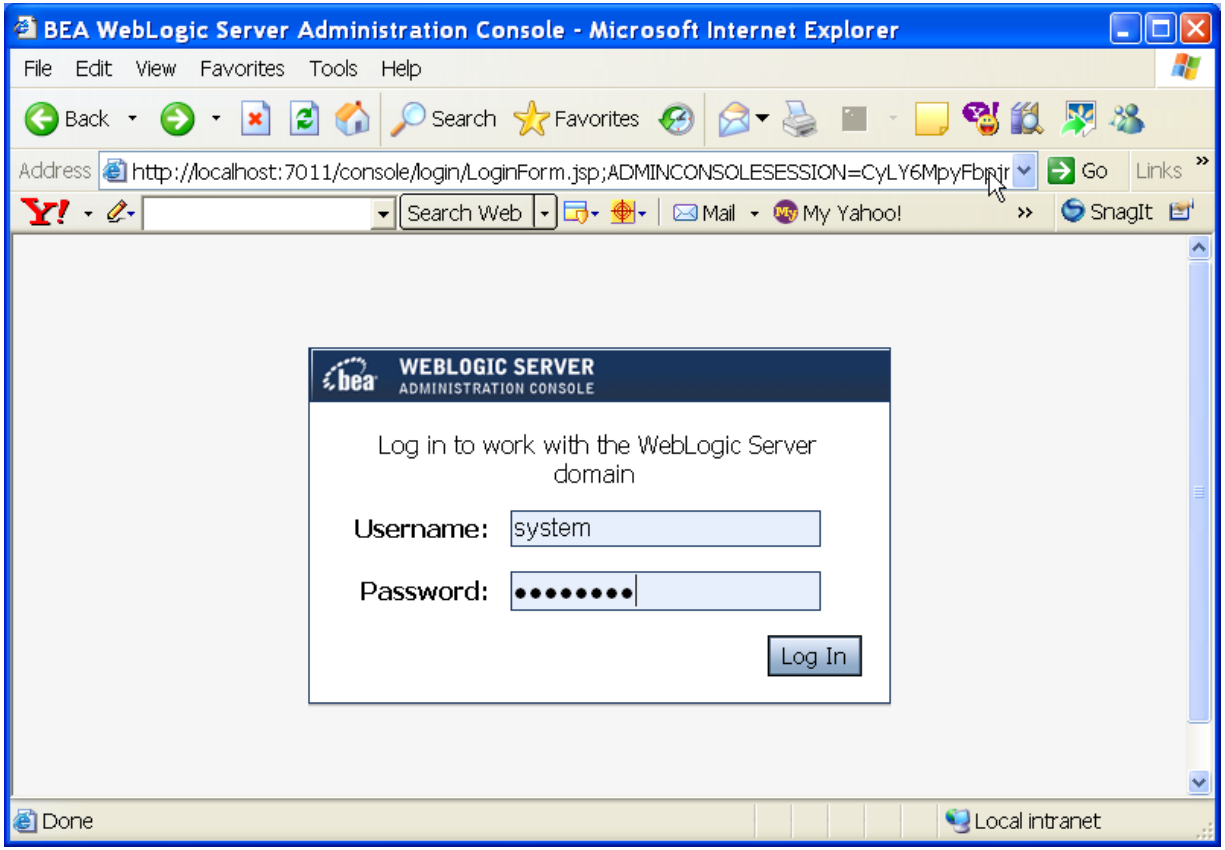
hostname := name or IP address of the administration server
port := port number the administration server is listening on
secureport := SSL port number the administration server is listening on

Example URLs:

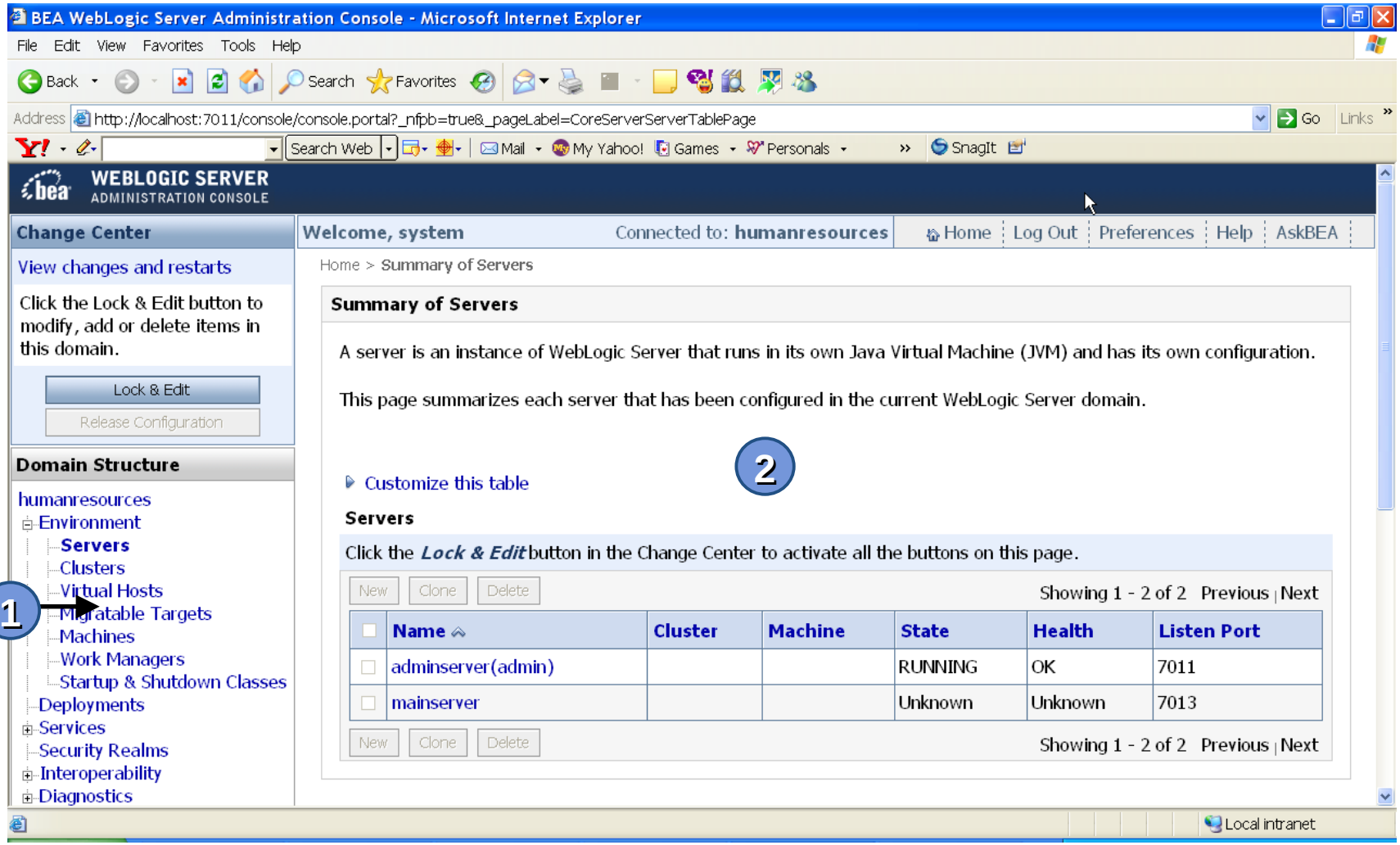
```
http://localhost:7001/console
http://adminDNSName:7001/console
https://127.0.0.1:7002/console
```

Console Login

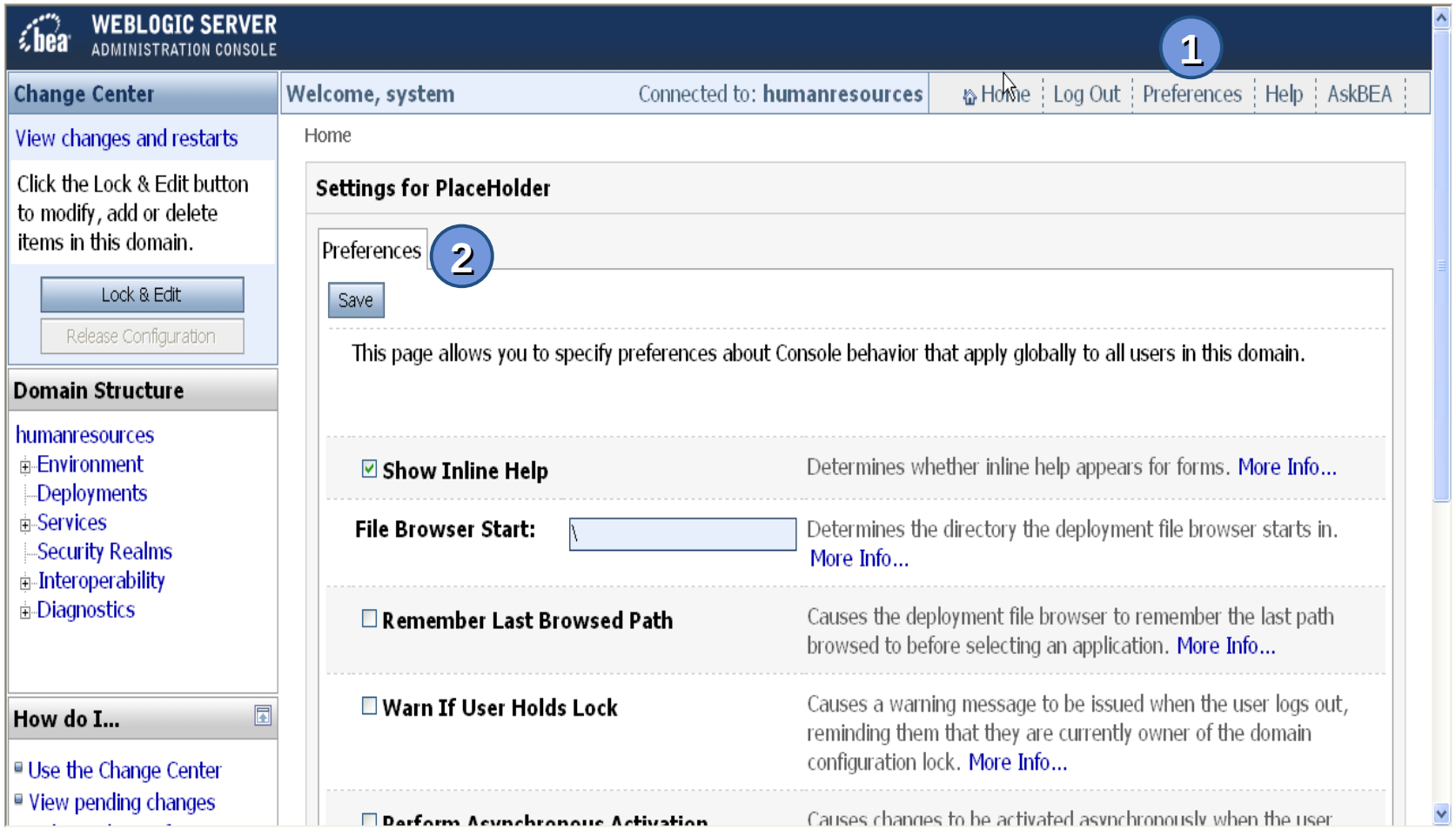
- Enter the user name and password that you set when creating your domain.



Using the Administration Console

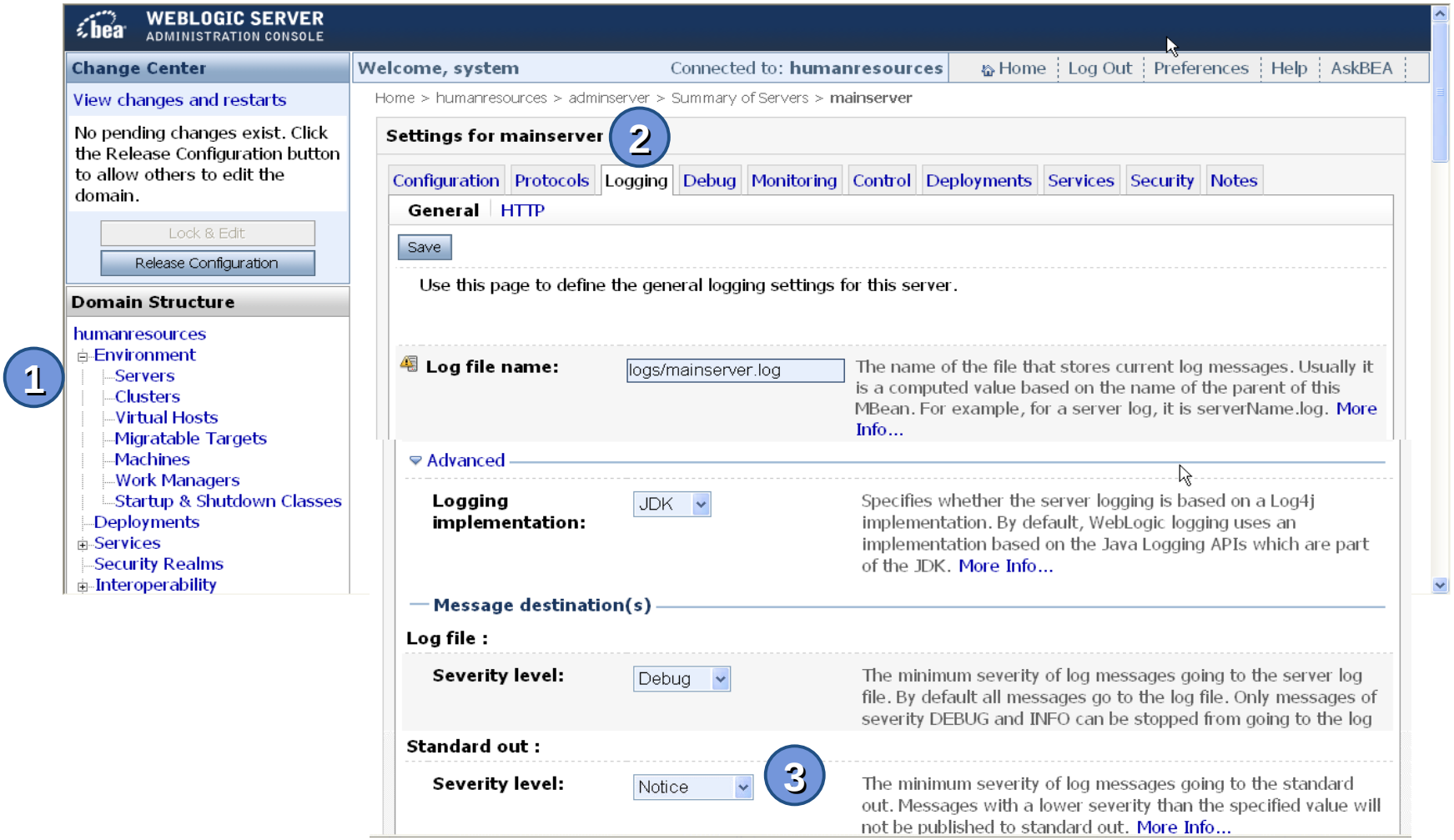


Using the Administration Console…

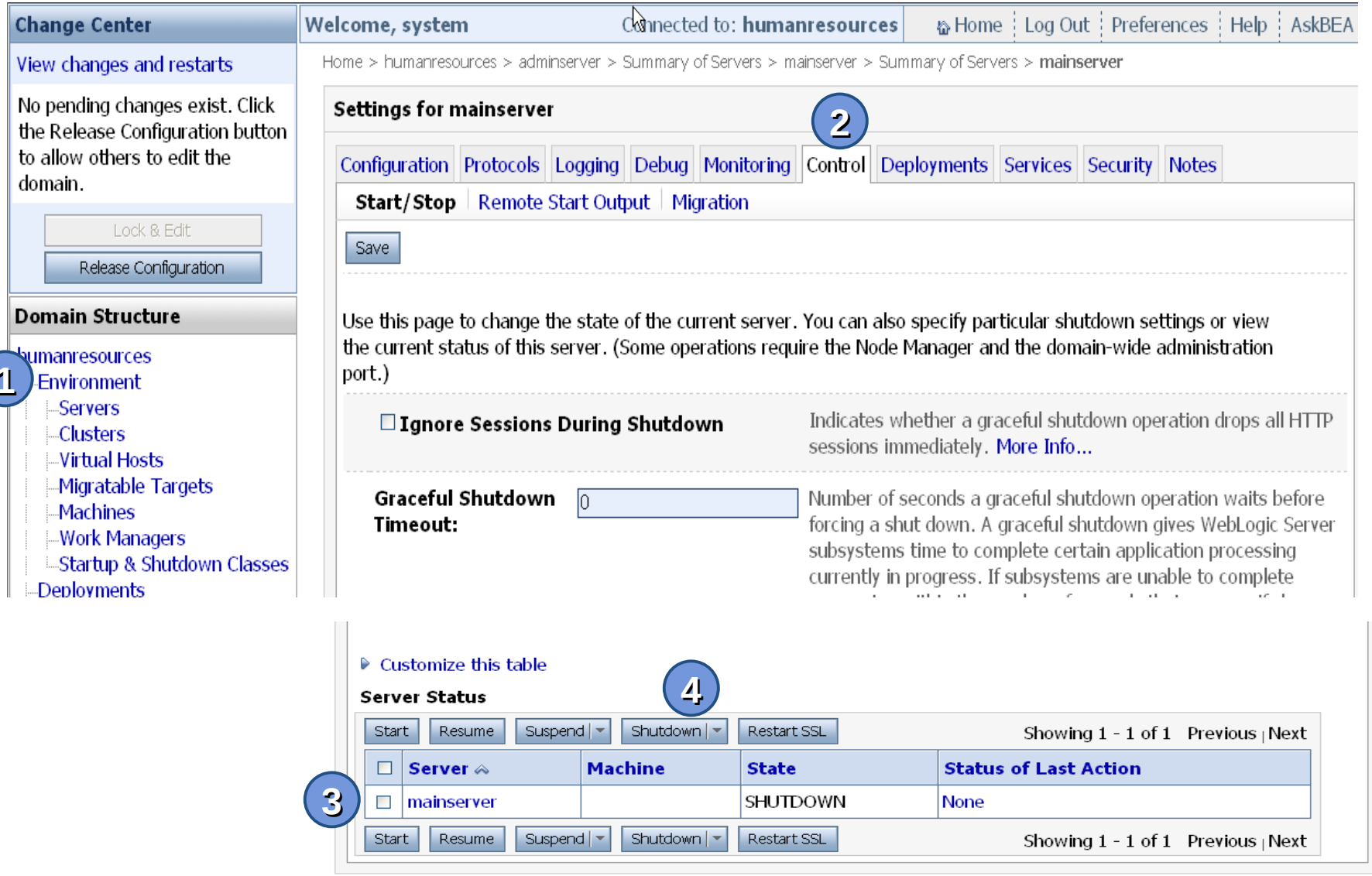


Setting Basic Properties

- Changing Stdout severity threshold:

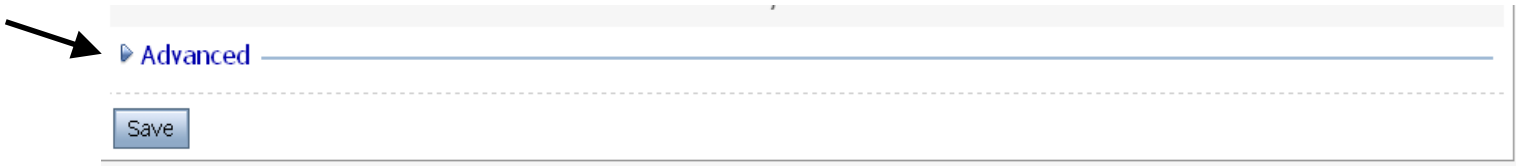


Shutting Down a Server

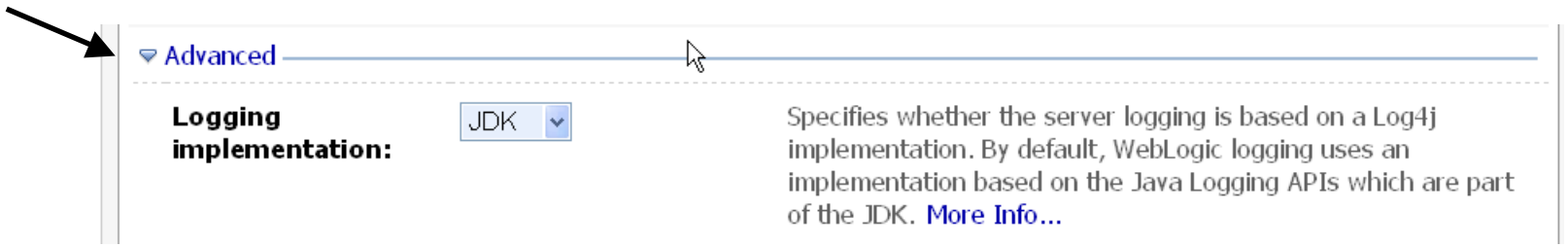


Advanced Options in Console

- The Weblogic Server Administration Console hides options which are not frequently used
- To display the Advanced Options section, click the "Advanced" link.



- If you do not want to see the Advanced Options on the console display, click the "Advanced" link one more time.



XML Schema for `config.xml`

- The `config.xml` file adheres to an XML schema that can be used for validation
- `config.xml` aggregates configuration information from other configuration files representing WLS subsystems, which adhere to their own XML schemas
- `config.xml` is now located (by default) in the `user_projects/domains/domain_name/config` folder
- Subsidiary configuration files are located in subfolders

Configuration Directory Structure

Directory	Column Head
config	The current domain configuration and deployment state (**`config.xml`**).
configCache	Cached configuration information.
deployments	The staging area for staged applications.
diagnostics	System modules for instrumentation in the WebLogic Diagnostic Service.
jdbc	System modules for JDBC.
jms	System modules for JMS.
lib	Jar files added to the system classpath of the server.
nodemanager	Node Manager configuration information.
security	System modules for the security framework.
startup	System modules that contain startup plans. Startup plans are used to generate shell scripts that can be used as part of server startup.

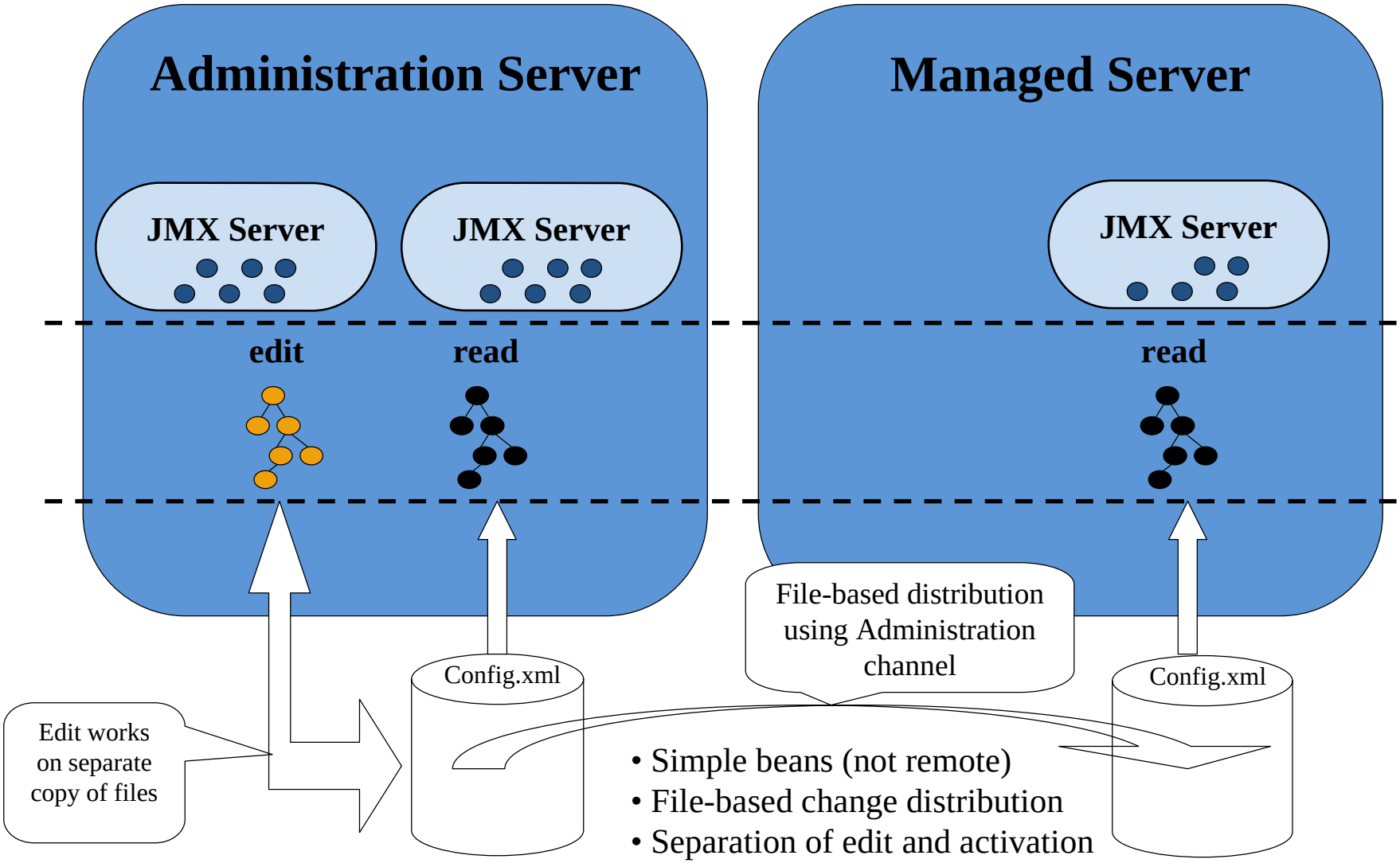
Predictable Distribution of Domain Configuration Changes

- Change management features of WLS enables you to distribute configuration changes throughout a domain securely, consistently, and predictably.
- Change management behavior is the same, regardless of whether you are using:
  - The WLS Administration Console
  - The new WebLogic Scripting Tool
  - JMX
- To use change management, use the *Change Center* region in the WLS Administration Console.

The Configuration Change Process

- WLS configuration change management process loosely resembles a database transaction.
- The domain configuration is represented two ways:
  - On the file system by a set of XML configuration files, centralized in the `config.xml` file
  - At run time by a hierarchy of Configuration MBeans
- When you edit the domain configuration, you edit a separate hierarchy of Configuration MBeans that resides on the administration server.
- When you activate changes, it's a 2-phase commit (2PC) process:
  - Each server determines whether it can accept the change.
  - If all servers are able to accept the change, they update their working configuration hierarchy and the change is completed.

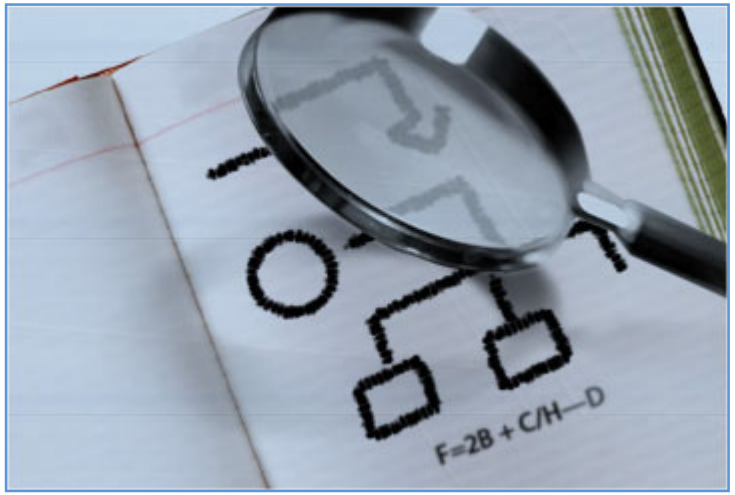
Configuration Management Architecture



Section Review

In this section we discussed:

- ✓ How to start the Administration Console
- ✓ Setting basic properties using the console



Exercise

Using the Administration Console

- In this lab you will configure and stop WLS using the Console.
- Ask the instructor for any clarification.
- The instructor will determine the stop time.

Lab Exercise

Road Map

1. Configuring Domains
2. Configuring Servers
3. Domain Templates
4. Console Administration
5. **Command Line Administration**
   - Managing WebLogic Server Via the Command Line WLST tool

The WebLogic Scripting Tool (WLST)

- Command-line tools are useful:
  - For automating administration using scripts
  - As an alternative to the Administration Console
- WLST provides a command-line interface that:
  - Configures WLS instances and domains
  - Manages and persists WLS configuration changes

WLST

- WLST enables you to:
  - Retrieve domain configuration and runtime information
  - Edit the domain configuration and persist the changes in `config.xml`
  - Navigate and edit custom, user-created management beans (MBeans) and non-WLS MBeans such as WebLogic Integration Server and WebLogic Portal Server MBeans
  - Automate configuration tasks and application deployment (repeatability)
  - Clone WebLogic Server domains
  - Access Node Manager and start, stop, and suspend server instances remotely or locally, without requiring the presence of a running Administration Server

Built on Jython

- Jython advantages include:
  - 100% pure Java implementation of Python
  - Simple & clear syntax
  - Proven to be fast and reliable
  - Highly extensible (create your own commands, use any existing Java classes)
- WLST interprets commands in two ways:
  - Interactively, supplied one-at-a-time from a command prompt
  - In batch supplied in a file (script), or embedded in your Java code

Main Steps for using WLST

- Setting environment
- Invoking WLST
- Requirement for entering WLST commands
- Running Scripts
- Importing WLST as a Jython Module
- Exiting WLST

Setting Environment

- Install and configure the WebLogic Server software
- Add WebLogic Server classes to the CLASSPATH environment variable
- Add *WL_HOME*\server\bin to the PATH environment variable

Invoking WLST

- Connecting to WLS through SSL listen port
  - java -Dweblogic.security.SSL.ignoreHostnameVerification=true -Dweblogic.security.TrustKeyStore=DemoTrust weblogic.WLST
- Not connecting to WLS through the SSL listen port
  - java weblogic.WLST
  - A welcome message and WLST prompt appears:

    wls:/(offline)>
  - To use WLST online:
    connect('*username*','*password*','*t3s://localhost:7012*')
  - To shut down a server, for example:
    shutdown('myserver','Server','false',1000)
  - To exit WLST:
    exit()

Requirement for Entering WLST commands

- Case sensitive names and arguments of commands
- Arguments enclosed in single or double quotes
- Precede the quoted string by **r** while specifying backslash (\) in a string
  - Example: readTemplate(r'c:\mytemplate.jar')
- Invalid characters in object names while using WLST offline:
  - Period (.)
  - Forward slash (/)
  - Backward slash (\)
- Cannot access security information through WLST while updating a domain
- Display help
  - Example: **wls:/mydomain/serverConfig> help('disconnect')**

Running Scripts

- WLST incorporates two Jython functions that support running scripts:
  - java weblogic.WLST *filePath*.py, which invokes WLST and executes a script file in a single command,
  - execfile("*filePath*.py") which executes a script file after you invoke WLST.

Import WLST as a Jython Module

1) Invoke WLST:

```
c:\>java weblogic.WLST
wls:/offline>
```

2) Use the writeIniFile command to convert WLST definitions and method declarations to a .py file:

```
wls:/offline> writeIniFile("wl.py")
```

3) Open a new command shell and invoke Jython directly by entering the following command:

```
c:\>java org.python.util.jython
```

4) Import the WLST module into your Jython module using the Jython import command:

```
>>>import wl
```

5) Now you can use WLST methods in the module. For example, to connect WLST to a server instance:

```
wl.connect('username','password')
```

Exiting WLST

Exit WLST:

```
wls:/mydomain/serverConfig> exit()
Exiting WebLogic Scripting Tool ...
c:\>
```

Online and Offline Modes

- Online (connected to a running server):
  - WLST provides simplified access to MBeans.
  - You can perform administrative tasks and initiate WLS configuration changes while connected to a running server.
- Offline (not connected to a running server):
  - WLST limits access to only persisted configuration information.
  - You can create a new domain or update an existing domain without connecting to a running WLS—this functionality resembles that of the Configuration Wizard.

Modes of Operation

- Interactive
  - When you enter a command in the WLST console and view the response immediately
- Script
  - When you create a text file, with .py extension, that contains a series of WLST commands
- Embedded
  - When you instantiate an instance of the WLST interpreter in your Java code and use it to run WLST commands

Simplified Command-Line Access

- WLST includes the capabilities of:
  - `weblogic.Admin` (deprecated in 9.X)
  - `weblogic.Deployer`
  - `wlconfig` Ant tasks
  - Configuration wizard (silent mode, deprecated in 9.X)
- It allows you to navigate the WLS MBean tree like a file system.
- To access WLST:
  - In a non-secure environment use: `java weblogic.WLST`
  - In a secure environment use:

```
java -Dweblogic.security.SSL.ignoreHostnameVerification=true
-Dweblogic.security.TrustKeyStore=DemoTrust weblogic.WLST
```

WLST Example

```
C:\>java weblogic.WLST

Initializing WebLogic Scripting Tool (WLST) ...

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

wls:/offline> connect('system','weblogic','t3://localhost:7011')

Connecting to weblogic server instance running at t3://localhost:7011 as usernam
e system ...

Successfully connected to Admin Server 'adminserver' that belongs to domain 'hum
anresources'.

Warning: An insecure protocol was used to connect to the server.
To ensure on-the-wire security, the SSL port or Admin port
should be used instead.

wls:/humanresources/serverConfig> cd('Servers')
wls:/humanresources/serverConfig/Servers> ls()
dr--   adminserver
dr--   mainserver

wls:/humanresources/serverConfig/Servers> cd('mainserver')
wls:/humanresources/serverConfig/Servers/mainserver> get('StartupMode')
'RUNNING'
wls:/humanresources/serverConfig/Servers/mainserver> exit()

Exiting WebLogic Scripting Tool ...
```

Some WLST Commands…

Command	Description	Syntax
`connect`	**Connects to a server instance (online mode)**	`connect('weblogic', 'weblogic', 't3://localhost:7011')`
`disconnect`	**Disconnects from a server instance (online mode)**	`disconnect()`
`exit`	**Exits WLST (online mode)**	`exit()`
`readDomain`	**Opens an existing domain for updating (offline mode)**	`readDomain('c:/bea/user_projects/domains/onlinestore')`
`updateDomain`	**Updates and saves the current domain (offline mode)**	`updateDomain()`

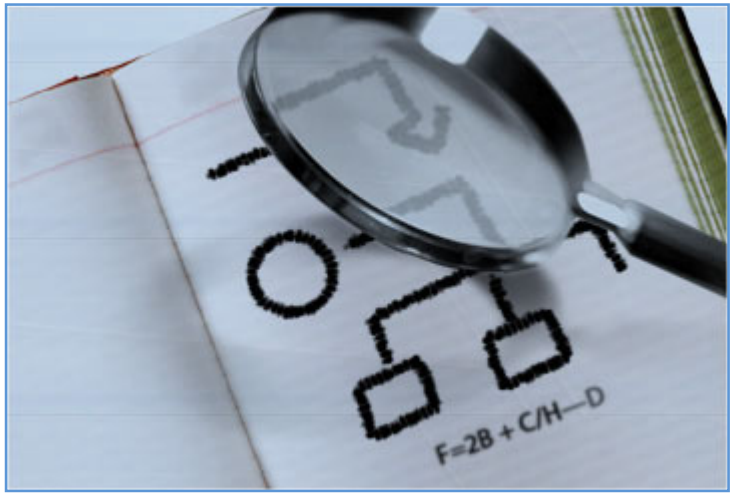
…Some WLST Commands

Command	Description	Syntax
`start`	**Start a managed server instance or a cluster (online mode)**	`start('mycluster', 'Cluster')`
`suspend`	**Suspends a running server (online mode)**	`suspend('mainserver')`
`shutdown`	**Gracefully shuts down a WLS instance or a cluster (online mode)**	`shutdown('myCluster', 'Cluster')`
`startServer`	**Starts the administration server (offline mode)**	`startServer('adminserver', 'humanresources', 't3://localhost:7011')`
`resume`	**Resumes a server that is in the ADMIN state (online mode)**	`resume('mainserver')`

Section Review

In this section we discussed:

- ✓ Managing WebLogic Server from the command-line



Exercise

Using Command-line Administration

- In this exercise you are going to gain experience using the command-line administration utility.
- Ask the instructor for any clarification.
- The instructor will determine the stop time.

Lab Exercise

Module Review

In this module we discussed:

- ✓ The WebLogic Server Administration Console
- ✓ Domain concepts
- ✓ How to create domains
- ✓ How to create and start Administration server and managed servers
- ✓ The creation and use of domain templates