

6

Using Administration Console and WLST

Objectives

After completing this lesson, you should be able to:

- Start the WebLogic Server Administration Console
- Explain the layout and navigation of the Administration Console
- Access context-sensitive help within the Administration Console
- Customize Administration Console preferences
- Use the Administration Console breadcrumb trail
- Create, commit, undo, and monitor a change session using the Administration Console
- Access advanced resource attributes in the Administration Console

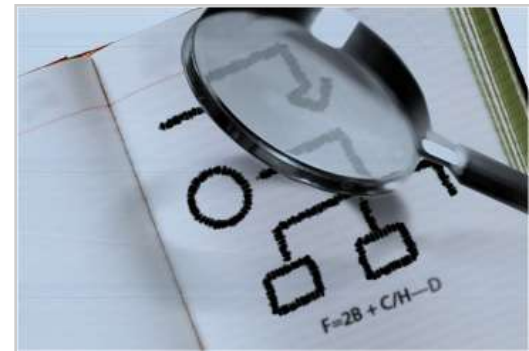
Objectives

After completing this lesson, you should be able to:

- Identify dynamic and nondynamic attribute changes in the Administration Console
- Customize monitoring tables within the Administration Console
- Use the WebLogic Scripting Tool (WLST) command-line utility
- Invoke WLST and navigate through the domain structure
- Describe how change management is performed internally using JMX and MBeans

Road Map

- Performing Administration Console configuration
 - Navigation
 - Help
 - Setting preferences and properties
- Performing command-line configuration



Benefits of Using the Administration Console

Using the Administration Console, you can:

- Configure attributes of servers and their resources
- Deploy and secure applications
- Configure, collect, and view diagnostic information
- Start and shut down servers or perform other management actions



Accessing the Administration Console

After starting the administration server, you can access the Administration Console in a Web browser of your choice.

```
http://[hostname]:[port]/console
```

```
https://[hostname]:[secureport]/console
```

hostname = The name or IP address of the Administration Server

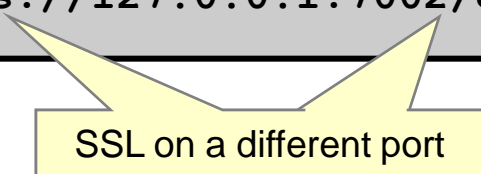
port = The port number that the Administration Server listens on

secureport = The SSL port number that the Administration Server listens on

```
http://localhost:7001/console
```

```
http://adminDNSName:7001/console
```

```
https://127.0.0.1:7002/console
```



SSL on a different port

Administration Console Login

Enter the username and password that you set when creating your domain. The password is not displayed.

ORACLE
WebLogic Server® 11g
Administration Console

Welcome
Log in to work with the WebLogic Server domain

Username:

Password:

Member of the Administrators group

WebLogic Server Version: 10.3.1.0
Copyright © 1996,2009, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Basic Navigation

The screenshot displays the Oracle WebLogic Server Administration Console interface. Key navigation elements are highlighted with callouts and numbered circles:

- Help system:** A callout points to the 'Help' link in the top navigation bar.
- Breadcrumb:** A callout points to the breadcrumb path: Home > MedRecDomain > Summary of Servers > MedRecAdmSvr.
- Domain Structure:** A callout points to the 'Domain Structure' tree on the left, which shows the hierarchy of the MedRecDomain. A green circle with the number '1' is placed next to this tree.
- Change Center:** A callout points to the 'Lock & Edit' button in the Change Center, which is used to modify settings. A green circle with the number '2' is placed next to this button.
- Tabs:** A callout points to the 'HTTP' tab under the 'Protocols' section, indicating the current configuration page. A green circle with the number '2' is placed next to this tab.

The interface also shows the 'Settings for MedRecAdmSvr' page, which includes tabs for Configuration, Protocols, Logging, Debug, Monitoring, Control, Deployments, Services, and Security. The 'HTTP' tab is selected, and the page content describes the HTTP settings for the server.

Using the Help System

ORACLE WebLogic Server® Administration Console

Home Log Out Preferences Record Help

Welcome, weblogic Connected to: MedRecDomain

Home > Summary of Deployments > Summary of Security Realms > myrealm

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings Providers Migration

General RDBMS Security Store User Lockout Performance

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

Save

Use this page to configure the general behavior of this security realm.

Note:
If you are implementing security using JACC (Java Authorization Contract for Containers as defined in JSR 115), you must use the DD Only security model. Other WebLogic Server models are not available and the security functions for Web applications and EJBs in the Administration Console are disabled.

Name: myrealm The name of this security realm. [More Info...](#)

Security Model DD Only Specifies the default security model for Web applications or EJBs that are secured by this security realm. You can override this default during deployment. [More Info...](#)

☒ **Combined Role Mapping Enabled** Determines how the role mappings in the Enterprise Application, Web application, and EJB containers interact. This setting is valid only for Web applications and EJBs that use the Advanced security model and that initialize roles from deployment descriptors. [More Info...](#)

Change Center

View changes and restarts

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

Lock & Edit

Release Configuration

Domain Structure

MedRecDomain

- Environment
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

How do I...

- Manage security for Web applications and EJBs
- Set the default security model
- Delegate MBean authorization to the realm

System Status

Health of Running Servers

- Failed (0)
- Critical (0)
- Overloaded (0)
- Warning (0)
- OK (1)

General Administration Console User Preferences

Home Log Out **Preferences** Record Help

Welcome, weblogic Connected to: MedRecDomain

Home > MedRecDomain > Summary of Servers > MedRecAdmSvr > **Preferences**

Preferences

User Preferences Shared Preferences Extensions WLST Script Recording

Save

You can configure some features of the Administration Console by setting preferences.

Use this page to specify preferences that will be in effect only for your user account and only in the current domain.

User Preferences

☒ **Show Inline Help**

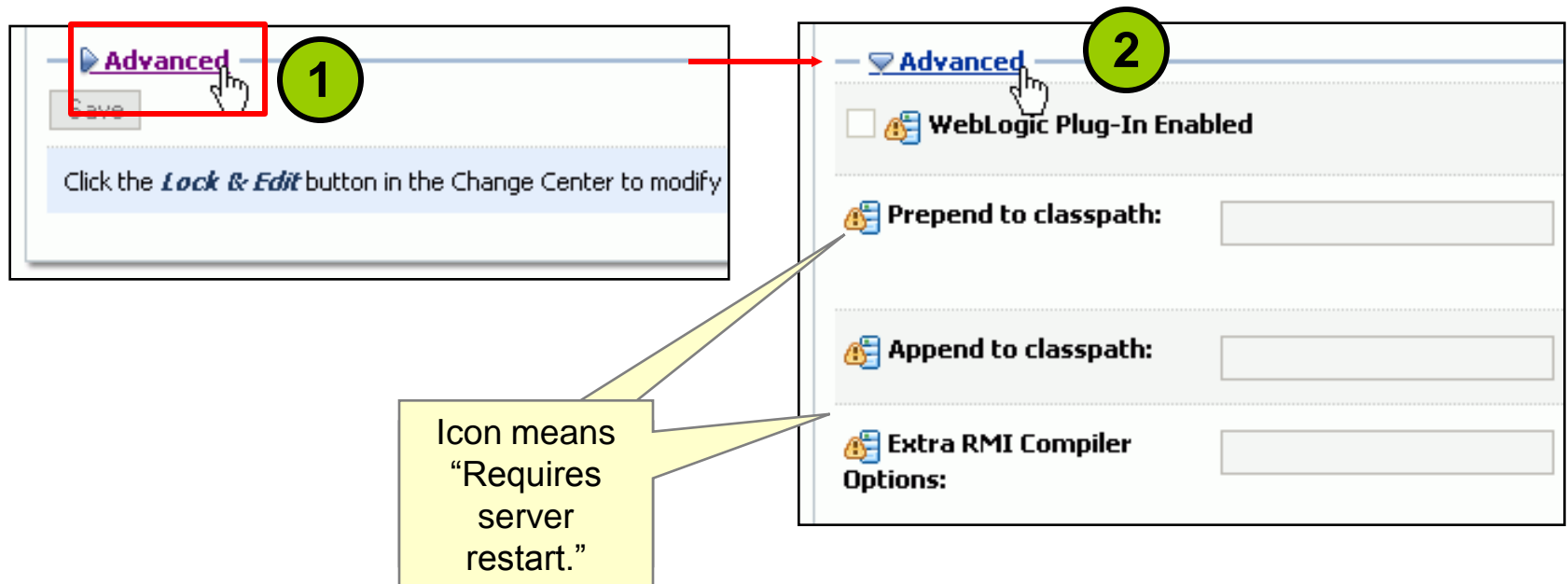
File Browser Start: \

☒ **Remember Last Used Path**

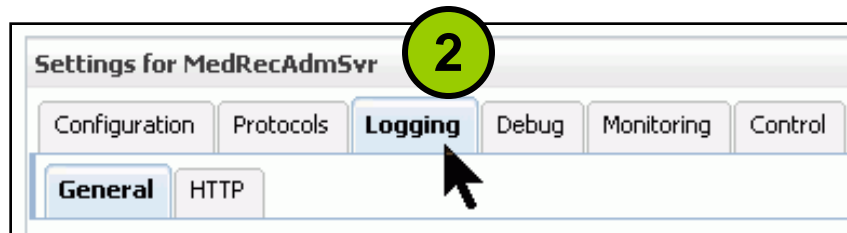
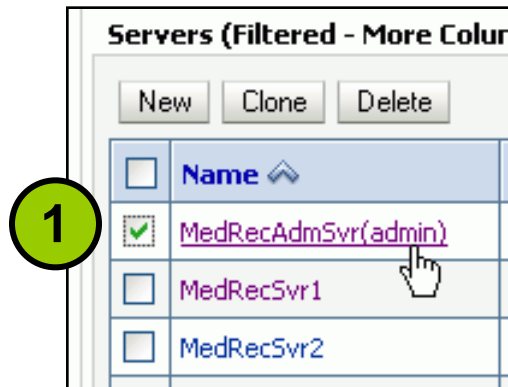
Paths To Remember: 4

Advanced Console Options

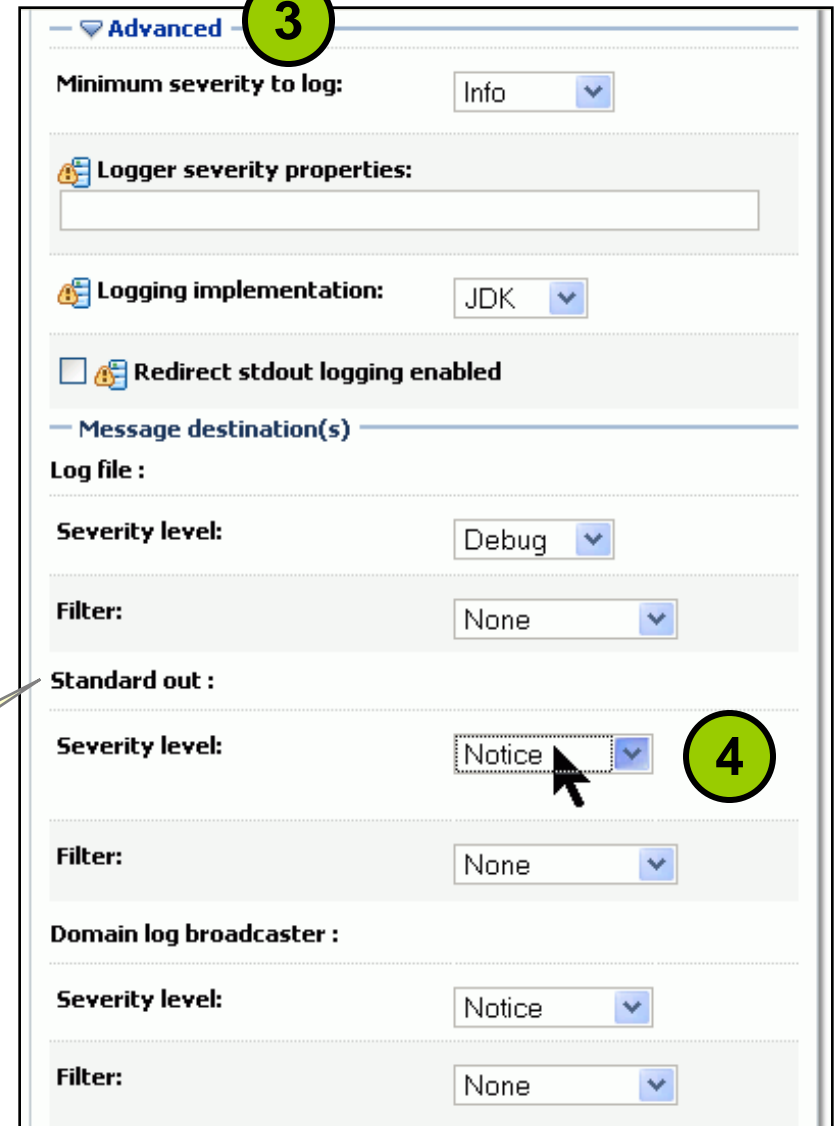
In a screen's Advanced section, the console shows or hides the options that are not frequently used.



Setting Basic Properties



Changing the
Standard out
severity threshold



Administration Console Monitoring

The Administration Console offers many monitoring capabilities for servers, application deployments, and Java EE services.

Settings for MedRecAdmSvr

Configuration Protocols Logging Debug **Monitoring** Control Deployments Services Security Notes

General Health Channels Performance **Threads** Timers Workload Security Default Store JMS SAF

Dump Thread Stacks

This page provides information on the thread activity for the current server.

[Customize this table](#) Customize display data columns.

Self-Tuning Thread Pool (Filtered - More Columns Exist)

Showing 1 to 1 of 1 Previous | Next

Active Execute Threads	Execute Thread Total Count	Execute Thread Idle Count	Queue Length	Pending User Request Count	Completed Request Count	Hogging Thread Count	Standby Thread Count	Throughput	Health
11	12	10	0	0	97618	0	1	1.998001998001998	OK

Showing 1 to 1 of 1 Previous | Next

Configuration Change Management

For change management, use the Change Center in the WLS Administration Console to lock the configuration files.

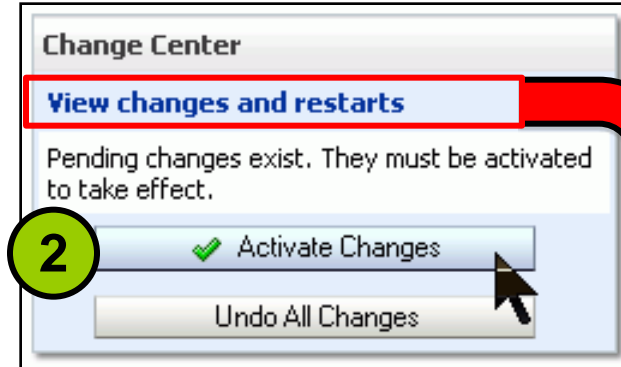
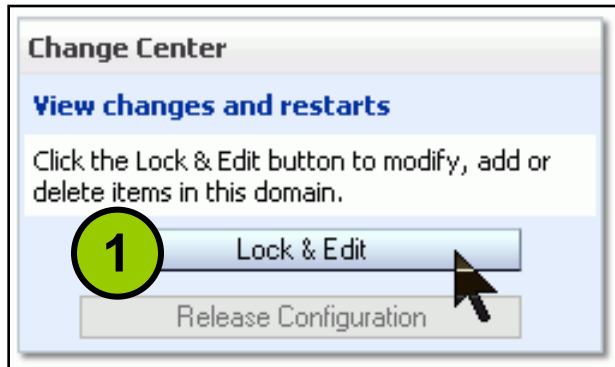
The change management features of WLS:

- Enable you to distribute configuration changes throughout a domain securely, consistently, and predictably
- Are the same, regardless of whether you are using:
 - The WLS Administration Console
 - The WebLogic Scripting Tool (WLST)
 - The Java Management Extension (JMX) APIs



ORACLE

Configuration Change Management Using the Administration Console Change Center



Changes and Restarts

Change List Restart Checklist

This page lists the pending changes currently applied to the domain, but not yet activated. Click **Activate Changes** to save all changes to the domain. Be sure to check the Restart Checklist page for a list of changes that require restarts.













⚠ You currently have a mix of changes, some requiring restart, others do not. The changes that do not require restart will not take effect until the restarts are completed.

Pending Changes

Activate Changes Undo All Changes Showing 1 to 3 of 3 Previous | Next

<input type="checkbox"/>	Change ^	Type	Description	Restart Required
<input type="checkbox"/>	GuardianEnabled	Domain attribute	Modified in MedRecDomain from false to true	Yes
<input type="checkbox"/>	ListenPort	Server attribute	Modified in MedRecAdmSvr from 7020 to 7050	
<input type="checkbox"/>	ListenPort	SSL attribute	Modified in MedRecAdmSvr from 7002 to 7021	

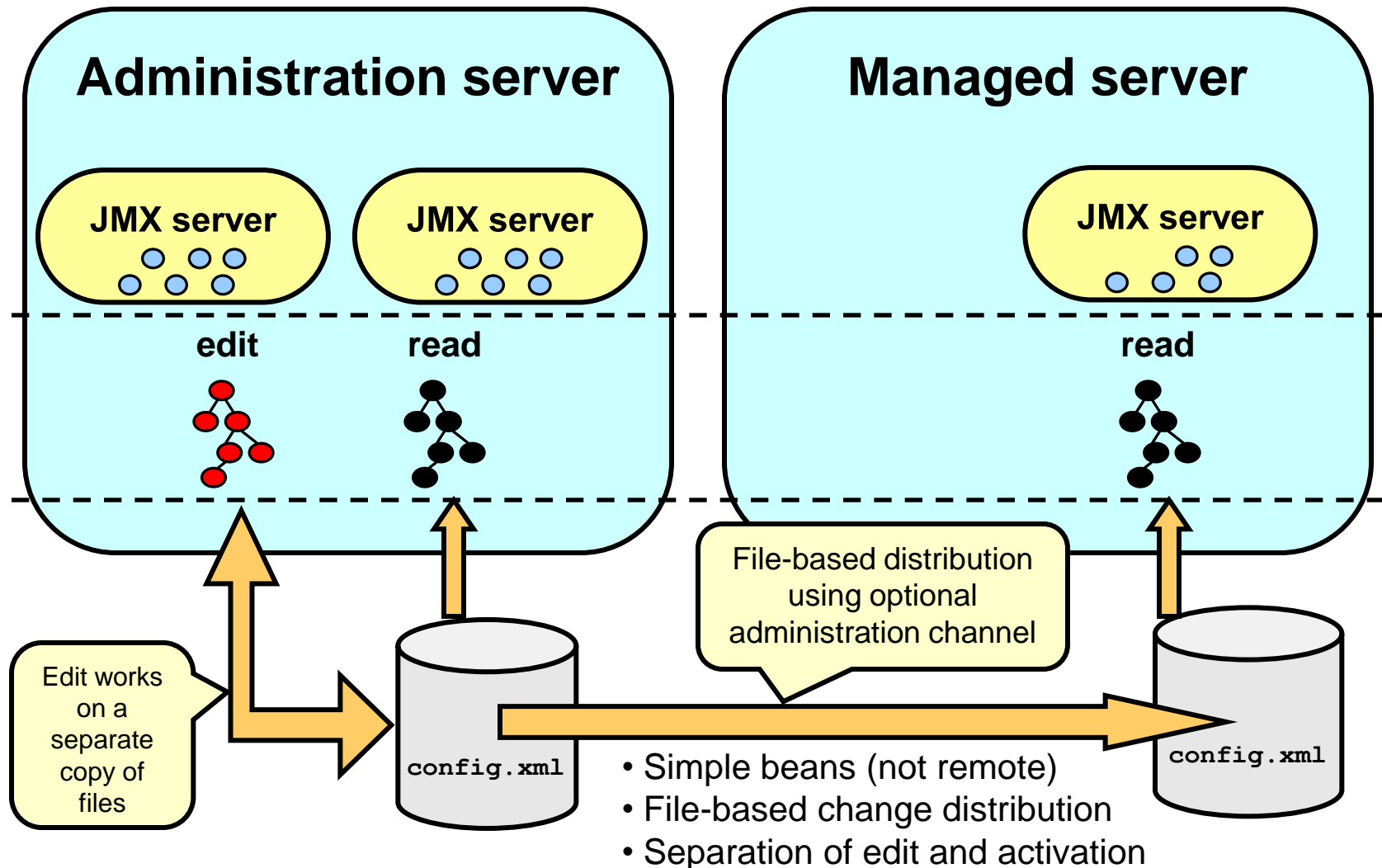
Domain Configuration Repository

Directory	Description
 config	Root folder of domain configuration files
+  configCache	Cached configuration data
+  deployments	Staging area for deployment applications
+  diagnostics	Configuration modules for diagnostics framework
+  jdbc	Configuration modules for data services
+  jms	Configuration modules for messaging services
+  lib	Not currently used. See <code><domain>/lib</code> .
+  nodemanager	Node Manager configuration files
+  security	Configuration modules for the security framework
+  startup	Any scripts to run as part of server startup
 config.lock	Lock file used for change management
 config.xml	Primary domain configuration file

Configuration Change Process

- Domain configuration is represented in two ways:
 - On the file system by a set of XML configuration files, including `config.xml`
 - At run time by a hierarchy of in-memory JMX objects
- When you activate changes, it is a two-phase 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. Otherwise, the pending changes are rolled back and nothing happens.

Configuration Management Architecture

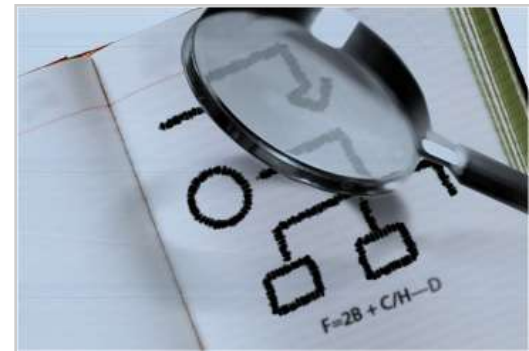


XML Schema for `config.xml`

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

Road Map

- Performing Administration Console configuration
- Performing command-line configuration
 - WLST
 - Jython



WebLogic Scripting Tool (WLST)

- The WLS command-line tools are useful:
 - For automating common administration activities
 - As an alternative to the Administration Console
 - When graphical tools are not supported
- WLST provides a command-line interface for:
 - Creating new WLS domains
 - Retrieving and updating WLS domain configurations
 - Deploying applications
 - Obtaining run-time server statistics

Jython

Jython is a Java implementation of the popular Python scripting language:

- Simple and clear syntax
- Indentation to structure code (white space critical!)
- Interactive command mode
- Custom commands
- Integration with any existing Java libraries

```
list = ['ab', 'cd', 'ef']  
  
if len(list) >= 3:  
    for x in list:  
        print x, len(x)  
    print 'done'
```

```
from java.util import ArrayList  
  
list = ArrayList()  
list.add('ab')
```

Using Jython

Jython can interpret commands in three ways:

Interactive: Supply commands one at a time from a command prompt. Enter a command in the WLST console and view the response immediately.

Batch: Provide a series of commands in a script file (`.py`) when you create a text file with the `.py` extension that contains a series of WLST commands.

Embedded: Run the Jython interpreter within a Java class when you instantiate an instance of the WLST interpreter in your Java code and use it to run WLST commands.

WLST Modes

- Online mode:
 - Connected to a running server
 - Access to all WLS configuration and run-time attributes
 - To create and activate change sessions similar to the WLS console
- Offline mode:
 - Domain not running
 - Access to only persisted domain configuration (`config.xml`)
 - To create or update domains similar to using the Configuration Wizard

WLST Example

```
[oracle@edvmr1p0 /]$ 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('weblogic','password','t3://localhost:7001')
```

```
Connecting to t3://localhost:7001 with userid system ...
```

```
Successfully connected to Admin Server 'myAdmin' that belongs to domain 'mydomain'.
```

```
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:/mydomain/serverConfig> cd('Servers')
```

```
wls:/mydomain/serverConfig/Servers> ls()
```

```
dr--    myAdmin
```

```
dr--    myserver1
```

```
dr--    myserver2
```

```
wls:/mydomain/serverConfig/Servers> cd('myserver1')
```

```
wls:/mydomain/serverConfig/Servers/myserver1> get('StartupMode')
```

```
'RUNNING'
```

```
wls:/mydomain/serverConfig/Servers/myserver1> exit()
```

```
Exiting WebLogic Scripting Tool.
```

```
[oracle@edvmr1p0 /]$
```

WLST Command Requirements

- Use case-sensitive names and arguments of commands.
- Use arguments enclosed in single or double quotation marks.
- Precede the quoted string with **r** while specifying the backward slash (\) in the string.
 - Example: `readTemplate(r'c:\mytemplate.jar')`
- Do not use the following invalid characters in object names when you use WLST offline:
 - Period (.)
 - Slash (/)
 - Backward slash (\)
- Use the display help:
 - Example: `wls:/mydomain/serverConfig> help('disconnect')`

Running WLST Scripts

- Use the `setWLSEnv` script to initialize the `PATH` and `CLASSPATH` required for WLST.
 - If no script file is supplied, WLST runs in interactive mode.
- Use the `execfile()` command to run additional scripts.

```
>. ./setWLSEnv.sh  
>java weblogic.WLST [scriptfile.py]
```

To support SSL connection to a server:

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

Importing WLST as a Jython Module

- Invoke WLST:

```
[OS prompt] java weblogic.WLST  
wls:/offline>
```

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

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

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

```
[OS prompt] java org.python.util.jython
```

- Import the WLST module into your Jython module by using the Jython import command:

```
>>>import wl
```

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

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

General WLST Commands

Most of the commands need parameters in parentheses, even if null. The `help()` command shows this; the others are understood to be the same way.

Command	Description
<code>help()</code>	Get help for a given WLST command.
<code>exit</code>	Quit WLST.
<code>dumpVariables</code>	Display all variables used by WLST.
<code>dumpStack</code>	Display the stack trace for the last error that occurred in WLST.
<code>redirect / stopRedirect</code>	Redirect all WLST output to a file.

Offline WLST Commands

Command	Description
<code>createDomain</code>	Create a domain by using a given template.
<code>readDomain</code>	Open an existing domain on the file system.
<code>readTemplate</code>	Open an existing domain template.
<code>addTemplate</code>	Apply a template file to the current domain.
<code>updateDomain</code>	Save changes to the current domain.
<code>writeDomain</code>	Save changes to the current domain to a specified directory.
<code>writeTemplate</code>	Save the current domain to a template file.
<code>assign / unassign</code>	Target applications or services to servers.
<code>setOption</code>	Configure domain creation options (domain name, Java home, start mode, and so on).

Creating a Domain: Example



```
readTemplate('mybasetemplate.jar')
```



```
setOption('DomainName','mydomain')
```


```
setOption('JavaHome','/home/myjdk')
```

```
setOption('ServerStartMode','prod')
```



```
writeDomain('/home/mydomains')
```

```
closeTemplate()
```



```
readDomain('/home/mydomains/mydomain')
```

```
addTemplate('myjms.jar')
```

```
addTemplate('myapps.jar')
```

```
updateDomain()
```

```
closeDomain()
```

```
exit()
```

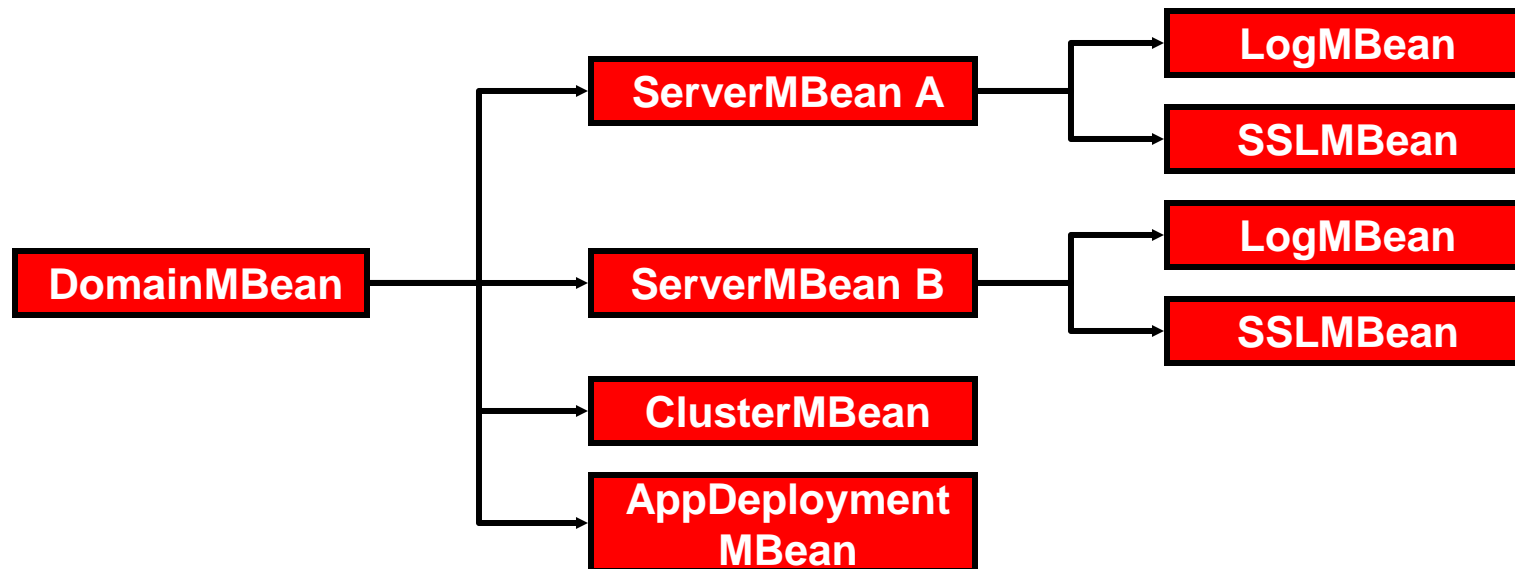
Online WLST Commands

Command	Description
<code>connect</code>	Connect to a server by using supplied credentials.
<code>disconnect</code>	Disconnect from the current server.
<code>shutdown</code>	Shut down servers.
<code>start</code>	Use the Node Manager to start servers.
<code>startEdit</code>	Begin a new change session.
<code>stopEdit</code>	Release the edit lock and discard any changes.
<code>activate</code>	Commit all changes in the current session.
<code>showChanges</code>	List all changes made in the current session.
<code>isRestartRequired</code>	Determine if any changes require a server restart.
<code>deploy/redeploy</code>	Deploy an application to servers.
<code>undeploy</code>	Shut down a running application on servers.

WebLogic JMX: Overview

JMX MBeans:

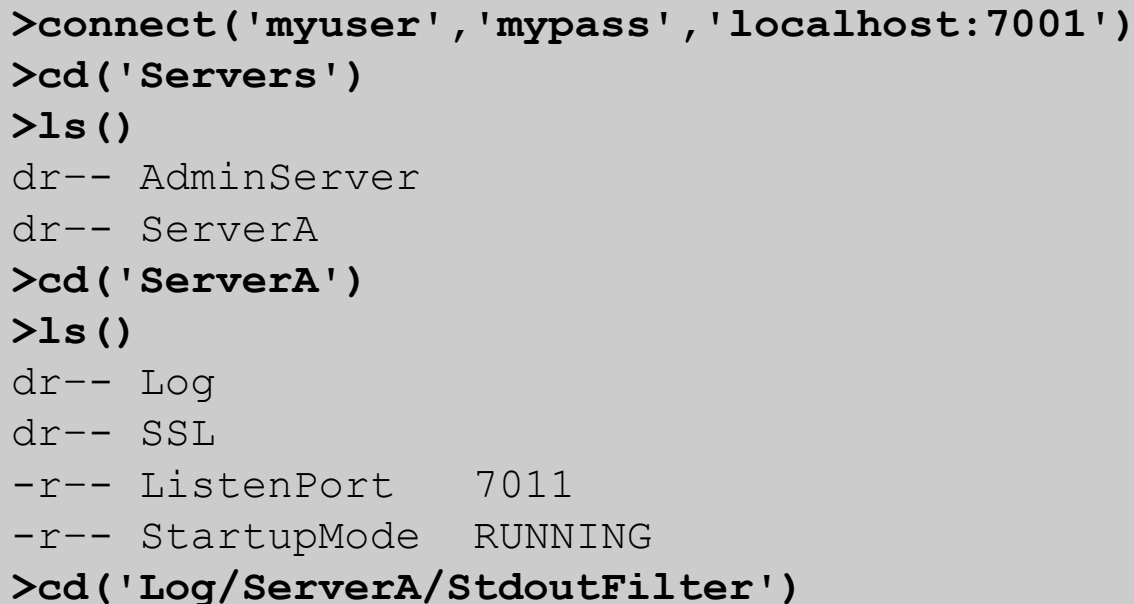
- Are hierarchical Java objects found on the server
- Have attributes and operations
- Support the configuration, management, and monitoring of all types of server resources



Navigating JMX MBeans

- Use the `cd`, `ls`, and `pwd` commands to navigate the server configuration or run-time MBeans, similar to a file system.
- Use the `get` or `set` commands to read or update the MBean attributes.

The `cmo` variable refers to the current MBean.



```
>connect('myuser','mypass','localhost:7001')
>cd('Servers')
>ls()
dr-- AdminServer
dr-- ServerA
>cd('ServerA')
>ls()
dr-- Log
dr-- SSL
-r-- ListenPort      7011
-r-- StartupMode     RUNNING
>cd('Log/ServerA/StdoutFilter')
```

A terminal window with a light gray background and a black border. On the left side of the terminal, there are three red triangle icons pointing to the right, aligned with the first, fourth, and seventh lines of the command sequence. The commands are entered in a monospaced font, and the output is displayed on subsequent lines.

Generating a WLST Script



ORACLE

Quiz

Which link would you click in the console to add or remove columns to or from a monitoring page?

1. Customize this table
2. Change Monitoring View
3. Update Columns
4. Manage Preferences

Quiz

WLST communicates with Oracle WebLogic Server's _____ to retrieve and update resources on a running server.

1. Templates
2. Logs
3. MBeans
4. Scripts

Quiz

The _____ panel in the Administration Console uses a tree to represent your domain resources.

1. Preferences
2. Domain Structure
3. How do I?
4. Change Center

Quiz

The Administration Console is unavailable if the administration server is shut down.

1. True
2. False

Quiz

Both the administration server and the managed servers can be started by using the Administration Console.

1. True
2. False

Quiz

Using WLST's _____ mode, you can supply commands one at a time and get immediate feedback.

1. Management
2. Operational
3. Sequential
4. Template
5. Interactive

Summary

In this lesson, you should have learned how to:

- Customize the console and tables
- Use the breadcrumbs to repeat recent tasks
- Operate WLST
- Describe how change management is performed internally using JMX and MBeans

Practice 6 Overview:

Using the Administrative Console and WLST

This practice covers the following topics:

- Invoking the Administration Console and navigating through common pages
- Modifying properties of WebLogic Server using the Administration Console
- Configuring and viewing log file entries for a WebLogic Server using the Administration Console.
- Performing tasks with the WLST
- Shutting down a server or an entire domain using WLST