



*WebSphere Application
Server V8.5.5 Administration*

(Course code WA855 / VA855)

Student Exercises

ERC 1.0



WebSphere Education

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

AFS®	AIX®	CICS®
CloudBurst®	DataPower®	DB™
DB2®	developerWorks®	Express®
HACMP™	IMS™	Lotus®
MVS™	Notes®	OS/400®
Passport Advantage®	Power®	RACF®
Rational®	RDN®	Redbooks®
Tivoli®	U®	WebSphere®
z/OS®	zSeries®	

Adobe is either a registered trademark or a trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Intel and Intel Core are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows and Windows Vista are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

VMware and the VMware “boxes” logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the “Marks”) of VMware, Inc. in the United States and/or other jurisdictions.

Other product and service names might be trademarks of IBM or other companies.

July 2013 edition

The information contained in this document has not been submitted to any formal IBM test and is distributed on an “as is” basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer’s ability to evaluate and integrate them into the customer’s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Contents

Trademarks	vii
Exercises description	xi
Verifying the image and course materials	xiii
Linux licensing agreement	xv
Exercise 1. Installing IBM Installation Manager	1-1
Section 1: Resetting the WebSphere environment	1-2
Section 2: Logging in	1-2
Section 3: Installing IBM Installation Manager	1-2
Section 4: Confirming installation of IBM Installation Manager	1-7
Exercise 2. Installing WebSphere Application Server	2-1
Section 1: Resetting the WebSphere environment	2-3
Section 2: Install WebSphere Application Server	2-3
Section 3: Modify WebSphere Application Server	2-13
Section 4: Create a profile with the Profile Management Tool	2-17
Section 5: Verify installation of WebSphere Application Server	2-30
Section 6: Create a backup of profile1	2-39
Section 7: Start and stop the WebSphere Application Server	2-41
Section 8: Explore the directory structure of WebSphere Application Server	2-43
Section 9: Check the installation log files	2-44
Section 10: Check the application server log files	2-45
Section 11: Correct the website security certificate error (optional)	2-49
Exercise 3. Installing IBM HTTP Server	3-1
Section 1: Resetting the WebSphere environment	3-3
Section 2: Installation of the IBM HTTP Server	3-3
Section 3: Install IBM HTTP Server plug-in	3-11
Section 4: Install WebSphere Customization Toolbox	3-12
Section 5: Configuring the web server plug-in	3-15
Section 6: Explore the IBM HTTP Server and plug-in installation	3-27
Section 7: Check installation log files	3-28
Section 8: Review configuration and error files	3-28
Section 9: Adding a web server as an IBM Installation Manager repository (optional)	3-29
Exercise 4. Exploring the administrative console	4-1
Section 1: Resetting the WebSphere environment	4-2
Section 2: Start the server	4-2
Section 3: Start the administrative console	4-3
Section 4: Explore the navigation tree	4-8

Section 5: Explore guided activities	4-11
Section 6: Explore server settings	4-13
Section 7: Examine application settings	4-21
Section 8: Examine environment settings	4-23
Section 9: Examine resource settings	4-25
Section 10: Examine troubleshooting	4-26
Section 11: Modify the administrative console session timeout	4-27
Section 12: Log out of the administrative console	4-31
Section 13: Explore configuration files	4-32
Exercise 5. Assembling an application5-1
Section 1: Resetting the WebSphere environment	5-4
Section 2: Start the IBM Assembly and Deploy Tools for WebSphere Administration	5-5
Section 3: Create an enterprise application project	5-6
Section 4: Add the Plants By WebSphere utility module	5-13
Section 5: Add the web module	5-16
Section 6: Add a test server	5-20
Section 7: Configure WebSphere data sources	5-24
Section 8: Export the enterprise archive (EAR) file	5-31
Exercise 6. Installing an application6-1
Section 1: Resetting the WebSphere environment	6-2
Section 2: Start the server and the administrative console	6-2
Section 3: Create J2C authentication aliases	6-3
Section 4: Create a JDBC provider and data sources for the application	6-6
Section 5: Install the PlantsByWebSphere enterprise application	6-14
Section 6: Test the enterprise application	6-20
Section 7: Use a monitored directory to deploy an enterprise application	6-26
Exercise 7. Problem determination7-1
Section 1: Resetting the WebSphere environment	7-3
Section 2: Working with log files of the application server	7-3
Section 3: Set up and configure HPEL	7-9
Section 4: Use the Log Viewer in the administrative console to examine log data and trace data	7-16
Section 5: Enable tracing for an application server and view trace data from the Log Viewer	7-19
Section 6: Enable cross-component trace (XCT)	7-23
Section 7: Collecting JVM data	7-28
Section 8: Clean up server1	7-37
Section 9: READ ONLY: Using IBM Support Assistant tools to analyze JVM data	7-38
Exercise 8. Using wsadmin8-1
Section 1: Resetting the WebSphere environment	8-3
Section 2: wsadmin command-line arguments	8-3
Section 3: Configuring the wsadmin environment	8-10

Section 4: Work with wsadmin administrative objects	8-13
Section 5: Creating a wsadmin script	8-16
Section 6: Using console command assist	8-25
Section 7: Exploring the scripting libraries	8-29
Section 8: Using properties file-based configuration	8-34
Section 9: Using SWING with Jython (optional)	8-38
Exercise 9. Creating a federated cell	9-1
Section 1: Resetting the WebSphere environment	9-2
Section 2: Use the Profile Management Tool to create a deployment manager profile	9-3
Section 3: Back up the Dmgr profile configuration	9-15
Section 4: Federate profile1 into the cell of the deployment manager	9-17
Section 5: Create a custom profile and federate it into the deployment manager cell	9-23
Section 6: Add the IBM HTTP Server to the cell	9-29
Section 7: Add the web server to the configuration	9-31
Section 8: Mapping modules to servers	9-36
Section 9: Working with the plug-in configuration file	9-38
Section 10: Test the plug-in configuration	9-41
Exercise 10. Clustering and workload management	10-1
Section 1: Resetting the WebSphere environment	10-2
Section 2: Check nodes and node agents	10-2
Section 3: Creating the PlantsCluster cluster	10-3
Section 4: Set the applications to run on the cluster	10-10
Section 5: Create a cluster scoped JDBC resource	10-11
Section 6: Test the application	10-18
Section 7: Configure session replication settings	10-25
Section 8: Test the application for session failover	10-29
Exercise 11. Configuring the service integration bus	11-1
Section 1: Resetting the WebSphere environment	11-4
Section 2: Making the configuration decisions	11-4
Section 3: Examine the scripting environment	11-8
Section 4: Setting up the service integration bus	11-14
Section 5: Configuring the JMS resources	11-19
Section 6: Installing the messaging applications	11-22
Section 7: Examine the configured environment	11-24
Section 8: Applications that are used in this exercise	11-34
Section 9: Testing the applications and exploring messaging engine policies ..	11-35
Section 10: Read only: Configure the scalability messaging engine policy ..	11-50
Exercise 12. Configuring WebSphere security	12-1
Section 1: Resetting the WebSphere environment	12-3
Section 2: Verify administrative security	12-3
Section 3: Defining WebSphere administrative console users	12-3

Section 4: Authenticate to the WebSphere administrative console and test mapped users	12-9
Section 5: Enabling fine-grained control	12-10
Section 6: Test the fine-grained access	12-17
Section 7: Examine security domains	12-20
Exercise 13. Configuring application security	13-1
Section 1: Resetting the WebSphere environment	13-2
Section 2: Enabling application security	13-2
Section 3: Securing the PlantsByWebSphere application	13-4
Section 4: Exploring the details (optional)	13-17
Exercise 14. Configuring SSL for WebSphere	14-1
Section 1: Resetting the WebSphere environment	14-3
Section 2: Create a backup	14-3
Section 3: Create a profile	14-4
Section 4: Examine the node certificates	14-9
Section 5: Examine certificate expiration and updating	14-17
Section 6: Plug-in key ring propagation	14-21
Section 7: Configuring SSL for IBM HTTP Server (optional)	14-26
Section 8: Testing the SSL connection	14-30
Exercise 15. Working with the Liberty profile	15-1
Section 1: Resetting the WebSphere environment	15-3
Section 2: Start the IBM Assembly and Deploy Tools for WebSphere Administration and install the Liberty profile	15-3
Section 3: Explore the Liberty profile directory	15-13
Section 4: Use IBM Assembly and Deploy Tools to deploy an application	15-21
Section 5: Deploy an application through the dropins directory	15-25
Section 6: Deploy the blog application with a data source	15-27
Section 7: Configure SSL and application security	15-40
Section 8: Explore the use of shared folders and include statements	15-48
Section 9: Configure the server to generate a web server plug-in configuration file	15-56
Exercise 16. Using the performance monitoring tools	16-1
Section 1: Resetting the WebSphere environment	16-3
Section 2: Verifying the environment	16-3
Section 3: Enabling performance monitoring and setting user preferences	16-3
Section 4: Viewing servlet and web applications module data	16-6
Section 5: Using the Tivoli Performance Viewer performance advisor	16-11
Section 6: Using request metrics	16-13
Appendix A. Resetting the WebSphere environment	A-1
Appendix B. List of Linux commands	B-1
Glossary of abbreviations and acronyms	X-1

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

AFS®	AIX®	CICS®
CloudBurst®	DataPower®	DB™
DB2®	developerWorks®	Express®
HACMP™	IMS™	Lotus®
MVS™	Notes®	OS/400®
Passport Advantage®	Power®	RACF®
Rational®	RDN®	Redbooks®
Tivoli®	U®	WebSphere®
z/OS®	zSeries®	

Adobe is either a registered trademark or a trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Intel and Intel Core are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows and Windows Vista are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

VMware and the VMware “boxes” logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the “Marks”) of VMware, Inc. in the United States and/or other jurisdictions.

Other product and service names might be trademarks of IBM or other companies.

Exercises description

This course includes the following exercises:

- Exercise 1: Installing IBM Installation Manager
- Exercise 2: Installing WebSphere Application Server
- Exercise 3: Installing IBM HTTP Server
- Exercise 4: Exploring the administrative console
- Exercise 5: Assembling an application
- Exercise 6: Installing an application
- Exercise 7: Problem determination
- Exercise 8: Using wsadmin
- Exercise 9: Creating a federated cell
- Exercise 10: Clustering and workload management
- Exercise 11: Configuring the service integration bus
- Exercise 12: Configuring WebSphere security
- Exercise 13: Configuring application security
- Exercise 14: Configuring SSL for WebSphere
- Exercise 15: Working with the Liberty profile
- Exercise 16: Using the performance monitoring tools

In the exercise instructions, you can check off the line before each step as you complete it to track your progress.

Most exercises include required sections that should always be completed. It might be necessary to complete these sections before you can start later exercises. Some exercises also include optional sections that you might want to complete if you have sufficient time and want an extra challenge.

Verifying the image and course materials

The student books and VMware image for this course display a release number called the edition revision code (ERC). To verify that the books and image are at the same level, compare the ERC of the VMware image to the ERC of the student books.

- ___ 1. Determine the ERC number of your course materials.
 - ___ a. Open all of the books you received (either printed or PDF).
 - ___ b. Note the ERC listed on the front page of your books. The ERC number is listed under the course title on the first page of the books as in the following example:

*Administration of IBM
WebSphere Process Server
V7*

(Course code WB722 / VB722)

Student Exercises

ERC 1.0

- ___ 2. Determine the ERC number of the VMware image.
 - ___ a. On the image desktop, open the **readme.txt** file.
 - ___ b. Note the ERC listed in the file. The ERC number is indicated on the “ERC number” line as in the following example:

```
-----
-----
Student workstation virtual Machine Information
Websphere Education

Copyright (C) 2011 IBM Corporation
Course materials, including this virtual machine, may not be
reproduced in
whole or in part without prior written permission of IBM.
-----

Course code: WB722/VB722
Course title: Administration of IBM websphere
Process Server V7
Application Server
ERC number: 1.0
Last modified date: April 26, 2011

Operating system: Microsoft windows XP Professional SP3
Primary account username: Administrator
Primary account password: web1sphere
```

- ___ 3. Verify the image and course materials.
 - ___ a. If the ERC number on the image does not match the ERC number on the printed materials, notify your instructor that the materials are not synchronized.



Stop

Do NOT proceed with the exercises if the ERC numbers on the course materials and course image do not match; ask your instructor for further direction.



Linux licensing agreement

Logging in



Important

If you are taking an ILO (instructor-led online) virtual course, you can skip this part. You are automatically logged in as: `root`

Remember to not log out of the machine.

- ___ 1. When you start your VMware image, you are prompted for a user ID and password.

```
Welcome to SUSE Linux Enterprise Server 11 (i586) - Kernel 2.6.27.45-0.1-vmi

susehost login: root
Password:
```

At this prompt, enter:

- User ID: `root`
- Password: `websphere`

Be sure to use lowercase. The login fields are case-sensitive.

- ___ 2. If you are already logged in, but not as `root`, then log off your current ID and log in as: `root`
- ___ a. Click **Computer > Logout**.
- ___ b. Select **Logout**.
- ___ c. In the login window, enter `root` as the user name and `websphere` as the password.

Accepting the license agreements in an ILT class

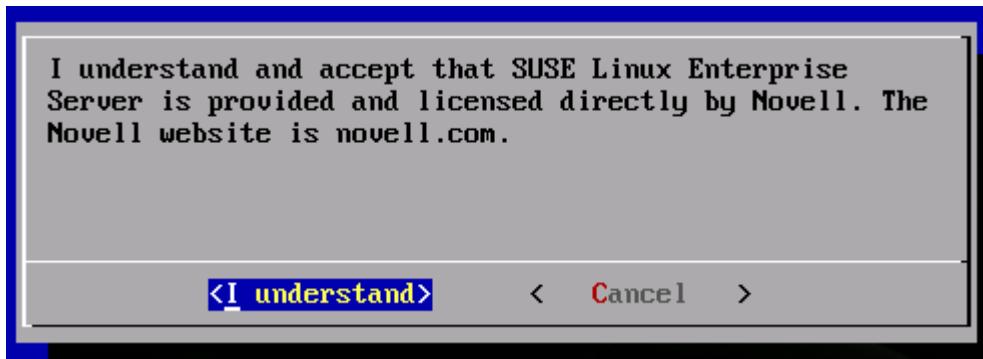
The first time your lab system is started, the system might go through some licensing agreements. In some cases (such as some online courses), the license agreements are already accepted on the students' behalf, so this section can be ignored.

As soon as the license agreements are accepted, you can log in to the operating system through the normal graphical interface.

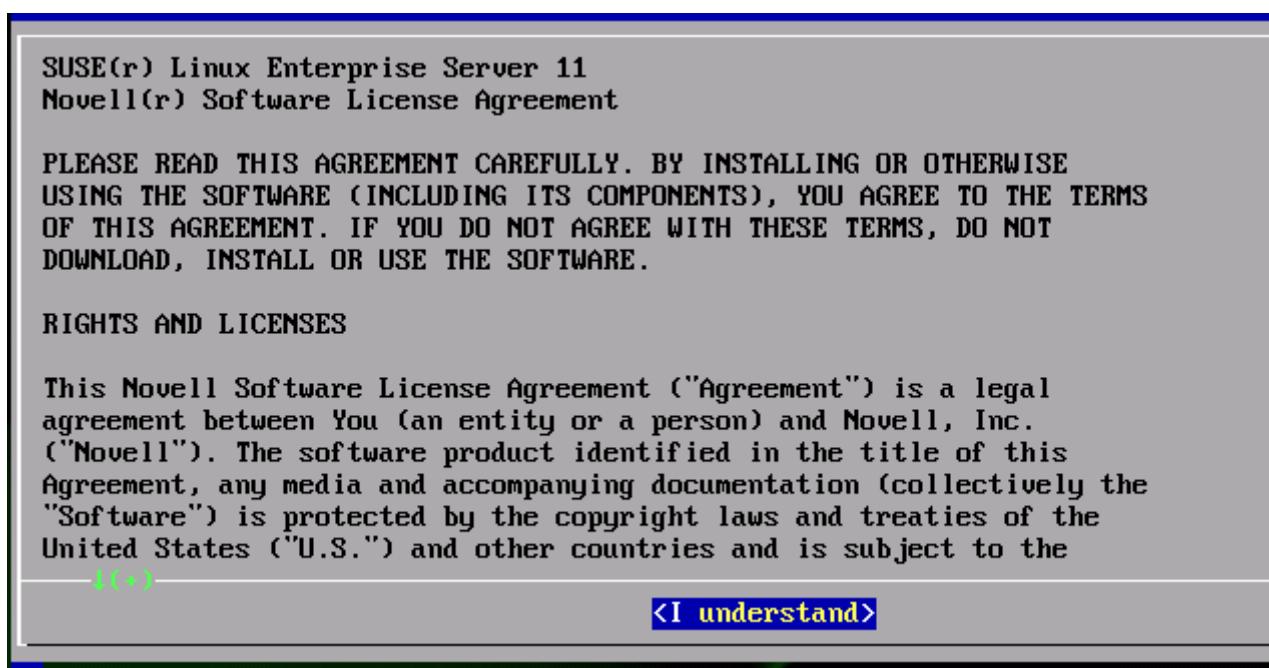
**Important**

If you are taking an ILO (instructor-led online) virtual course, you are not prompted to accept the license agreements. This part of the exercise is completed for you.

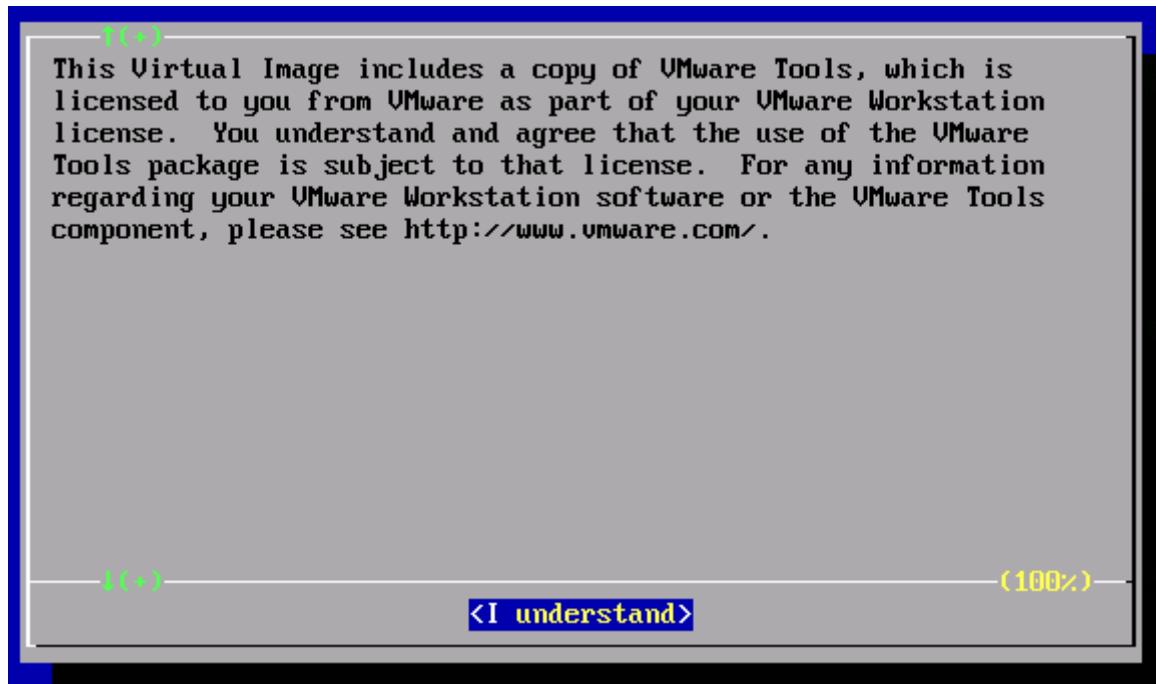
- ___ 1. Navigate through the License wizard and accept the terms on each screen (note: depending on the version of your system, the license screens might look slightly different).
 - ___ a. The licensing wizard starts. Press the Enter key to accept **I understand** for the Linux Distribution Statement (explains that SUSE Linux Enterprise Server is provided and licensed directly by Novell).



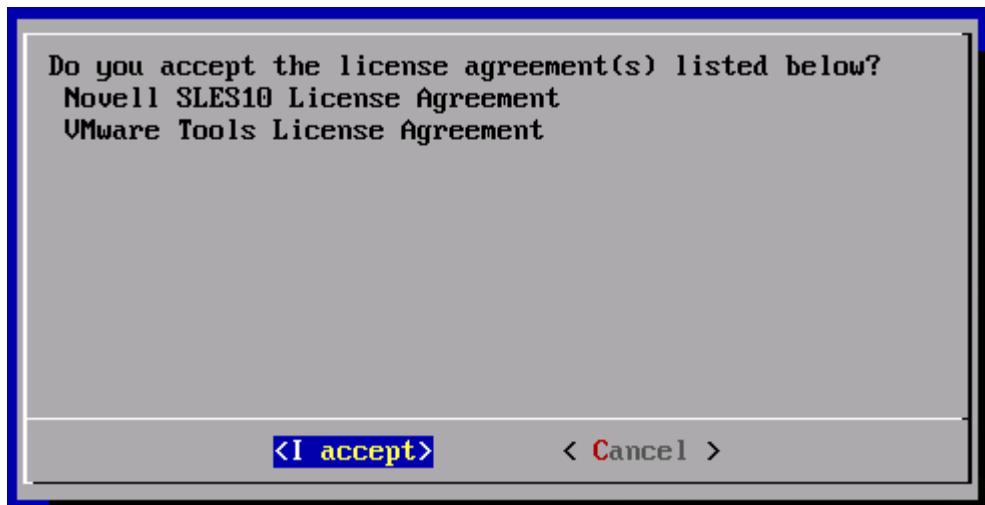
- ___ b. Press the Enter key to accept the **I understand** for the SUSE Linux Enterprise Server 11 Software License Agreement.



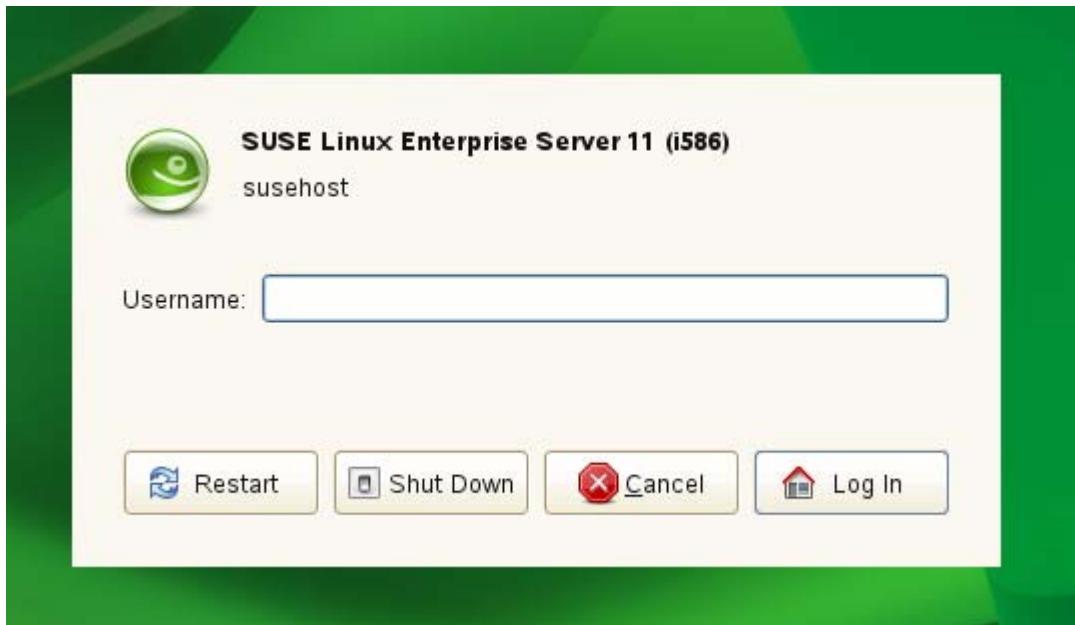
- ___ c. Press the Enter key to accept the **I understand** for the VMware Tools License Agreement. (The agreement explains that the image includes a copy of VMware Tools, which is licensed to you from VMware as part of your VMware Workstations license.)



- ___ d. Press the Enter key to select the **I accept** for the License Agreement.



- ___ 2. The license wizard is completed and is not required to run again. The system now prompts you for a normal graphical login.



- ___ a. Enter the following when prompted:

- Username: root
- Password: web1sphere

Exercise 1. Installing IBM Installation Manager

What this exercise is about

This exercise covers the installation of IBM Installation Manager.

What you should be able to do

At the end of this exercise, you should be able to:

- Install IBM Installation Manager
- View the installation log file
- Confirm the installation of Installation Manager

Introduction

IBM Installation Manager is an installation management tool that installs and maintains Installation Manager-based software packages. It is the Eclipse-based tool that provides you the ability to install and modify packages, search for updates, uninstall, and roll back. Installation Manager makes it easier for you to download and install code for a number of IBM software packages.

IBM Installation Manager comes in the form of an installation kit, which contains a set of Installation Manager installation files and a flat-file repository for the product. The installation kit is used only for the setup and maintenance of Installation Manager. It is unnecessary to run Installation Manager except on those systems on which you install or update product code. You normally need only one Installation Manager on a computer because one Installation Manager can track any number of product installations.

Requirements

To complete this exercise, you need the IBM Installation Manager installation binary file.

Exercise instructions

Section 1: Resetting the WebSphere environment



Note

To reset your WebSphere environment, read **Appendix A** for instructions on how to complete this procedure.

Section 2: Logging in

- ___ 1. When you start your computer, you are prompted for a user ID and password. At this prompt, enter:
 - User ID: `root`
 - Password: `web1sphere`

If you are already logged in, but not as `root`, then log off your current ID and log in as: `root`

Section 3: Installing IBM Installation Manager

Before you install IBM Installation Manager, you must first decide in which mode to run Installation Manager. The mode determines which user or user group can complete the installation. The choices are administrator, non-administrator, or group. Second, you must decide where the product files and runtime data are going to be located.



Information

You can find more details about the IBM Installation Manager in the information center at:

<http://pic.dhe.ibm.com/infocenter/install/v1r6/index.jsp>

- ___ 1. Locate the repository and package.
 - ___ a. Open a terminal window. Click **Computer > Gnome Terminal** or click the **Gnome Terminal** icon in the bottom center of the image.



- ___ b. Navigate to the following directory:

`/usr/IBM-repositories/IIM162/`

**Note**

A **repository** is where the installable packages are found. The repository includes metadata that describes the software version and how it is installed. It has a list of files that are organized in a tree structure. The repository can be local or on a remote server.

A **package** is a software product that Installation Manager installs. It is a separately installable unit that can operate independently from other packages of that software.

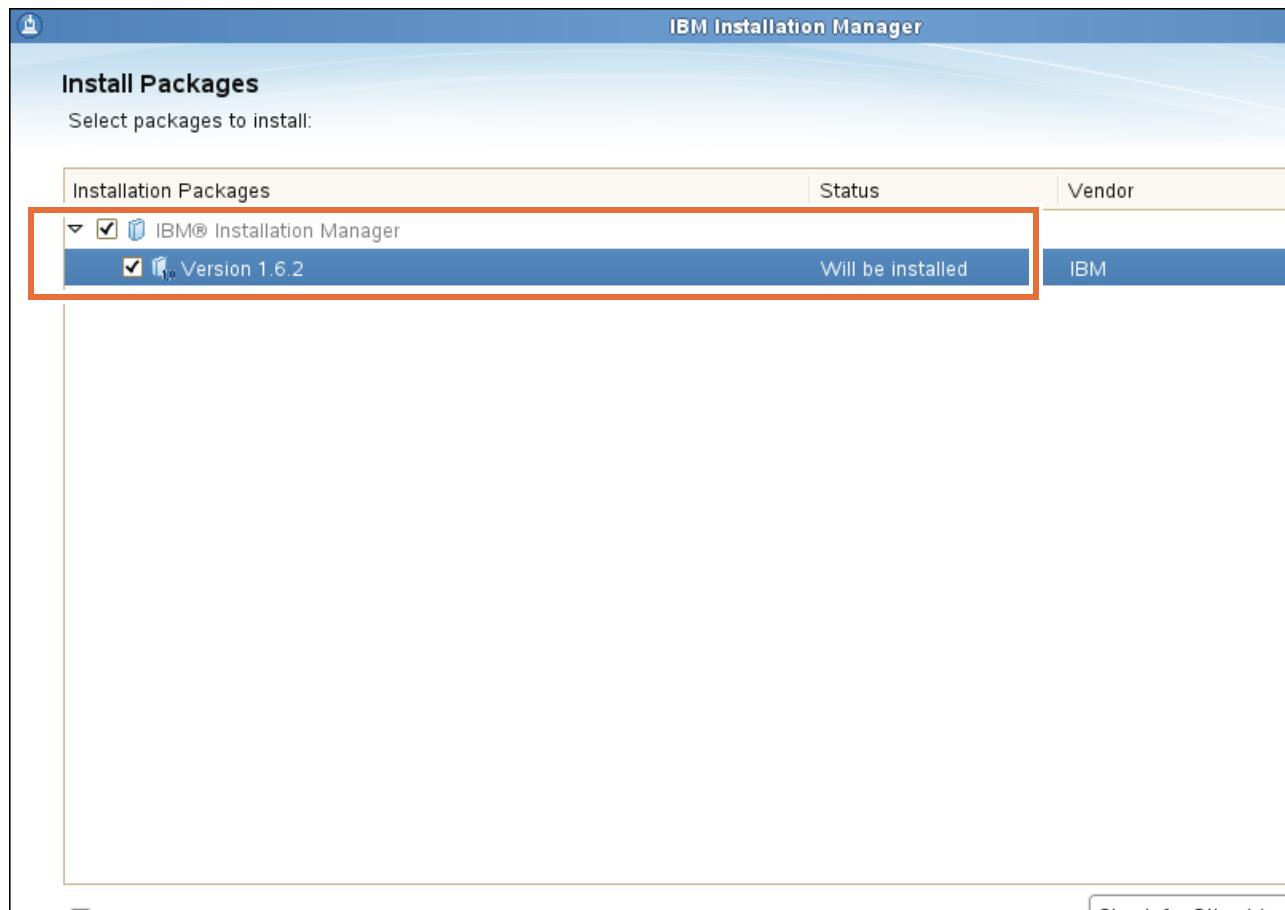
A **package group** represents a directory that contains resources that packages share with other packages in the same group. Some packages support installing to the same package group, and others must be installed to a new package group.

- ___ c. Enter the following command to start the installation:

```
./install
```

```
Terminal
File Edit View Terminal Tabs Help
was85host:~ # cd /usr/IBM-repositories/IIM162/
was85host:/usr/IBM-repositories/IIM162 #
was85host:/usr/IBM-repositories/IIM162 # ./install
```

- ___ d. Since Installation Manager is not installed yet, the Installation Manager package is selected by default. The package is ready to install, as the status message indicates.



- ___ e. Click **Next**.

- ___ f. Select **I accept the terms in the license agreement**. Click **Next**.

I accept the terms in the license agreement

[Print All...](#)

I do not accept the terms in the license agreement

< Back **Next >** Install Cancel

- g. Accept /opt/IBM/InstallationManager/eclipse as Installation Manager Directory.

Install Packages

Select a location for Installation Manager.

Once installed, IBM Installation Manager will be used to install, update, modify, manage and uninstall your packages.

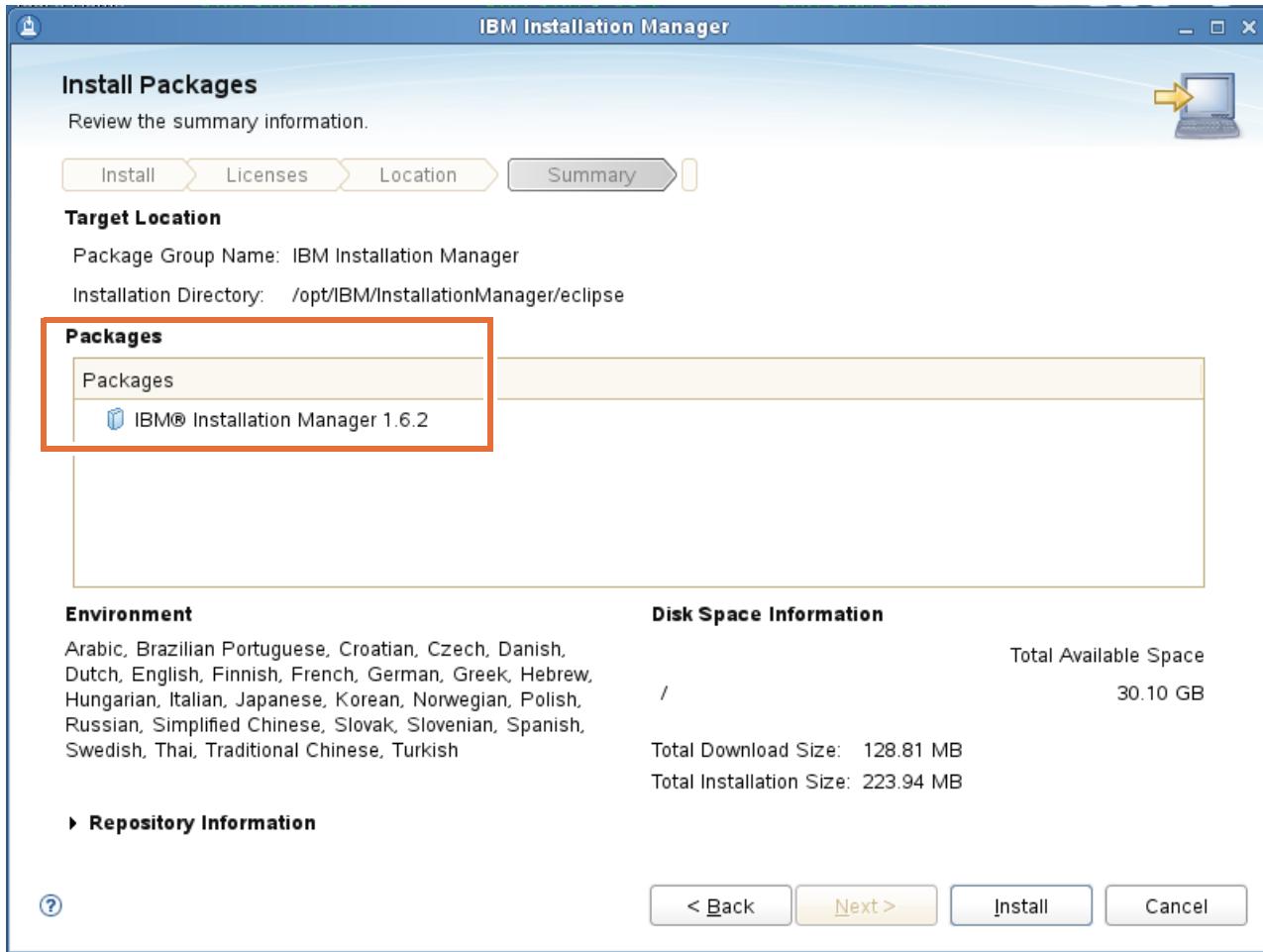
Installation Manager Directory: [Browse...](#)

Disk Space Information

Volume	Available Space
/	20.85 GB

- h. Click **Next**.

- __ i. On the summary page, review the packages. Only one package is listed, IBM Installation Manager 1.6.2. Click **Install**.



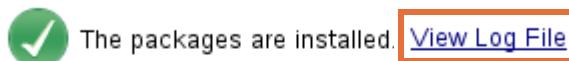
- __ j. At the bottom of the panel, observe the progress of the installation.
__ k. A message indicates that the packages are installed.



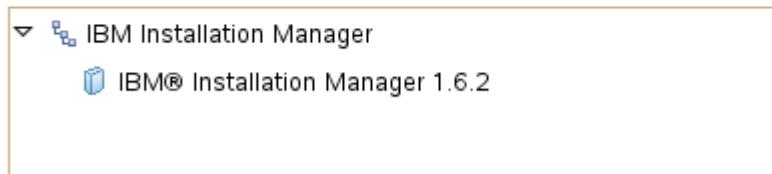
Section 4: Confirming installation of IBM Installation Manager

During installation, Installation Manager creates a session installation log file. This file is useful to either confirm a successful installation or to check for installation problems.

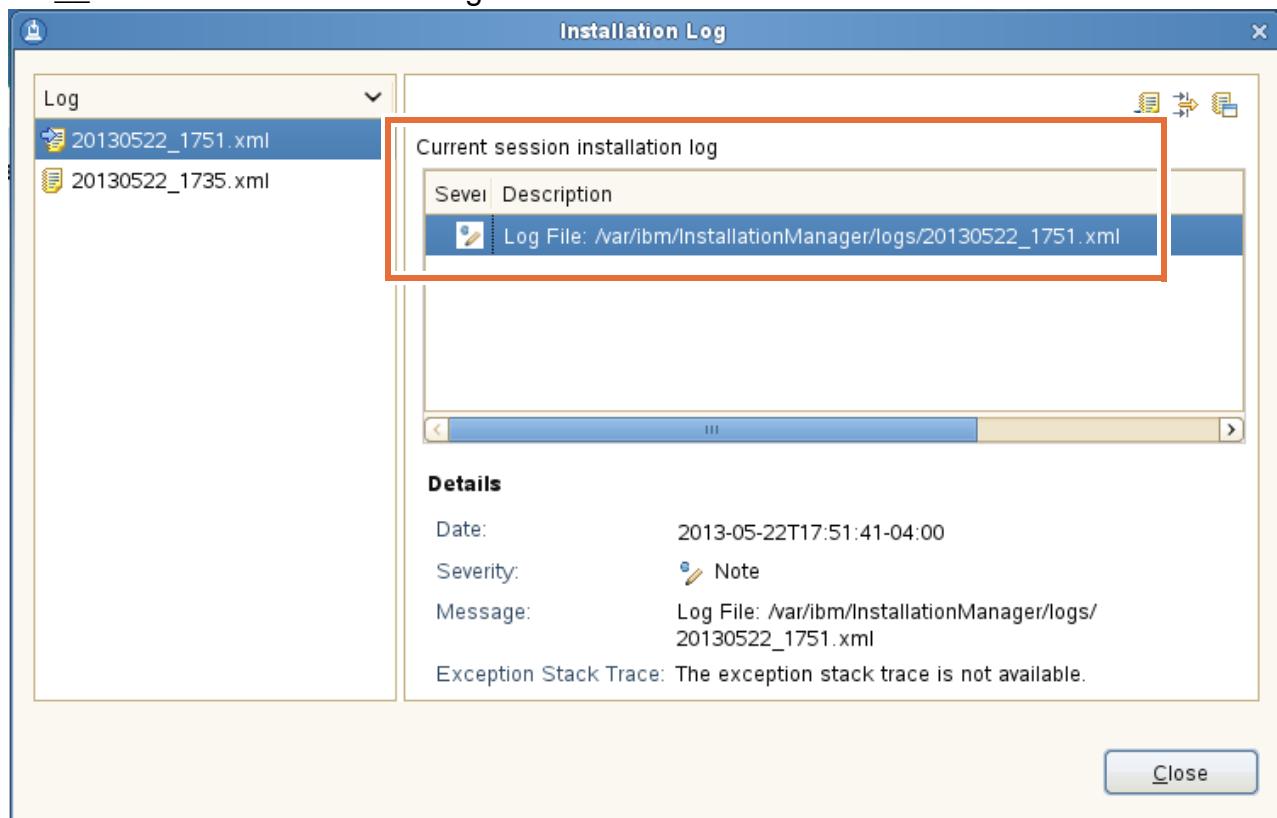
- __ 1. View the installation log file.
- __ a. Click **View Log File**.



The following package was installed:



- __ b. The current session log file is listed.



- __ c. Click the **Open log file** icon.



- d. The Install Log page opens in a browser window. Scan the page for any obvious error messages. Messages in the file indicate "complete", "install", "post-install", and other installation phases. These messages are indicators that the installation proceeded normally without errors.

Install Log			
49	INFO	09:26.71	Custom operation register Agent in Unix env in unit com.ibm.cic.agent.non.win32
50	INFO	09:28.73	Elapsed time 01:09.40 for: Performing "install" phase
51	INFO	09:28.73	Performing "post-install" phase
52	INFO	09:28.74	Elapsed time 00:00.00 for: Performing "post-install" phase
53	INFO	09:28.74	Performing "post-install configure" phase
54	INFO	09:29.73	Elapsed time 00:00.99 for: Performing "post-install configure" phase
55	INFO	09:29.75	Performing "complete" phase
56	INFO	09:30.11	Elapsed time 00:00.36 for: Performing "complete" phase
57	INFO	09:30.11	Performing "initialize" phase
58	INFO	09:30.11	Elapsed time 00:00.00 for: Performing "initialize" phase
59	INFO	09:30.12	Performing "pre-install configure" phase
60	INFO	09:30.12	Elapsed time 00:00.00 for: Performing "pre-install configure" phase
61	INFO	09:30.12	Performing "pre-install" phase
62	INFO	09:30.12	Elapsed time 00:00.00 for: Performing "pre-install" phase
63	INFO	09:30.12	Performing "install" phase
64	INFO	09:30.15	Custom operation com.ibm.cic.agent.nativeFixup.FixupAgent in unit com.ibm.cic
65	INFO	09:30.19	Custom operation com.ibm.cic.agent.nativePlatformFixup.FixupAgentPlatform in
66	INFO	09:30.21	Elapsed time 00:00.08 for: Performing "install" phase
67	INFO	09:30.21	Performing "post-install" phase
68	INFO	09:30.21	Elapsed time 00:00.00 for: Performing "post-install" phase
69	INFO	09:30.21	Performing "post-install configure" phase
70	INFO	09:30.25	Custom operation com.ibm.cic.agent.nativeAdminFixup.NativeAdminFixup in uni
71	INFO	09:30.26	Elapsed time 00:00.04 for: Performing "post-install configure" phase
72	INFO	09:30.26	Performing "complete" phase
73	INFO	09:30.26	Elapsed time 00:00.00 for: Performing "complete" phase
74	INFO	09:32.54	Elapsed time 01:48.42 for: Installing com.ibm.cic.agent 1.5.2000.20120223_090 selectors: ws=gtk, os=linux, arch=x86, nl=de,ru,ko,el,lt,en,it,pt_BR,fr,hu,es,zh,cs,ar

- e. Close the browser window.
 f. Close the Installation Log window.
 g. Close IBM Installation Manager.

End of exercise

Exercise review and wrap-up

In this exercise, you installed IBM Installation Manager. Installation Manager is the software that is used to install WebSphere Application Server and other tools.

Exercise 2. Installing WebSphere Application Server

What this exercise is about

In this exercise, you install WebSphere Application Server Network Deployment V8.5.5. In version 8.5.5, the installation of WebSphere Application Server Network Deployment is a two-step process. The first step uses IBM Installation Manager to install a set of shared product binary files. The second step uses the Profile Management Tool, part of the WebSphere Customization Toolbox, to create an application server profile.

After installation, you test the product to ensure that WebSphere Application Server was installed successfully. You also examine several log files to verify installation.

What you should be able to do

At the end of the exercise, you should be able to:

- Use IBM Installation Manager to install WebSphere Application Server Network Deployment
- Use IBM Installation Manager to modify a product feature
- Use the Profile Management Tool to create a profile
- Verify that the installation was successful by examining log files
- Start and stop the application server

Introduction

In this exercise, you install WebSphere Application Server Network Deployment V8.5.5. WebSphere Application Server relies on TCP/IP networking, so you must have TCP/IP correctly configured, and it is important that the host name of the computer remains unchanged.

The lab computer must be configured appropriately. Make sure that you know the host name of the computer because you need it for this exercise.

Requirements

This exercise requires that the IBM Installation Manager is already installed.

Exercise instructions

Preface

To do this exercise, you must complete the Installing IBM Installation Manager exercise.

Section 1: Resetting the WebSphere environment



Note

To reset your WebSphere environment, read **Appendix A** for instructions on how to complete this procedure.

Section 2: Install WebSphere Application Server

In this part of the exercise, you use the IBM Installation Manager to install WebSphere Application Server. Before the WebSphere Application Server installation can be started, an installation repository that contains the WebSphere Application Server installation binary files must be defined.



Information

You can find more details about WebSphere Application Server V8.5 in the information center at:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v8r5/index.jsp>

— 1. Start IBM Installation Manager.

- a. Open a terminal window.
- b. Navigate to the following directory:

/opt/IBM/InstallationManager/eclipse/

- c. Use the following command to start the IBM Installation Manager:

./IBMMIM

```
Terminal
File Edit View Terminal Tabs Help
was85host:~ # cd /opt/IBM/InstallationManager/eclipse/
was85host:/opt/IBM/InstallationManager/eclipse # ./IBMMIM
```

- ___ d. Installation Manager starts and shows the available wizards. Installation Manager contains a number of wizards to help install and maintain various packages.

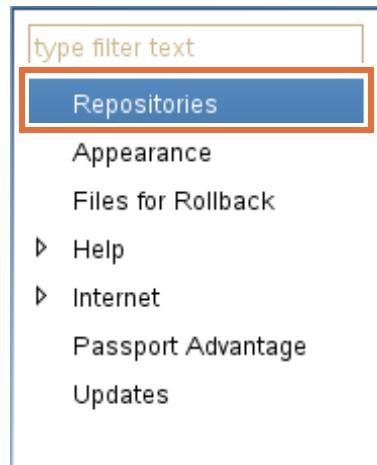


Information

Installation Manager uses repositories to identify the packages or updates to install. A repository is a location that stores data for installing, modifying, rolling back, updating, or uninstalling packages. Each installed package has an embedded location for its default update repository. You can add, edit, or remove repositories for use by Installation Manager.

By default, Installation Manager is configured to use a service repository that is made up of repositories at ibm.com. In this case, Internet access is required. If a computer does not have Internet access, Installation Manager can be configured to look for a local repository. Updates can be downloaded and placed in a temporary directory on the computer. Installation Manager looks in this directory for installable updates. You must manually configure local repositories.

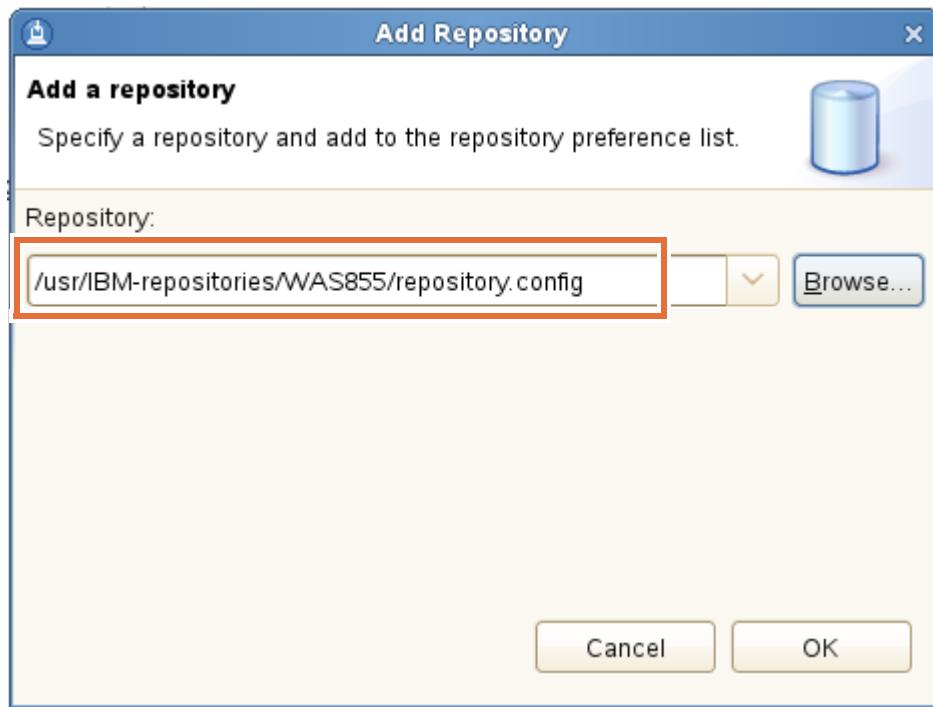
2. The Install Packages wizard takes you through the installation process for one or more packages. To locate the packages, you must configure the repository.
- a. Click **File > Preferences**.
 - b. Select **Repositories**.



- c. Click **Add Repository**.



- __ d. Browse to /usr/IBM-repositories/WAS855/repository.config and click Open.



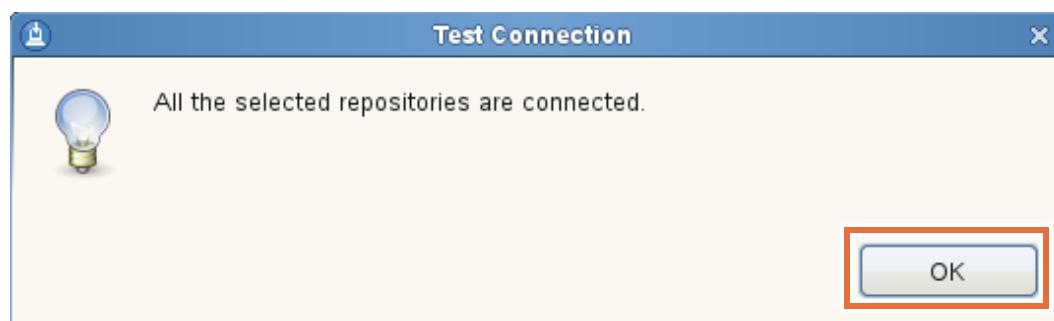
- __ e. Click OK.
__ f. The repository is added to the list of repositories. Click Apply.

A screenshot of a software interface titled "Repositories". The main area shows a table with one row. The first column is "Location" and contains the value "/usr/IBM-repositories/WAS855/repository.config", which is preceded by a checked checkbox. The second column is "Connection" and contains a small blue icon with a green and yellow bar. To the right of the table is a vertical toolbar with several buttons: "Add Repository...", "Edit Repository...", "Remove Repository", "Move Up", "Move Down", "Clear Credentials", and "Test Connections". The entire row in the table is highlighted with a red box.

- ___ 3. Test the connection to the local repository. This step is more important when you configure remote repositories and need access to the repository to complete a product installation.
- ___ a. Click **Test Connections**.



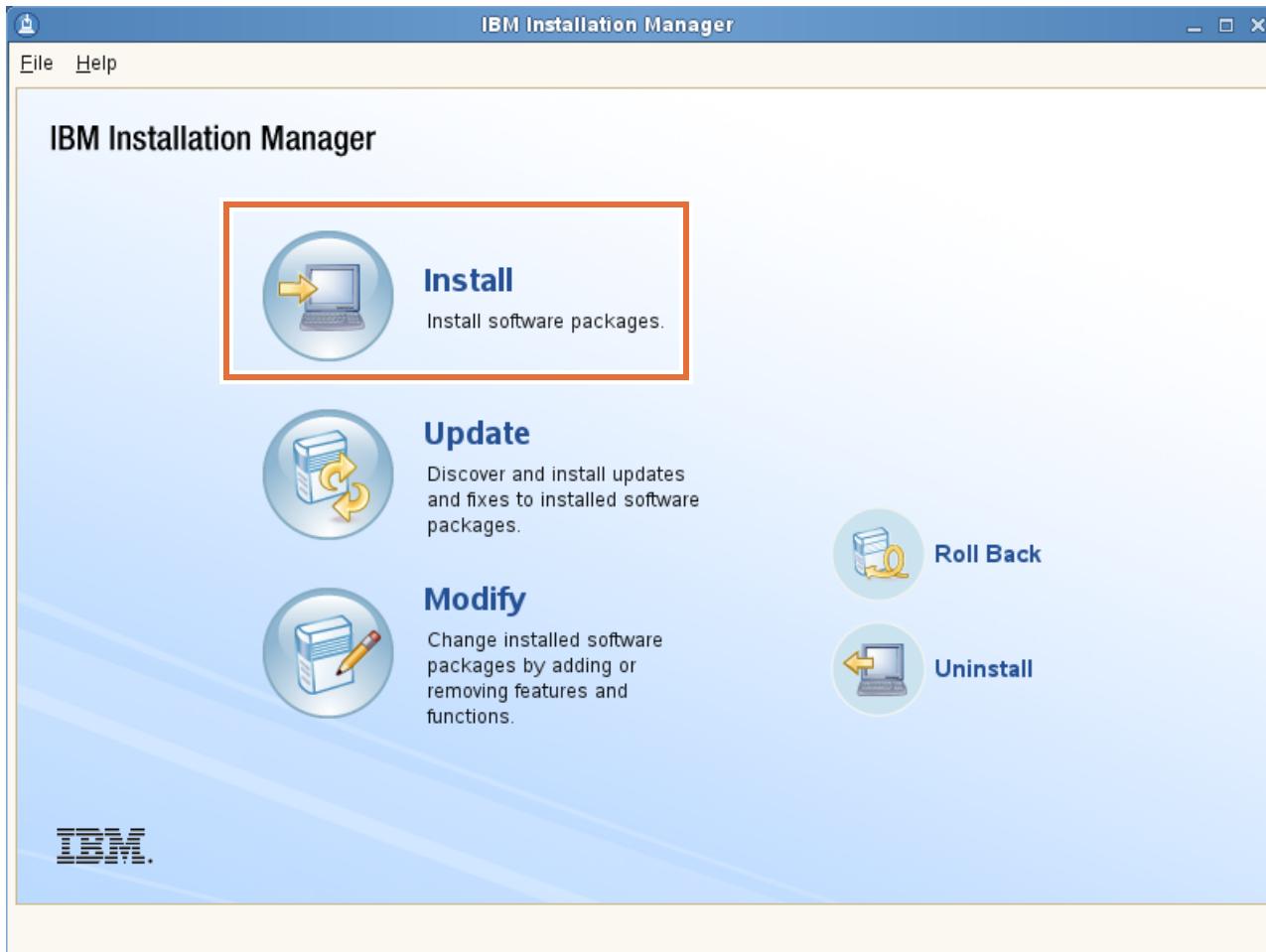
- ___ b. A message indicates that the repository is connected. Click **OK** to close the dialog box.



- ___ c. Click **OK** to close the preferences page.

__ 4. Install WebSphere Application Server.

- __ a. Click the **Install** icon.



- __ b. Installation Manager lists all the packages that it finds in the configured repositories. In this case, it searched the repository and found the following packages. Select **IBM WebSphere Application Server Network Deployment**.

The screenshot shows the 'Install Packages' panel. It has a title 'Install Packages' and a subtitle 'Select packages to install:'. Below this is a table with columns 'Installation Packages', 'Status', and 'Vendor'. A checkbox next to the package name is checked. The package listed is 'IBM WebSphere Application Server Network Deployr' (note the extra 'r') with version 'Version 8.5.5.0'. The status is 'Will be installed' and the vendor is 'IBM'. The entire row for this package is highlighted with a red box.

Installation Packages	Status	Vendor
<input checked="" type="checkbox"/> IBM WebSphere Application Server Network Deployr <input checked="" type="checkbox"/> Version 8.5.5.0	Will be installed	IBM

- __ c. The Install Packages panel is updated to indicate the status for Version 8.5.5.0 as *Will be installed*. Click **Next**.

- ___ d. The Licenses panel opens. You can read the license agreements for any of the packages. Select **I accept the terms in the license agreement** and click **Next**.

I accept the terms in the license agreement [Print All...](#)

I do not accept the terms in the license agreement

[**< Back**](#) Next > [Install](#) [Cancel](#)

- ___ e. On the Install Packages panel, accept the default location of /opt/IBM/IMShared for the Shared Resources Directory, and click **Next**.

Install Packages
Select a location for the shared resources directory.

Install Licenses **Location** Features Summary

When you install packages, files are stored in two locations:

1) The shared resources directory - resources that can be shared by multiple packages.
2) The installation directory - any resources that are unique to the package that you are installing.

! Important: You can only select the shared resources directory the first time you install a package using Installation Manager. For best results select the drive with the most available space because there will be less space for the shared resources of future packages.

Shared Resources Directory: **/opt/IBM/IMShared**

Disk Space Information

Volume	Available Space
/	20.66 GB

- ___ f. On the Location panel, the package for WebSphere Application Server Network Deployment V8.5 is selected along with **Create a new package group**. Verify that the Installation Directory is /opt/IBM/WebSphere/AppServer. The package group name is automatically modified to be unique. Click **Next**.

Install Packages

A package group is a location that contains one or more packages. Some compatible packages can be added into a common package group and will share a common user interface. Select an existing package group or a new one.

Install

Licenses

Location

Features

Summary

Use the existing package group

Create a new package group

Package Group Name

Installation Directory

IBM WebSphere Application Server V8.5

/opt/IBM/WebSphere/AppServer

Package Group Name: IBM WebSphere Application Server V8.5

Installation Directory: /opt/IBM/WebSphere/AppServer

Details

Shared Resources Directory: /opt/IBM/IMShared

Disk Space Information

Volume	Available Space
/	20.66 GB

- ___ g. The Features panel indicates the individual language packs for the WebSphere Application Server runtime environment and administrative console. Keep the default setting and click **Next**.

Install Packages

Select the translations to install.

Install

Licenses

Location

Features

Summary

Translations

English

Brazilian Portuguese

Czech

Hungarian

Italian

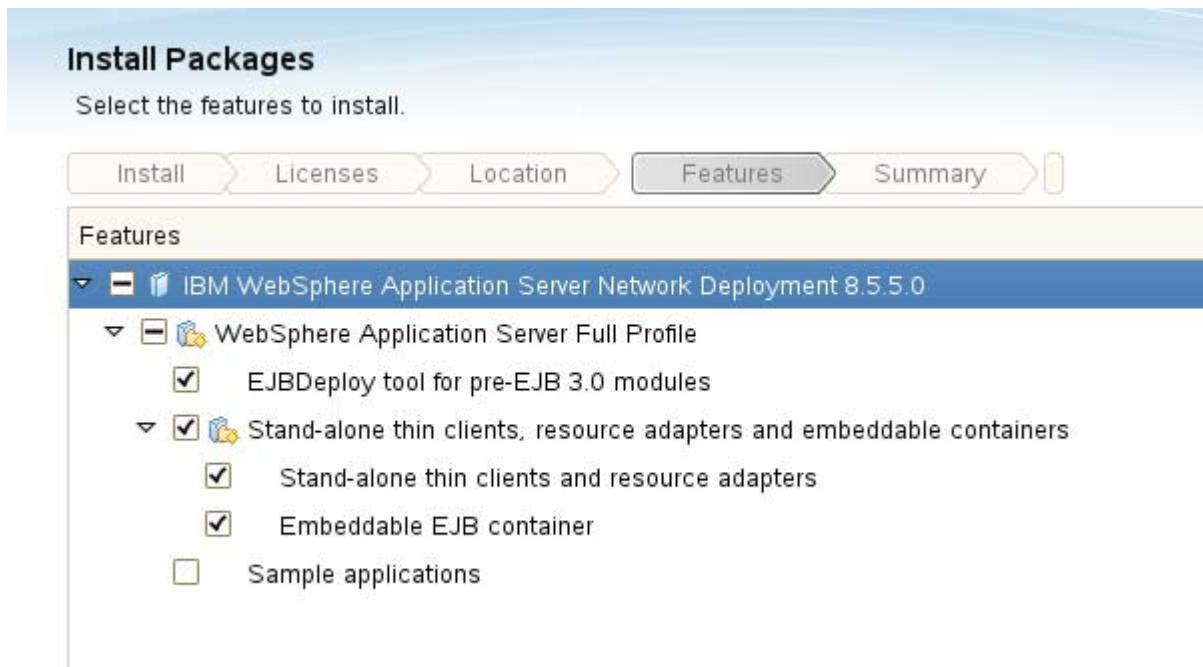
Japanese

Romanian

Russian

Simplified Chinese

- h. The Features panel lists more features for installation. Note the default selections and click **Next**.



Note

Do not select the Sample application feature. This feature is added to the installation later in this exercise.

- __ i. On the Summary panel, confirm your previous choices. Verify your information with the following figure. If necessary, you can click **Back** to modify any choices.

Install Packages

Review the summary information.

Install Licenses Location Features **Summary**

Target Location

Package Group Name: IBM WebSphere Application Server V8.5
 Installation Directory: /opt/IBM/WebSphere/AppServer
 Shared Resources Directory: /opt/IBM/IMShared

Packages

Packages

- ✓ IBM WebSphere Application Server Network Deployment 8.5.5.0
 - ✗ WebSphere Application Server Full Profile
 - ✗ EJBDeploy tool for pre-EJB 3.0 modules
 - ✗ Stand-alone thin clients, resource adapters and embeddable containers
 - ✗ Stand-alone thin clients and resource adapters
 - ✗ Embeddable EJB container

- __ j. Click **Install** to begin the installation. Depending on system resources, the installation takes several minutes. During the installation, verify the progress with the progress bar at the bottom of the window.
- __ k. When the installation completes, the installation results are displayed at the top of the panel. Select **None** in the area that indicates programs to start. Click **Finish**.

Install Packages

The packages are installed. [View Log File](#)

The following package was installed:

- ✗ IBM WebSphere Application Server V8.5
 - ✗ IBM WebSphere Application Server Network

Which program do you want to start?

- Profile Management Tool to create a profile.
- Profile Management Tool to create an application
- None**

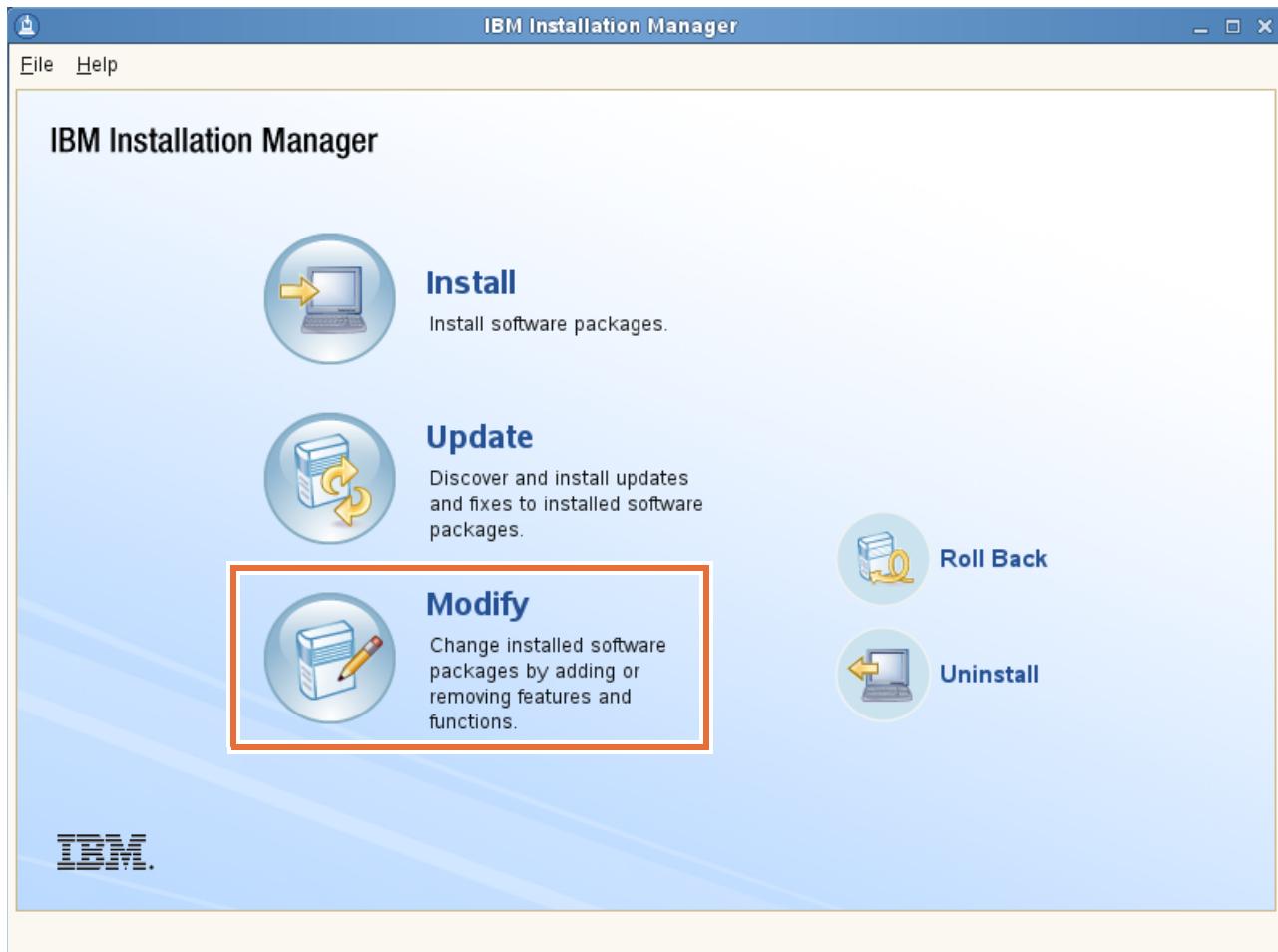
- __ I. Do not close Installation Manager as it is used in the next section of the exercise.

Section 3: Modify WebSphere Application Server

You can use Installation Manager to add or remove a product feature from an installed package. To modify an installed package, Installation Manager must have access to the repository that contains the package installation files.

In this part of the exercise, you modify the WebSphere Application Server installation from Section 1 to include the sample applications.

- __ 1. Modify the installation to include the sample applications.
__ a. From the main panel of Installation Manager, click **Modify**.



- ___ b. The Modify Packages panel is displayed. This panel lists the packages that are installed and can be modified. Select **IBM WebSphere Application Server Network Deployment V8.5**, which is installed in the /opt/IBM/WebSphere/AppServer directory. Click **Next**.

The screenshot shows the 'Modify Packages' interface. At the top, it says 'Select a package group to modify.' Below is a table with two columns: 'Package Group Name' and 'Directory'. A single row is visible, containing 'IBM WebSphere Application Server V8.5' in the first column and '/opt/IBM/WebSphere/AppServer' in the second column. The entire row is highlighted with a red box.

Package Group Name	Directory
IBM WebSphere Application Server V8.5	/opt/IBM/WebSphere/AppServer

Details

IBM WebSphere Application Server V8.5

- Shared Resources Directory: /opt/IBM/IMShared
- Installation Directory: /opt/IBM/WebSphere/AppServer
- Eclipse IDE: /opt/IBM/WebSphere/AppServer
- Translations: English

Installed Packages and Fixes

- IBM WebSphere Application Server Network Deployment 8.5.5.0

- ___ c. On the Features panel, you see a listing of individual language packs. Keep the default values and click **Next**.

- ___ d. The Features page lists the features for installation. Select **Sample applications** and click **Next**.

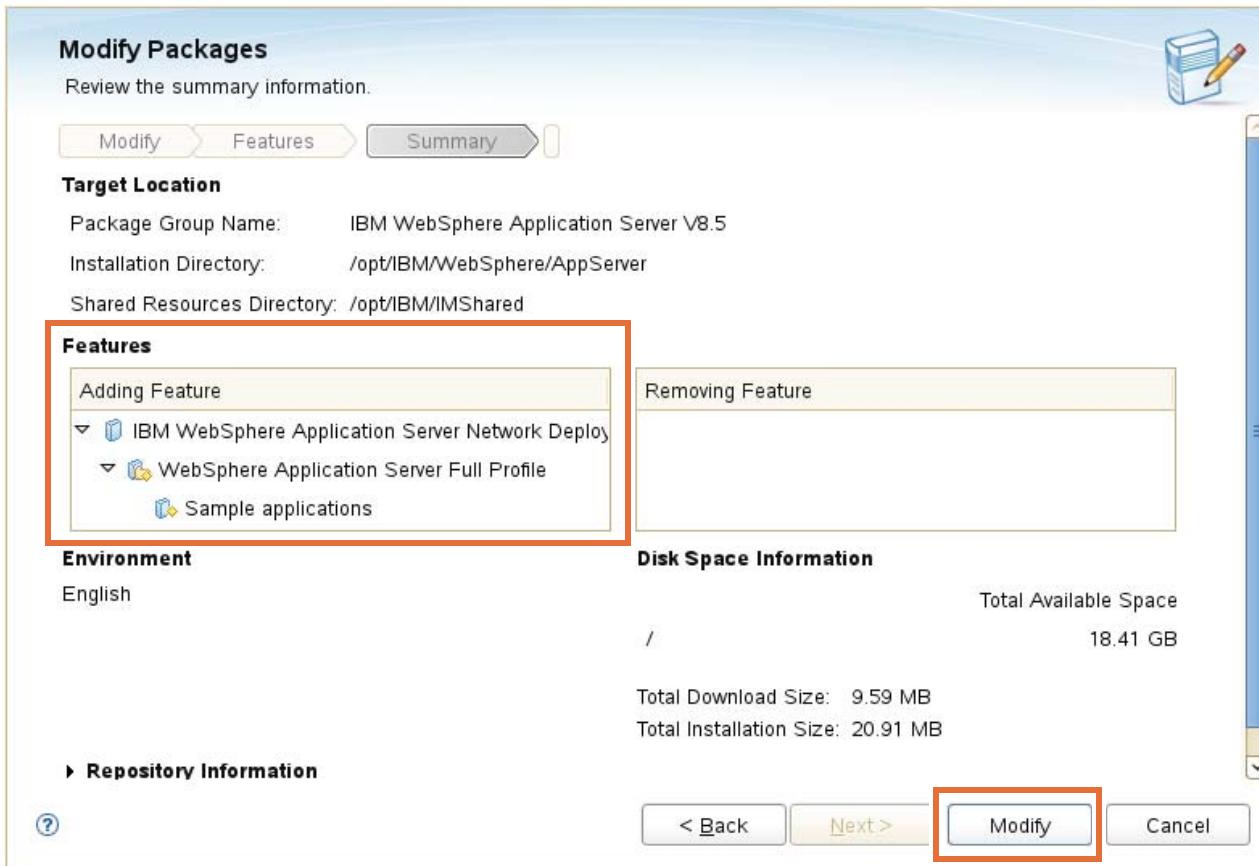


Information

In version 8.5, the only sample application that is included with the product is PlantsByWebSphere (PBW). PlantsByWebSphere is updated to use Java Platform, Enterprise Edition 6 technology. The PlantsByWebSphere sample application is available at `/opt/IBM/WebSphere/AppServer/samples` after installation.

All previous samples included in version 7 that are still relevant are online along with several new samples. You can obtain information about the samples in the WebSphere Application Server Information Center.

- __ e. In the Summary panel, verify that the added features include the sample applications. Click **Modify**.



- __ f. When the installation completes, the installation results are displayed at the top of the panel. Click **Finish**.

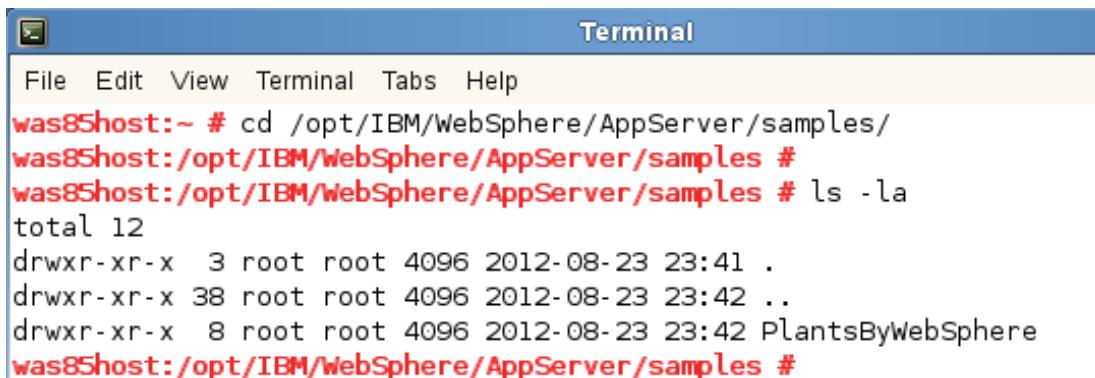


- __ g. Close Installation Manager.
- __ 2. Verify the application installation.
- __ a. Open a terminal and navigate to the following directory:

/opt/IBM/WebSphere/AppServer/samples

- ___ b. Verify that the PlantsByWebSphere sample application is in the directory by entering the following command:

```
ls -la
```



A screenshot of a terminal window titled "Terminal". The window shows a command-line interface with the following text:

```
File Edit View Terminal Tabs Help
was85host:~ # cd /opt/IBM/WebSphere/AppServer/samples/
was85host:/opt/IBM/WebSphere/AppServer/samples #
was85host:/opt/IBM/WebSphere/AppServer/samples # ls -la
total 12
drwxr-xr-x 3 root root 4096 2012-08-23 23:41 .
drwxr-xr-x 38 root root 4096 2012-08-23 23:42 ..
drwxr-xr-x 8 root root 4096 2012-08-23 23:42 PlantsByWebSphere
was85host:/opt/IBM/WebSphere/AppServer/samples #
```

Section 4: Create a profile with the Profile Management Tool

After the core product files for WebSphere Application Server are installed, you must create a profile to make the product functional. In this part of the exercise, you create an application server profile named `profile1` with the Profile Management Tool in the WebSphere Customization Toolbox. The Profile Management Tool is added to the WebSphere Customization Toolbox during the installation of WebSphere Application Server.

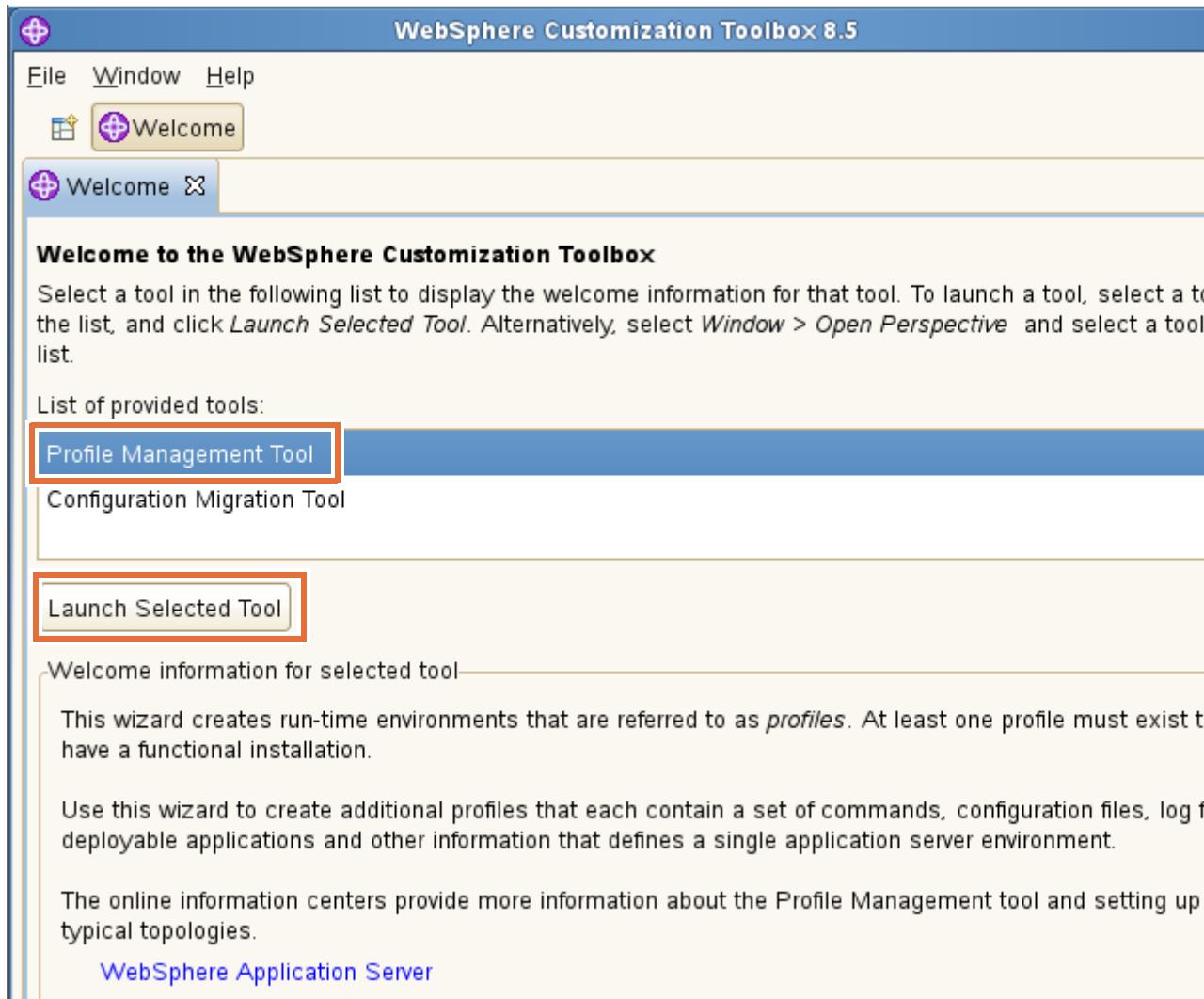


Information

Note the profile that is created is called `profile1`, which is profile number **one**. The name does not contain the character `l` or `L` (capital or lowercase “L”).

- ___ 1. Create an application server profile with the Profile Management Tool in the WebSphere Customization Toolbox.
- ___ a. In a terminal window, navigate to the following directory:
`/opt/IBM/WebSphere/AppServer/bin/ProfileManagement`
- ___ b. Enter the following command to start the WebSphere Customization Toolbox:
`./wct.sh`

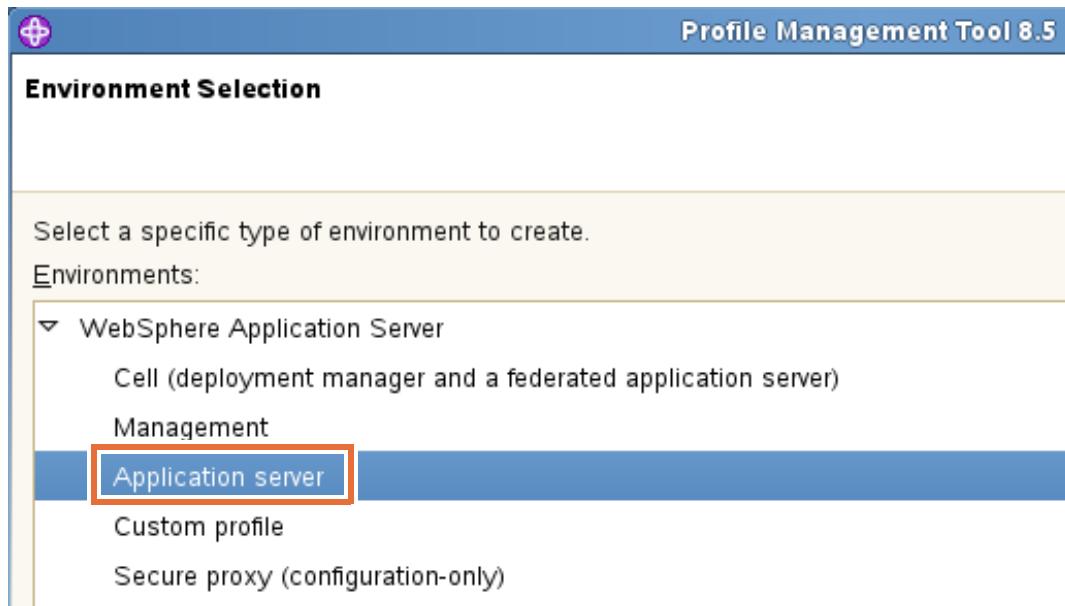
- ___ c. The WebSphere Customization Toolbox opens. Select **Profile Management Tool**, and click **Launch Selected Tool**.



- ___ d. The WebSphere Customization Toolbox opens the Profile Management Tool. Notice that there are no profiles in the list. Click **Create**.



- __ e. In the Environment Selection panel, select **Application server** and click **Next**.





Information

By creating profiles, you can create multiple runtime environments on a system without installing the core product files again. When you use the Profile Management Tool, several types of profiles can be created:

- **Cell** (deployment manager and a federated application server)
A cell creates two profiles: a management profile with a deployment manager and an application server profile. The application server is federated to the cell of the deployment manager.
- **Management**
A management profile provides the server and services for managing multiple application server environments. The administrative agent manages application servers on the same computer. A job manager provides loosely coupled management of topologies that are distributed over multiple computers. The Network Deployment edition also includes a deployment manager for tightly coupled management. Each instance of the deployment manager defines a unique cell.
- **Application server**
An application server environment runs your enterprise applications. An application server is managed from its own administrative console and functions independently from all other application servers. A new instance of a stand-alone node with a single application server is created. Stand-alone nodes have only one application server.
- **Custom profile**
A custom profile contains an empty node, which does not contain an administrative console or servers. The typical use for a custom profile is to federate its node to a deployment manager. After you federate the node, you use the deployment manager to create a server or a cluster of servers within the node.
- **Secure proxy** (configuration-only)
A secure proxy profile is for use with a DMZ secure proxy server. You cannot start the secure proxy server on the Network Deployment installation. This configuration-only profile is intended to be used only to configure the profile with the administrative console. After you configure the profile, you can export the profile configuration and then import it into the secure proxy profile in your DMZ.

- ___ f. In the Profile Creation Options panel, select **Advanced profile creation**. This selection is where you can specify your own values for some settings. Click **Next**.

Profile Creation Options



Choose the profile creation process that meets your needs. Pick the Typical option to allow the Profile Management Tool to assign a set of default configuration values to the profile. Pick the Advanced option to specify your own configuration values for the profile.

Typical profile creation

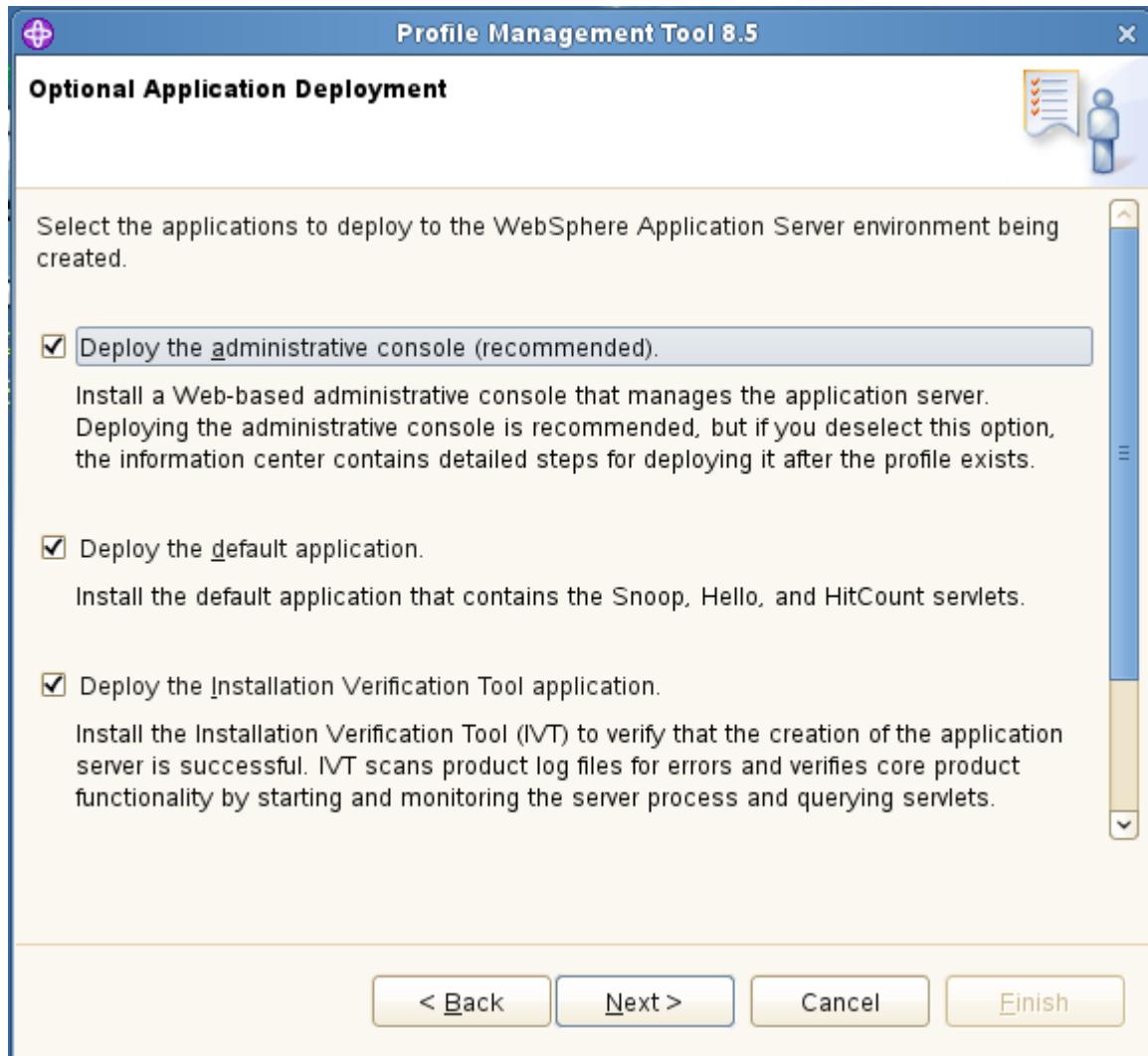
Create an application server profile that uses default configuration settings. The Profile Management Tool assigns unique names to the profile, node, and host. The tool also assigns unique port values. The administrative console and the default application will be installed. You can optionally select whether to enable administrative security. The tool might create a system service to run the application server depending on the operating system of your machine and the privileges assigned to your user account.

Note: Default personal certificates expire in one year. Select Advanced profile creation to create a personal certificate with a different expiration.

Advanced profile creation

Create application server using default configuration settings or specify your own values for settings such as the location of the profile and names of the profile, node, and host. You can assign your own port values. You can optionally choose whether to deploy the administrative console and Sample applications, and create a Web server definition. You might have the option to run the application server as a system service depending on the operating system of your machine and the privileges assigned to your user account.

- g. In the Optional Application Deployment panel, keep the default selections. Click **Next**.



- ___ h. In the Profile Name and Location panel, make the following modifications (the values that are entered are not the default values):
- **Profile name:** profile1 (profile and the number one)
 - **Profile directory:** /opt/IBM/WebSphere/AppServer/profiles/profile1

Profile Name and Location

Specify a profile name and directory path to contain the files for the run-time environment, such as configuration files, and log files. Click **Browse** to select a different directory.

Profile name:

profile1

Profile directory:

/opt/IBM/WebSphere/AppServer/profiles/profile1



Information

Server runtime performance tuning settings is a new feature in version 8. The performance monitoring infrastructure service is enabled to gather statistics so you can further tune the server yourself. Settings include:

- **Standard:** The standard settings are optimized for general-purpose usage with conservative settings.
- **Peak:** The peak settings are optimized for runtime performance in environments where updates to applications are infrequent.
- **Development:** The development settings are optimized for environments with less powerful hardware and where updates to applications are frequent.

Select the performance tuning settings that most closely match the type of environment in which the application server will run. Review the information center article on performance tuning settings before choosing a setting because additional tuning still might be necessary to optimize the performance of the server for your applications.

Server runtime performance tuning setting:

Standard

Description

The standard settings are optimized for general purpose usage with conservative settings. The performance monitoring infrastructure service is enabled to gather statistics so you can further tune the server yourself.

- ___ i. Click **Next**.
- ___ j. In the Node and Host Names panel, enter the following values (the default values must be changed for the course exercises):

- **Node name:** was85hostNode01
 - **Server name:** server1
 - **Host name:** was85host
-

Node and Host Names

Specify a node name, a server name, and a host name for this profile.

Node name:	was85hostNode01
Server name:	server1
Host name:	was85host

__ k. Click **Next**.

- __ I. The Administrative Security panel specifies whether to use initial administrative security. If selected, you specify an initial administrative user name and password, which are used for administrative activities such as console access. Verify that **Enable administrative security** is selected. Enter the following values:
- **User name:** wasadmin
 - **Password:** web1sphere
 - **Confirm password:** web1sphere

Administrative Security

Choose whether to enable administrative security. To enable security, supply a user name and password for administrative tools. This administrative user is created in a repository within the application. After you finish, you can add more users, groups, or external repositories.

Enable administrative security

User name:
wasadmin

Password:

Confirm password:

- __ m. Click **Next**.
- __ n. In the **Security Certificate (Part 1)** panel, accept the default selection. Click **Next**.

- __ o. In the **Security Certificate (Part 2)** panel, accept the default selection. Click **Next**.

Security Certificate (Part 2)

Modify the certificate information to create new certificates during profile creation. If you are importing existing certificates, verify whether the selected certificates contain the appropriate information. If the selected certificates do not, click **Import**.

Restore Defaults

Default personal certificate (a personal certificate for this profile, public and private key):

Issued to distinguished name:

`cn=was85host,ou=was85hostNode01Cell,ou=was85hostNode01,o=IBM,c=US`

Issued by distinguished name:

`cn=was85host,ou=Root Certificate,ou=was85hostNode01Cell,ou=was85hostNode01,o=IBM,c=US`

Expiration period in years:

1 

Root signing certificate (personal certificate for signing other certificates, public and private key):

Expiration period in years:

15 

- p. In the Port Values Assignment panel, accept the default values. Click **Next**.

Port Values Assignment

The values in the following fields define the ports for the application server and do not conflict with installation. Another installation of WebSphere Application Server or other programs might use run-time port conflicts, verify that each port value is unique.

[Default Port Values](#)

[Recommended Port Values](#)

Administrative console port (Default 9060):

9060

Administrative console secure port (Default 9043):

9043

HTTP transport port (Default 9080):

9080

HTTPS transport port (Default 9443):

9443

Bootstrap port (Default 2809):

2809

SIP port (Default 5060):

5060

SIP secure port (Default 5061):

5061

SOAP connector port (Default 8880):

8880

- q. In the Linux Service Definition panel, accept the default value. Do not run the application server process as a Linux service. Click **Next**.

Linux Service Definition

Choose whether to use a Linux service to run WebSphere Application Server.



[Run the application server process as a Linux service.](#)

- ___ r. In the Web Server Definition panel, accept the default value. Do not create a web server definition. Click **Next**.

Web Server Definition

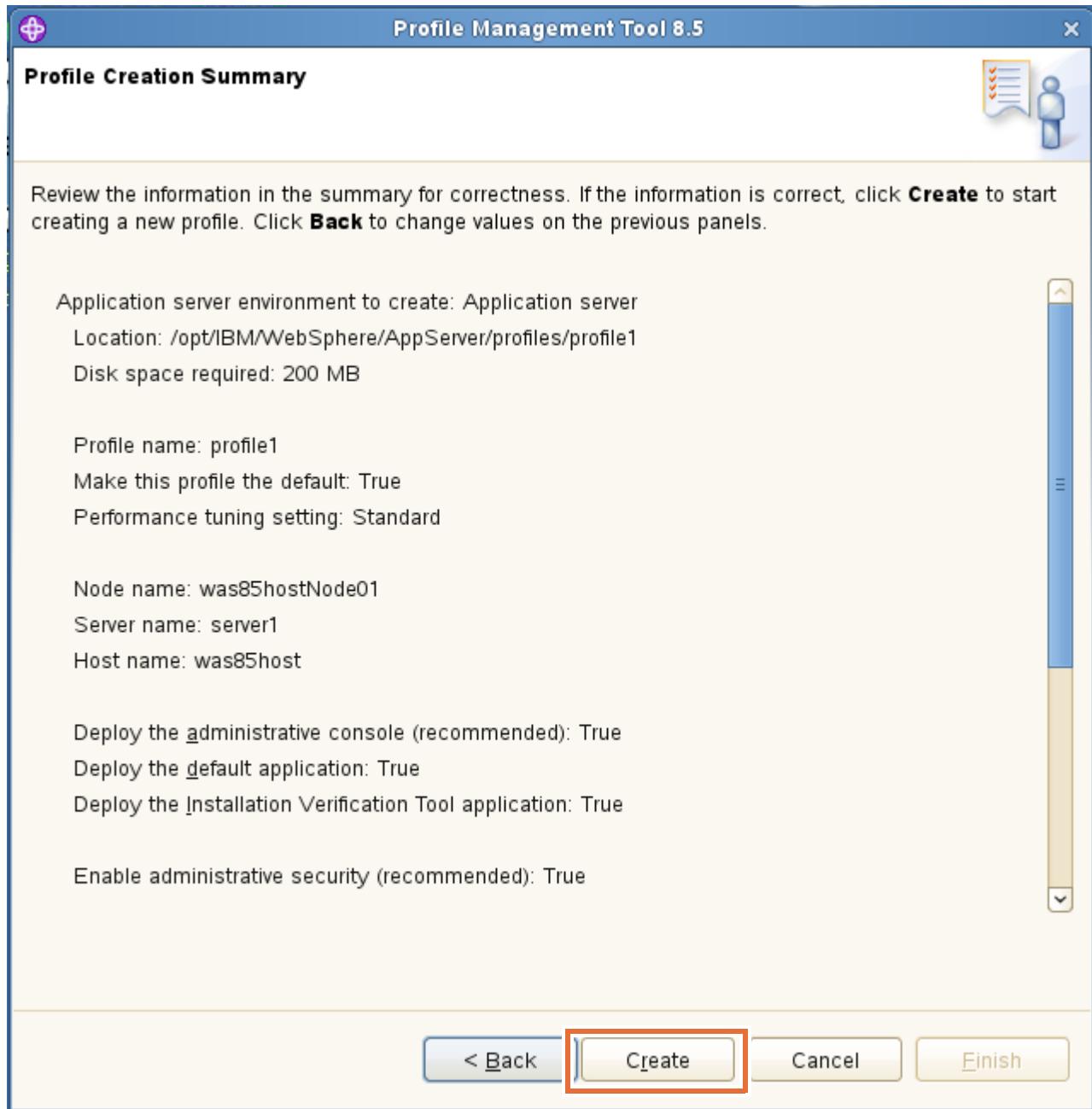
Optionally create a Web server definition if you use a Web server to route requests for dynamic content to the application server. Alternatively, you can create a Web server definition from the administrative console or a script that is generated during Web server plug-ins installation.

[Create a Web server definition](#)

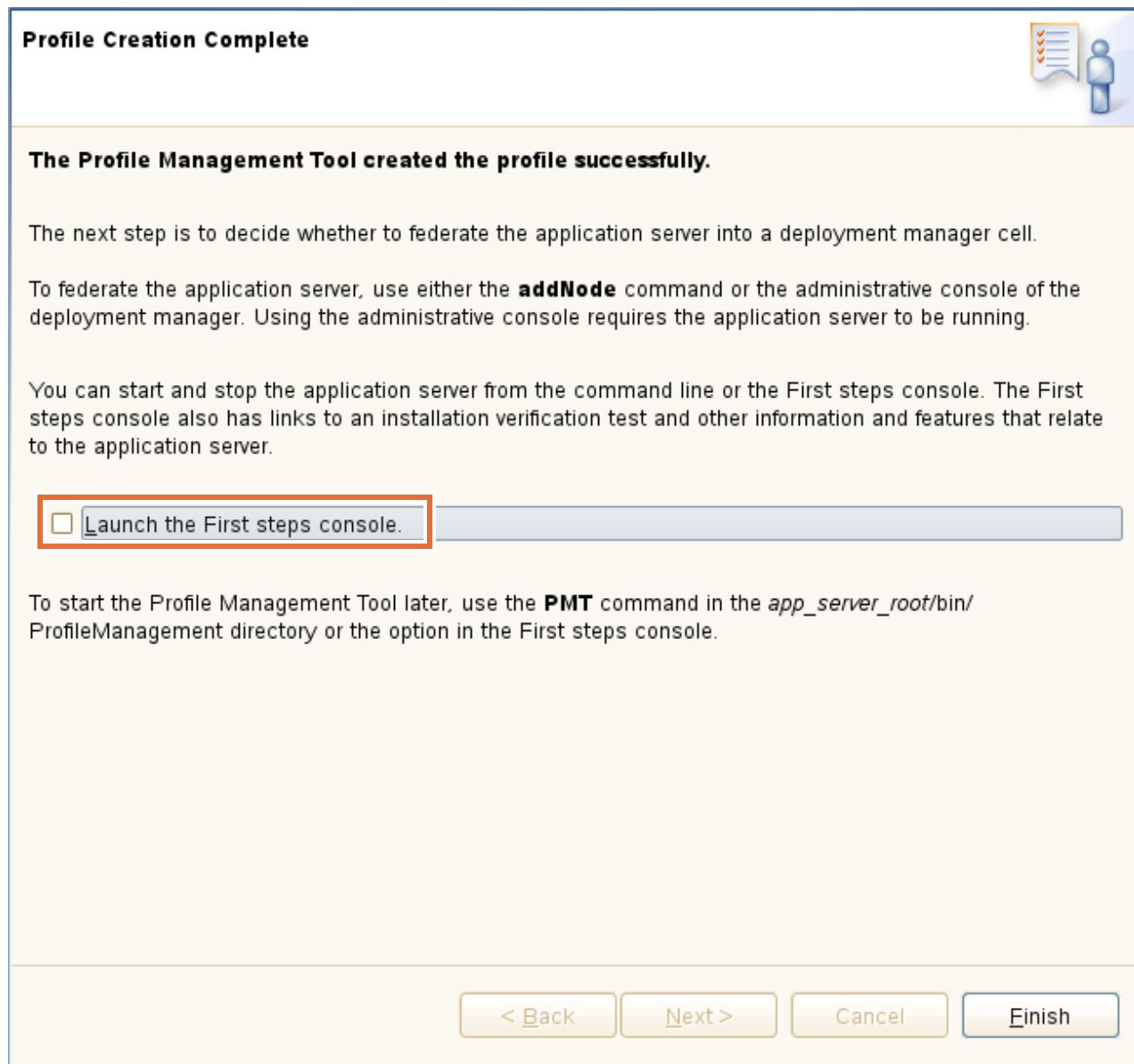
Web server type:

[IBM HTTP Server](#)

- __ s. Review the Profile Creation Summary panel. Click **Create**.



- __ t. The profile creation completes successfully in several minutes. In the Profile Creation Complete panel, clear the check box for **Launch the First steps console** and click **Finish**.



Profile Creation Complete

The Profile Management Tool created the profile successfully.

The next step is to decide whether to federate the application server into a deployment manager cell. To federate the application server, use either the **addNode** command or the administrative console of the deployment manager. Using the administrative console requires the application server to be running. You can start and stop the application server from the command line or the First steps console. The First steps console also has links to an installation verification test and other information and features that relate to the application server.

Launch the First steps console.

To start the Profile Management Tool later, use the **PMT** command in the *app_server_root/bin/ProfileManagement* directory or the option in the First steps console.

< Back Next > Cancel **Finish**

- __ u. The newly created profile is listed in the Profile Management Tool.



WebSphere Customization Toolbox 8.5

File Window Help

Profile Management Tool Welcome

Profiles

Profile name	Environment	Profile path	Create...
profile1	Application server	/opt/IBM/WebSphere/AppServer/profiles/profile1	Augment...

- ___ v. Exit the WebSphere Customization Toolbox by clicking **File > Exit**.

Section 5: Verify installation of WebSphere Application Server

The First steps console is where you can start or stop the server, access the WebSphere Information Center, and start various tools. Each application server profile has an associated First steps console.

- ___ 1. Start the First steps console for profile1.
 - ___ a. From a terminal window, navigate to the following directory:
`/opt/IBM/WebSphere/AppServer/profiles/profile1/firststeps`
 - ___ b. Enter the following command to start the First steps console:
`./firststeps.sh`
- ___ 2. Verify that the server is created and starts properly.
 - ___ a. Click **Installation verification**.

WebSphere Application Server **IBM**

First steps

Installation verification
Confirm that your server is installed and that it can start properly.

Start the server
Start the server and its applications.

Administrative console
Install and administer applications.

WebSphere Customization Toolbox
Launch this toolbox to access the Profile Management Tool and work with profiles, or to access the Migration Management Tool and migrate WebSphere Application Server 6.0, 6.1, 7.0 or 8.0 profiles to version 8.5.

Information center for WebSphere Application Server
Learn more about WebSphere Application Server and explore sample applications.

IBM Education Assistant for WebSphere software
Access multimedia content for WebSphere Application Server version 8.5 and other IBM software products.

Exit

- ___ b. The installation verification test tool runs and displays messages to indicate the verification status. Use the scroll bar to view all messages. At the bottom of the message list are the messages (some errors and warnings before the final success messages are expected):

IVTL0070I: The Installation Verification Tool verification succeeded.
IVTL0080I: The installation verification is complete.

```
First steps output - Installation verification
Node name is:was85hostNode01
Current encoding is:UTF-8
Start running the following command:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin/startServer
>ADMU0116I: Tool information is being logged in file
> /opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1/startServer.log
>ADMU0128I: Starting tool with the profile1 profile
>ADMU3100I: Reading configuration for server: server1
>ADMU3200I: Server launched. Waiting for initialization status.
>ADMU3000I: Server server1 open for e-business; process id is 15160
Server port number is:9080
IVTL0010I: Connecting to the was85host WebSphere Application Server on port: 9080
IVTL0015I: WebSphere Application Server was85host is running on port: 9080 for profile profile1
Testing server using the following URL:http://was85host:9080/ivt/ivtserver?parm2=ivtServlet
IVTL0050I: Servlet engine verification status: Passed
Testing server using the following URL:http://was85host:9080/ivt/ivtserver?parm2=ivtAddition.jsp
IVTL0055I: JavaServer Pages files verification status: Passed
Testing server using the following URL:http://was85host:9080/ivt/ivtserver?parm2=ivtejb
IVTL0060I: Enterprise bean verification status: Passed
IVTL0035I: The Installation Verification Tool is scanning the /opt/IBM/WebSphere/AppServer/profiles/
[8/24/12 0:52:25:564 EDT] 00000001 ComponentMeta W WSVR0174W: A duplicate component has been registered
[8/24/12 0:52:30:726 EDT] 00000001 WSKeyStore W CWPKI0041W: One or more key stores are available
[8/24/12 0:52:47:288 EDT] 00000001 ThreadPoolMgr W WSVR0626W: The ThreadPool setting on the server is
IVTL0040I: 2 errors/warnings are detected in the /opt/IBM/WebSphere/AppServer/profiles/profile1/loc.properties
VTL0070I: The Installation Verification Tool verification succeeded.
VTL0080I: The installation verification is complete.
```

- ___ c. Close the “First steps output - Installation verification” window.

- ___ d. From the First steps console, click **Administrative console** to start the administrative console for profile1. The application server from profile1 is started in the previous step during the installation verification test. Since the server is started, the administrative console is able to connect to the running application server.

The screenshot shows the 'First steps' section of the WebSphere Application Server interface. At the top, there's a blue header bar with the 'WebSphere Application Server' logo on the left and the 'IBM' logo on the right. Below the header, the title 'First steps' is displayed. There are several links listed:

- Installation verification**: A link to confirm server installation.
- Start the server**: A link to start the server and its applications.
- Administrative console**: A link to install and administer applications. This link is highlighted with a red rectangular box.
- WebSphere Customization Toolbox**: A link to access the Profile Management Tool and Migration Management Tool.
- Information center for WebSphere Application Server**: A link to learn more about the server and explore sample applications.
- IBM Education Assistant for WebSphere software**: A link to access multimedia content for WebSphere Application Server version 8.5 and other IBM software products.
- Exit**: A link to exit the console.

- __ e. A window opens with a warning of an untrusted connection. Expand **I Understand the Risks**. Click **Add Exception**.



This Connection is Untrusted

You have asked Firefox to connect securely to **localhost:9043**, but we can't confirm that it's secure.

Normally, when you try to connect securely, sites will present trusted identification to prove they're going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is impersonating the site, and you shouldn't continue.

[Get me out of here!](#)

► Technical Details

▼ I Understand the Risks

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **But if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

[Add Exception...](#)



Note

If you expand the Technical Details section, you see more information about the warning.

▼ Technical Details

localhost:9043 uses an invalid security certificate.

The certificate is not trusted because the issuer certificate is not trusted.
The certificate is not valid for any server names.

(Error code: sec_error_untrusted_issuer)

__ f. Click **Confirm Security Exception**.



Information

There are two issues here, and both involve the fact that the browser is using SSL to communicate with the administrative console application. To establish the SSL connection, the application server presents a certificate to the browser. The browser gives messages about two aspects of this certificate and asks if you want to continue. The two issues are as follows:

- **The host name for the certificate does not match what you entered in the URL.** In this case, the URL might use localhost, and the certificate was created (during profile creation) with a mapping to the host IP address. To resolve this issue, you can specify an appropriate host name during the certificate creation. You can also use the host name in the certificate when you enter the URL into the browser.
- **An unknown certificate authority signed the certificate that is presented to the browser.** By default, internal self-signed certificates are created for the

WebSphere cell. As such, the browser does not necessarily trust this unknown certificate authority and therefore asks the user if it is appropriate to proceed.

— g. Log in to the administrative console with the following details:

- User ID: `wasadmin`
- Password: `websphere`

— h. Click **Log in**.



Information

The user ID `wasadmin` was created during the profile creation process. It does not exist in the operating system registry or an LDAP registry, but instead exists in a file-based registry within the application server configuration. Now that the profile is created, security can be reconfigured to use any user registry.

3. The main page for the WebSphere Integrated Solutions Console is displayed and looks like the following screen capture:

The screenshot shows the 'Welcome' page of the WebSphere Integrated Solutions Console. The left sidebar lists various administrative tasks under 'View: All tasks'. The central content area displays a welcome message about the console's purpose and a table showing installed product suites. The right sidebar contains information about the integrated solutions and licensed materials.

Suite Name	Version
WebSphere Application Server	8.5.5.0

4. Verify that the DefaultApplication is installed and is running.
- a. Using the administrative console navigation tree, click **Applications > Application Types > WebSphere enterprise applications**.



The application status for the DefaultApplication is displayed as a green arrow to indicate that the application is running. If you place your cursor over the arrow, a message indicates that the application is started.

Enterprise Applications

Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

+ Preferences

Start Stop Install Uninstall Update Rollout Update Remove File Export

Select Name Application Status

You can administer the following resources:

<input type="checkbox"/>	DefaultApplication	
<input type="checkbox"/>	ivtApp	
<input type="checkbox"/>	query	

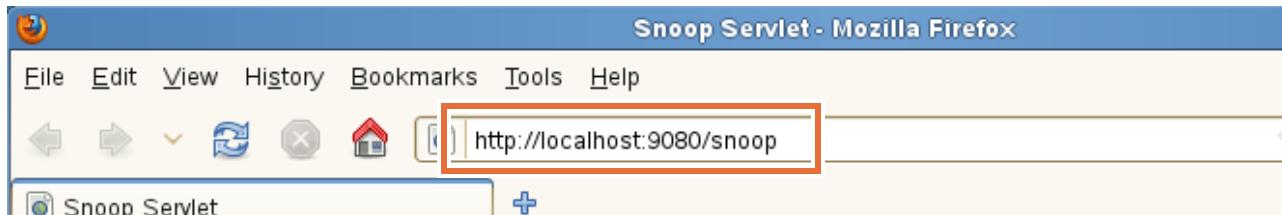
Total 3



- ___ b. Open another browser window and enter the following URL:

`http://localhost:9080/snoop`

The URL runs a servlet that is called Snoop, which comes with the DefaultApplication. Snoop displays a page with information about the runtime environment of the server. The page provides further confirmation that the application server is operating correctly.



Snoop Servlet - Request/Client Information

Requested URL:

http://localhost:9080/snoop

Servlet Name:

Snoop Servlet

Request Information:

Request method	GET
Request URI	/snoop
Request protocol	HTTP/1.1
Servlet path	/snoop

- ___ c. Close the browser that is running Snoop servlet.
- ___ d. Close the administrative console by clicking **Logout**. The administrative console is examined more in later exercises. Close the browser.
- ___ 5. Exit from the First steps console.

Section 6: Create a backup of profile1

- 1. Before you continue, create a backup of profile1 with the `backupConfig` command. The `backupConfig` command is a utility to back up the configuration of your profile to a compressed file. You can later restore this configuration if needed. When the `backupConfig` command runs, it first stops the application server before you create the backup file.
- a. Run the `backupConfig` command. From a terminal window, navigate to:
`/opt/IBM/WebSphere/AppServer/profiles/profile1/bin`
- b. Create the backup by entering the following command:
`./backupConfig.sh`
- c. As the backup process starts, you are challenged for a user ID and password. Provide the following values:
- For **User Identity**, enter: `wasadmin`
 - For **User Password**, enter: `websphere`



- d. Click **OK**.

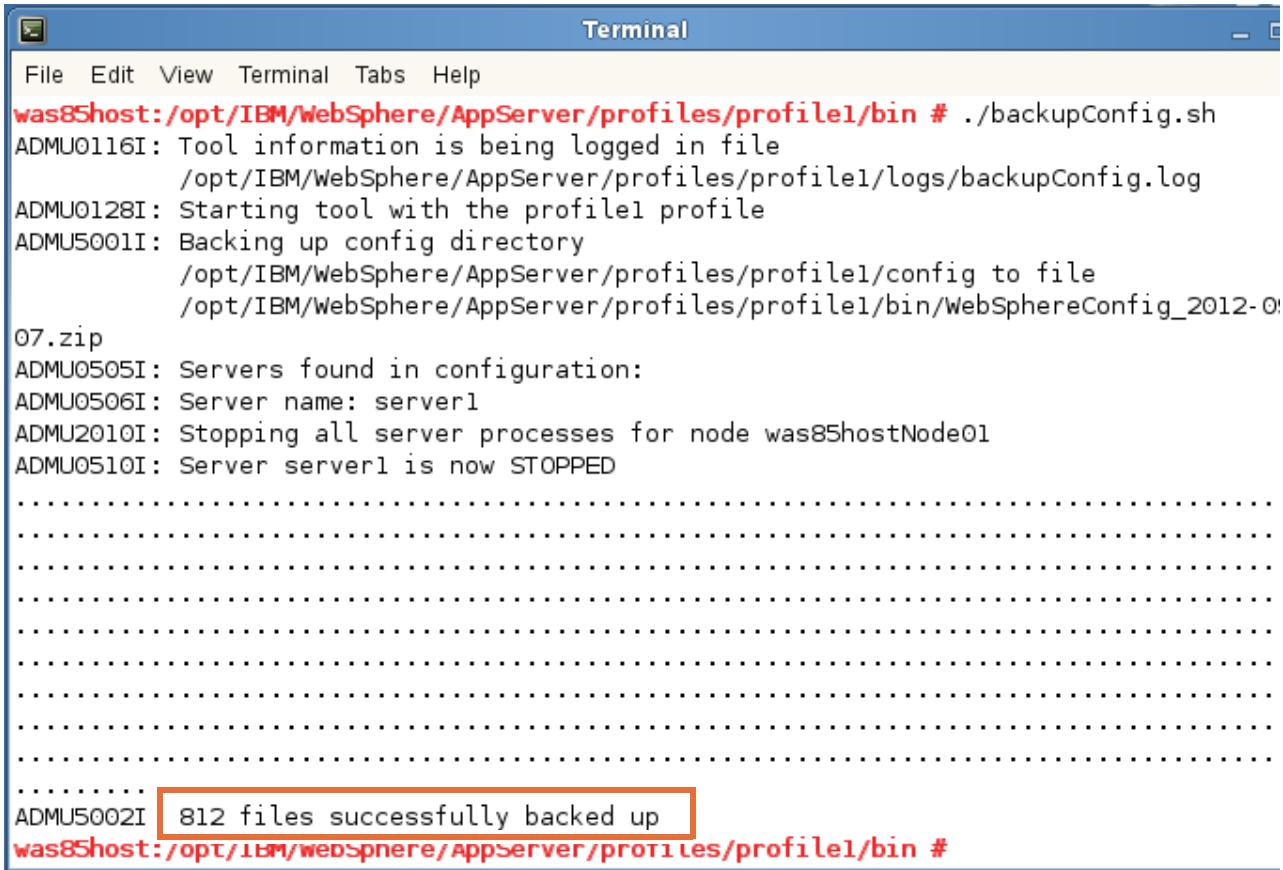


Information

You are prompted for credentials because administrative security is enabled during the creation of the profile by default. As such, all administrative functions, including backups, administrative console access, and wsadmin scripts, require authentication.

During the rest of the labs, enter the administrative user ID and password when prompted. The lab instructions do not always indicate this step.

- ___ e. When the backup finishes, a message indicates the number of files that were backed up successfully.



```

Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./backupConfig.sh
ADMU0116I: Tool information is being logged in file
    /opt/IBM/WebSphere/AppServer/profiles/profile1/logs/backupConfig.log
ADMU0128I: Starting tool with the profile1 profile
ADMU5001I: Backing up config directory
    /opt/IBM/WebSphere/AppServer/profiles/profile1/config to file
    /opt/IBM/WebSphere/AppServer/profiles/profile1/bin/WebSphereConfig_2012-01-07.zip
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: server1
ADMU2010I: Stopping all server processes for node was85hostNode01
ADMU0510I: Server server1 is now STOPPED
ADMU5002I 812 files successfully backed up
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #

```



Information

By default, all servers on the node are stopped before the backup is made to prevent partially synchronized information from being saved. The `-nostop` option can be used with the `backupConfig` command to prevent the servers from being stopped before you back up the configuration.

- ___ 2. Verify that the backup completed.

The command creates a backup file that is called `WebSphereConfig_<date>.zip` with the current date and places the compressed file in the `/opt/IBM/WebSphere/AppServer/profiles/profile1/bin/` directory. To distinguish between multiple backups, modify the name with something more descriptive, such as the name of the profile.

- __ a. Copy the backup file to another directory for safe keeping. Enter the following command:

```
cp WebSphereConfig_<YYYY-MM-DD>.zip
/usr/software/backups/backup_profile1.zip
```

```
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # cp WebSphereConf:
-09-07.zip /usr/software/backups/backup_profile1.zip
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```



Information

If you want to restore the configuration directory structure later, you can use the `restoreConfig` command. You must specify the name of the backup file. The command restores the directory:

```
/opt/IBM/WebSphere/AppServer/profiles/profile1/config
```

Section 7: Start and stop the WebSphere Application Server

Use the command line to start and stop the WebSphere Application Server in profile1. A number of command-line scripts can help you manage the application server. Three of the most useful are `startServer`, `stopServer`, and `serverStatus`.

- __ 1. Check the status of the application server.
- __ a. Using the command line, navigate to the `bin` directory from profile1.
- __ b. Use the following command to verify that `server1` is running:

```
./serverStatus.sh server1
```

The terminal window shows the output of the `./serverStatus.sh server1` command. It includes messages about tool logging, starting the tool with profile1, retrieving server status for server1, and indicating that the server is stopped.

```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./serverStatus.sh
ADMU0116I: Tool information is being logged in file
/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1/serverSta
ADMU0128I: Starting tool with the profile1 profile
ADMU0500I: Retrieving server status for server1
ADMU0509I: The Application Server "server1" cannot be reached. It appears to be
stopped.
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```



Information

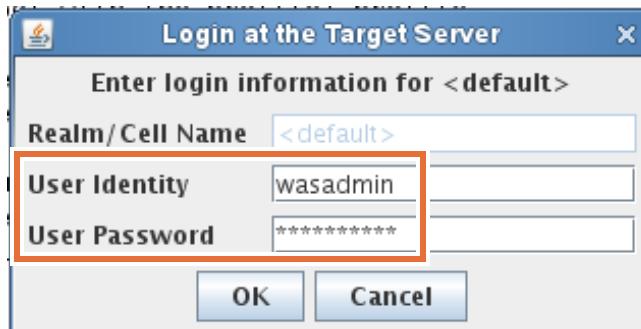
You can also run the `serverStatus` command with the `-all` option to give more details on all application servers on the node.

2. Start the application server.
- a. Enter the following command to start server1:

```
./startServer.sh server1
```

```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./startServer.sh
ADMU0116I: Tool information is being logged in file
/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1/startServer.log
ADMU0128I: Starting tool with the profile1 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 14253
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```

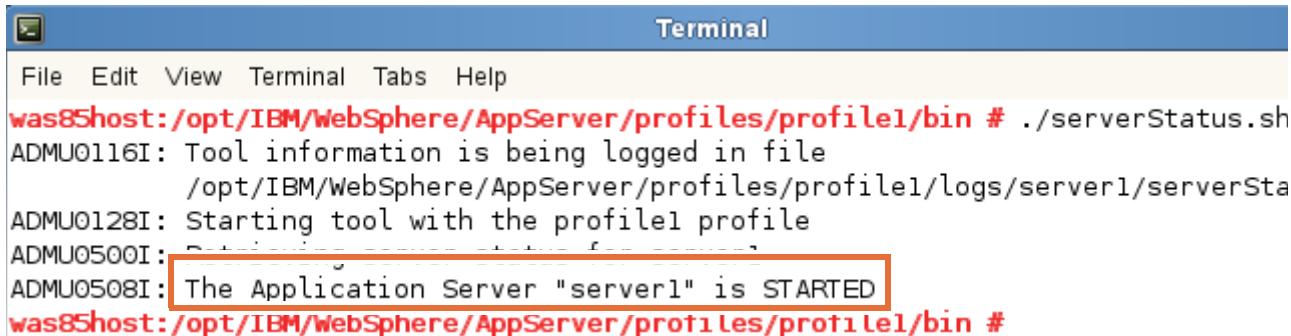
- b. Enter the following the command to verify the status:
- ```
./serverStatus.sh server1
```
- c. When prompted to authenticate, use wasadmin for User Identity and web1sphere for User Password.



### Information

In the previous use of the `serverStatus` command, you are not prompted to authenticate. The application server was not running and therefore did not challenge the client when it attempted to retrieve information from the server. Since `server1` is now running, it challenges the client application before you provide any information.

- \_\_\_ d. After the authentication is complete, the server status is displayed indicating that server1 is started.



A screenshot of a terminal window titled "Terminal". The window shows the command `./serverStatus.sh` being run. The output text is in red and includes log messages from ADMU0116I, ADMU0128I, ADMU0500I, and ADMU0508I. The line "The Application Server "server1" is STARTED" is highlighted with a red box.

```
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./serverStatus.sh
ADMU0116I: Tool information is being logged in file
 /opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1/serverSta
ADMU0128I: Starting tool with the profile1 profile
ADMU0500I:
ADMU0508I: The Application Server "server1" is STARTED
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```

- \_\_\_ 3. Stop the server.

- \_\_\_ a. Enter the following command to start server1 (use wasadmin and websphere when prompted to authenticate):

```
./stopServer.sh server1
```

- \_\_\_ b. Enter the following the command to verify the status:

```
./serverStatus.sh server1
```

## **Section 8: Explore the directory structure of WebSphere Application Server**

Now that WebSphere Application Server is installed, examine the directory structure and review what you installed.

- \_\_\_ 1. Explore the WebSphere Application Server `profile1` directory.

- \_\_\_ a. Using a terminal window, navigate to the following directory:

```
/opt/IBM/WebSphere/AppServer/profiles/profile1
```

- \_\_\_ 2. Review the subdirectories and their contents:

- bin: programs, scripts, and DLLs
- config: configuration files
- configuration: configuration settings
- consolepreferences
- etc: dummy key ring, keytab files, plug in keys
- firststeps: firststeps utility
- installableApps: applications that can be installed
- installedApps: applications that are installed in application server
- installedConnectors: installed resource adapters
- installedFilters

- logs: trace and log files
- properties: configuration property files that WebSphere uses
- servers: server configuration
- temp: temporary area for files that are created during JSP processing
- tranlog: transaction log files
- wstemp: temporary area for events

## Section 9: Check the installation log files

A number of log files are created during the installation and profile creation process. It is useful to check these files to verify that the installation completed successfully.

- 1. View the log file for the profile creation. This log file shows the results for the creation of profile1.
  - a. In a terminal window, navigate to the following directory:  
`/opt/IBM/WebSphere/AppServer/logs/manageprofiles`
  - b. Using an editor such as gedit, view the `profile1_create.log` file. This log file records creation events that occurred when you created profile1.  
`gedit profile1_create.log`

```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer : cd logs/manageprofiles
was85host:/opt/IBM/WebSphere/AppServer/logs/manageprofiles #
was85host:/opt/IBM/WebSphere/AppServer/logs/manageprofiles # gedit profile1_create.log
```

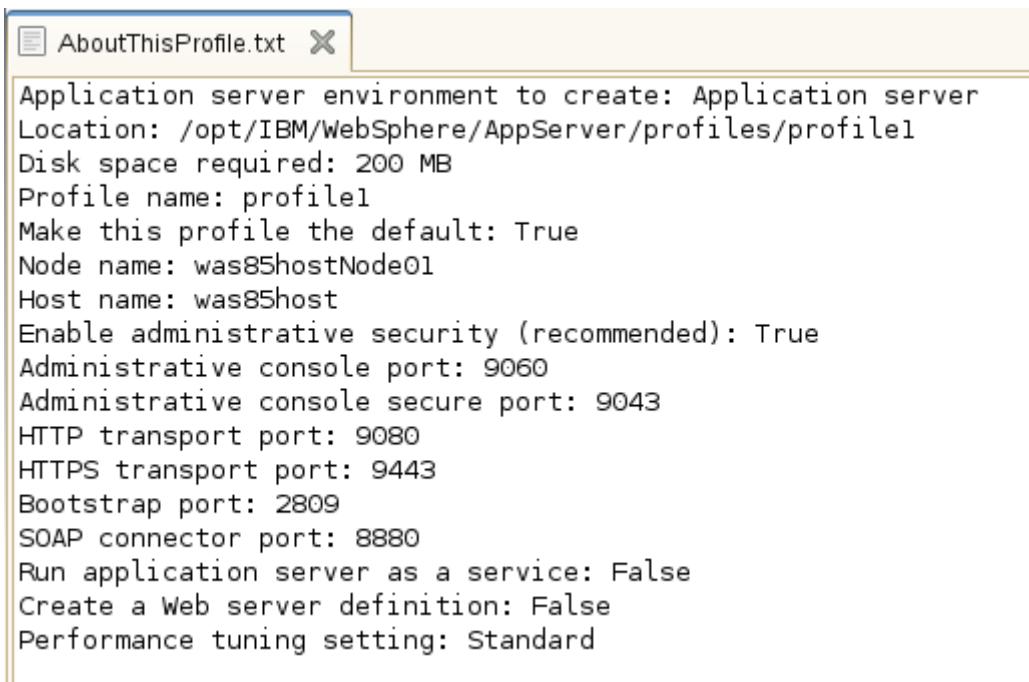
- c. Look near the end of the file for the log message “INSTCONFSUCCESS: Success: Profile profile1 now exists” to verify that profile1 was created successfully.

```
profile1_create.log (/opt/IBM/WebSphere/AppServer/logs/manageprofiles) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
profile1_create.log X
<method>executeWSProfileAccordingToMode</method>
<thread>1</thread>
<message>INSTCONFSUCCESS: Success: Profile profile1 now exists. Please consult /opt/IBM/websphere/AppServer/profiles/profile1 for more information about this profile.</message>
</record>
<record>
 <date>2012-08-24T12:43:03</date>
```

- 2. View the log files in the profile log directory.

- \_\_ a. In the terminal window, use the following command to navigate to the log directory for profile1:  

```
cd /opt/IBM/WebSphere/AppServer/profiles/profile1/logs
```
- \_\_ b. Using gedit, open and view the `backupConfig.log` file. This log records events that occur when you create a backup of the configuration directory structure.
- \_\_ c. Using gedit, open and view the `ivtClient.log` file. This file logs results from the installation verification command.
- \_\_ d. Using gedit, open and view the `AboutThisProfile.txt` file. This file logs information about the profile, including the profile name, the node and host names, and a number of other items.



```
Application server environment to create: Application server
Location: /opt/IBM/WebSphere/AppServer/profiles/profile1
Disk space required: 200 MB
Profile name: profile1
Make this profile the default: True
Node name: was85hostNode01
Host name: was85host
Enable administrative security (recommended): True
Administrative console port: 9060
Administrative console secure port: 9043
HTTP transport port: 9080
HTTPS transport port: 9443
Bootstrap port: 2809
SOAP connector port: 8880
Run application server as a service: False
Create a Web server definition: False
Performance tuning setting: Standard
```

## Section 10:Check the application server log files

Start WebSphere Application Server in profile1 and view the log files.

- \_\_ 1. View the log file for the application server startup. This log file shows the startup activities for the application server.
  - \_\_ a. In a terminal window, navigate to the following directory:  
`/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1`
  - \_\_ b. Using an editor such as gedit, view the `startServer.log` file. This log file records creation events that occurred during the server startup.  

```
gedit startServer.log
```

- \_\_\_ c. Look through the contents of the file. Look for the message that indicates the server is open for e-business.



```
[9/9/12 17:02:29:879 EDT] 00000001 WorkSpaceMana A WKSP0500I: Workspace configuration consistency check is disabled.
[9/9/12 17:02:30:380 EDT] 00000001 AdminTool A ADMU3200I: Server launched. Waiting for initialization status.
[9/9/12 17:03:05:280 EDT] 00000001 AdminTool A ADMU3000I: Server server1 open for e-business; process id is 18175
```

Ln 1, Col 1      INS

- \_\_\_ 2. Open the `SystemOut.log` file for `server1` and view the contents.



### Information

The `SystemOut.log` file contains the standard output from the Java virtual machine (JVM) running the application server. The log file contains more detailed messages, indicating the steps that are completed during startup of the server. Steps that are documented in the log include security initialization, messaging initialization, registering resources in the JNDI namespace, EJB initialization, and many others. The log file also contains messages from application `System.out` print line code.

- \_\_\_ a. From the same directory, use an editor such as gedit to open the `SystemOut.log` file.
- \_\_\_ b. Look through the contents and notice that the startup log entries that are in the `startServer.log` file are also in the `SystemOut.log` file.

- \_\_ 3. Start the application server while you watch the server log file.  
\_\_ a. In the same directory, use the following command to watch the server log file:

```
tail -f SystemOut.log
```

```
wec9hnet:/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1 tail -f SystemOut.log
[9/9/12 23:43:20:475 EDT] 0000004f ActivitySessi I WACS0049I: The ActivitySession
service is stopping.
[9/9/12 23:43:20:519 EDT] 0000004f ObjectPoolSer I OBPL0011I: The Object Pool serv
ice is stopping.
[9/9/12 23:43:20:521 EDT] 0000004f distSecurityC I securityServiceStarted is false
[9/9/12 23:43:20:526 EDT] 0000004f CGBridgeSubsc I CWRCB0104I: The core group brid
ge service has stopped the subscription router.
[9/9/12 23:43:20:539 EDT] 0000004f CGBridgeServi I CWRCB0103I: The core group brid
ge service has stopped.
[9/9/12 23:43:20:540 EDT] 0000004f DragDropDeplo I CWLDD0004I: Stopping monitored
directory application deployment service...
[9/9/12 23:43:20:541 EDT] 0000004f DragDropDeplo I CWLDD0005I: Monitored directory
application deployment service is stopped.
[9/9/12 23:43:20:549 EDT] 0000004f TCPChannel I TCPC0002I: TCP Channel TCPInbou
ndChannel_ipcc.Default_IPC_Connector_Name has stopped listening on host localhost (
IPv4: 127.0.0.1) port 9633.
[9/9/12 23:43:20:989 EDT] 0000004f FailureScopeC A WTRN0105I: The transaction serv
ice has shutdown successfully with no transactions requiring recovery.
[9/9/12 23:43:21:008 EDT] 0000004f ServerCollabo A WSVR0024I: Server server1 stopp
ed
```



## Information

The tail utility makes it easy to monitor what is being actively written to a text-based log file.

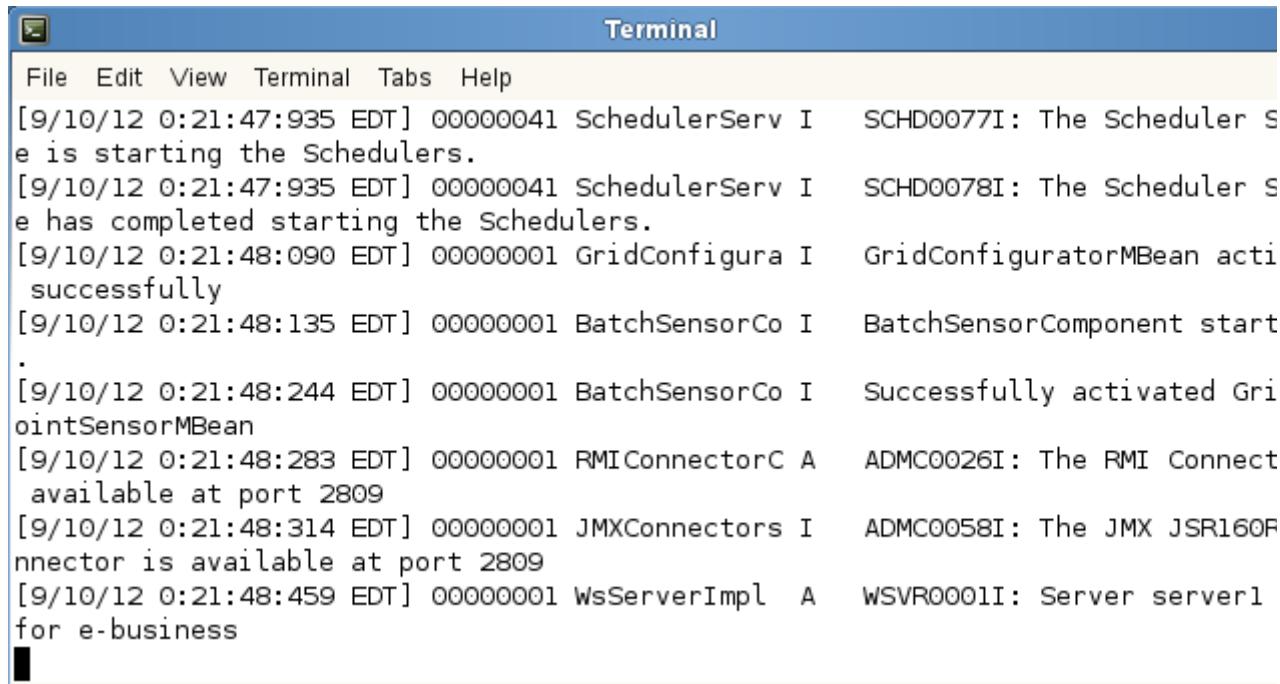
- Command: `tail <filename>`  
This command displays the last few lines of the file `<filename>`.
- Command: `tail - f <filename>`  
This command continuously updates your display when new messages are added to the file.

- \_\_\_ b. Open a second terminal window and navigate to the `bin` directory for profile1.  
\_\_\_ c. Use the following command to start the application server:

```
./startServer.sh server1
```

```
Terminal
File Edit View Terminal Tabs Help
was85host:~/Desktop $ cd /opt/IBM/WebSphere/AppServer/profiles/profile1/bin
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./startServer.sh
server1
ADMU0110I: Tool information is being logged in file
 /opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1/startServer.log
ADMU0128I: Starting tool with the profile1 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 19348
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```

- \_\_\_ d. Return to the other terminal that is running the `tail` command. Notice that as the application server starts, the log messages are constantly added to the end of the window. Look for the message “Server server1 open for e-business”, which indicates the server is ready.



The screenshot shows a terminal window titled "Terminal". The menu bar includes "File", "Edit", "View", "Terminal", "Tabs", and "Help". The main pane displays log messages from the server:

```

Terminal
File Edit View Terminal Tabs Help
[9/10/12 0:21:47:935 EDT] 00000041 SchedulerServ I SCHED0077I: The Scheduler Service is starting the Schedulers.
[9/10/12 0:21:47:935 EDT] 00000041 SchedulerServ I SCHED0078I: The Scheduler Service has completed starting the Schedulers.
[9/10/12 0:21:48:090 EDT] 00000001 GridConfigura I GridConfiguratorMBean activated successfully
[9/10/12 0:21:48:135 EDT] 00000001 BatchSensorCo I BatchSensorComponent start
.
[9/10/12 0:21:48:244 EDT] 00000001 BatchSensorCo I Successfully activated GridPointSensorMBean
[9/10/12 0:21:48:283 EDT] 00000001 RMIConnectorC A ADMC0026I: The RMI Connector is available at port 2809
[9/10/12 0:21:48:314 EDT] 00000001 JMXConnectors I ADMC0058I: The JMX JSR160 Connector is available at port 2809
[9/10/12 0:21:48:459 EDT] 00000001 WsServerImpl A WSVR0001I: Server server1 for e-business

```

- \_\_\_ e. When you are done with using the `tail` command, end it by entering **Ctrl-C** in the terminal window.
- \_\_\_ 4. Open and view the `SystemErr.log` file. This log contains the standard error output from the Java virtual machine (JVM) running the application server. This file can have numerous messages, but does not include any error messages if the server started correctly.

## **Section 11:Correct the website security certificate error (optional)**

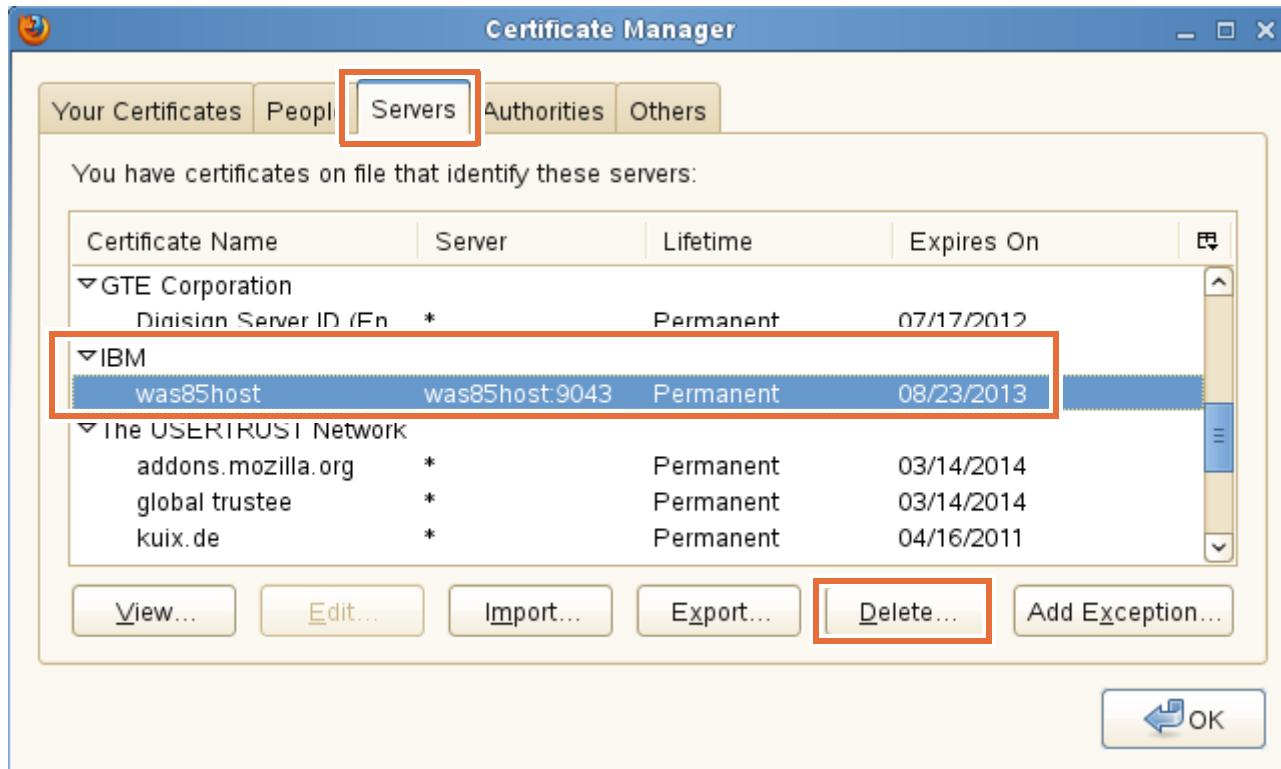
When you first open the administrative console after installation, you might receive a website security certificate error window. As is mentioned earlier, there are two reasons for seeing this warning. This part of the exercise examines the process of exploring and correcting the causes for the warning.

- \_\_\_ 1. Remove the certificate exception from your Firefox browser. In a previous section, you added an exception in your browser to remove the warning. For this part of the exercise, you want to see the exception.
  - \_\_\_ a. Open a Firefox window and use the following URL to browse to the administrative console:
   
`http://localhost:9060/ibm/console`
  - \_\_\_ b. If you see a warning message, proceed to the next step.

- \_\_ c. If you see the administrative console login screen, an exception is added and must be removed. From the Firefox window menu, click **Edit > Preferences**.
- \_\_ d. Select the **Advanced** icon and the **Encryption** tab. Click **View Certificates**.



- \_\_ e. In the **Certificate Manager** window, select the **Servers** tab. Scroll down and select all certificates named **was85host** (under IBM). Click **Delete**.



- \_\_ f. Click **OK** in the confirmation dialog box.
- \_\_ g. Click **OK** to close the Certificate Manager dialog box.

- \_\_ h. Click **Close** to close the Firefox Preference window.
- \_\_ 2. Add a certificate exception for localhost.
  - \_\_ a. Open a Firefox window and use the following URL to go to the administrative console:  
`http://localhost:9060/ibm/console`
  - \_\_ b. Now you see a warning of an untrusted connection. Expand **Technical Details**.



## This Connection is Untrusted

You have asked Firefox to connect securely to **localhost:9043**, but we can't confirm it is secure.

Normally, when you try to connect securely, sites will present trusted identification going to the right place. However, this site's identity can't be verified.

### What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is impersonating the site, and you shouldn't continue.

[Get me out of here!](#)

### ▼ Technical Details

localhost:9043 uses an invalid security certificate.

The certificate is not trusted because the issuer certificate is not trusted.

The certificate is not valid for any server names.

(Error code: sec\_error\_untrusted\_issuer)

### ► I Understand the Risks

- \_\_ c. Notice that the connection is coming into **localhost** (not **was85host**) and that two issues are listed:
  - The certificate is not trusted because the issuer certificate is not trusted.
  - The certificate is not valid for any server names.

The first is because the certificate is signed by (issued by) someone the browser does not know about (not a known certificate authority). The second is because the host name associated with the certificate (was85host) does not match the host name in the URL (localhost).

- \_\_ d. Expand I Understand the Risks and click Add Exception.



## This Connection is Untrusted

You have asked Firefox to connect securely to **localhost:9043**, but we can't confirm that you're secure.

Normally, when you try to connect securely, sites will present trusted identification to prove you're going to the right place. However, this site's identity can't be verified.

### What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

[Get me out of here!](#)

### ► Technical Details

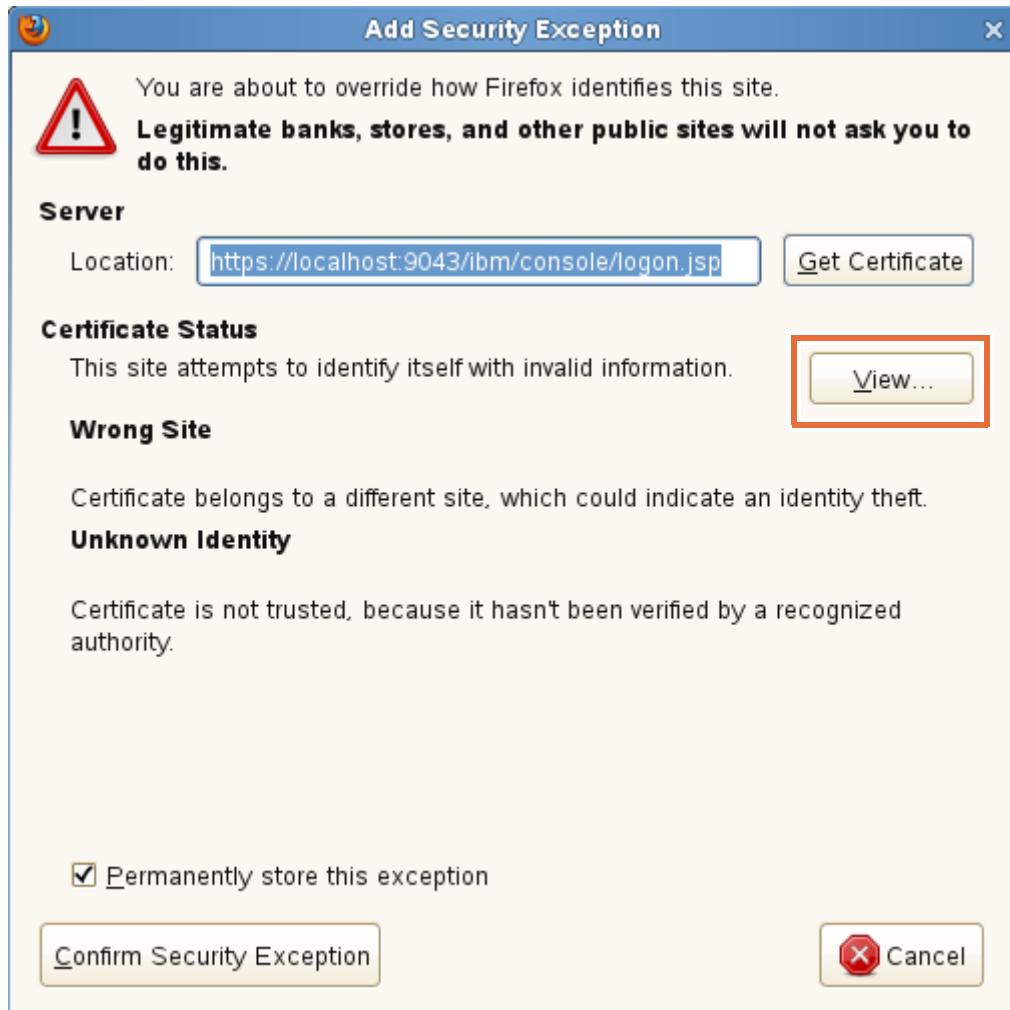
#### ▼ I Understand the Risks

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **But if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

[Add Exception...](#)

- \_\_ e. Click **View** to see the details for the certificates.

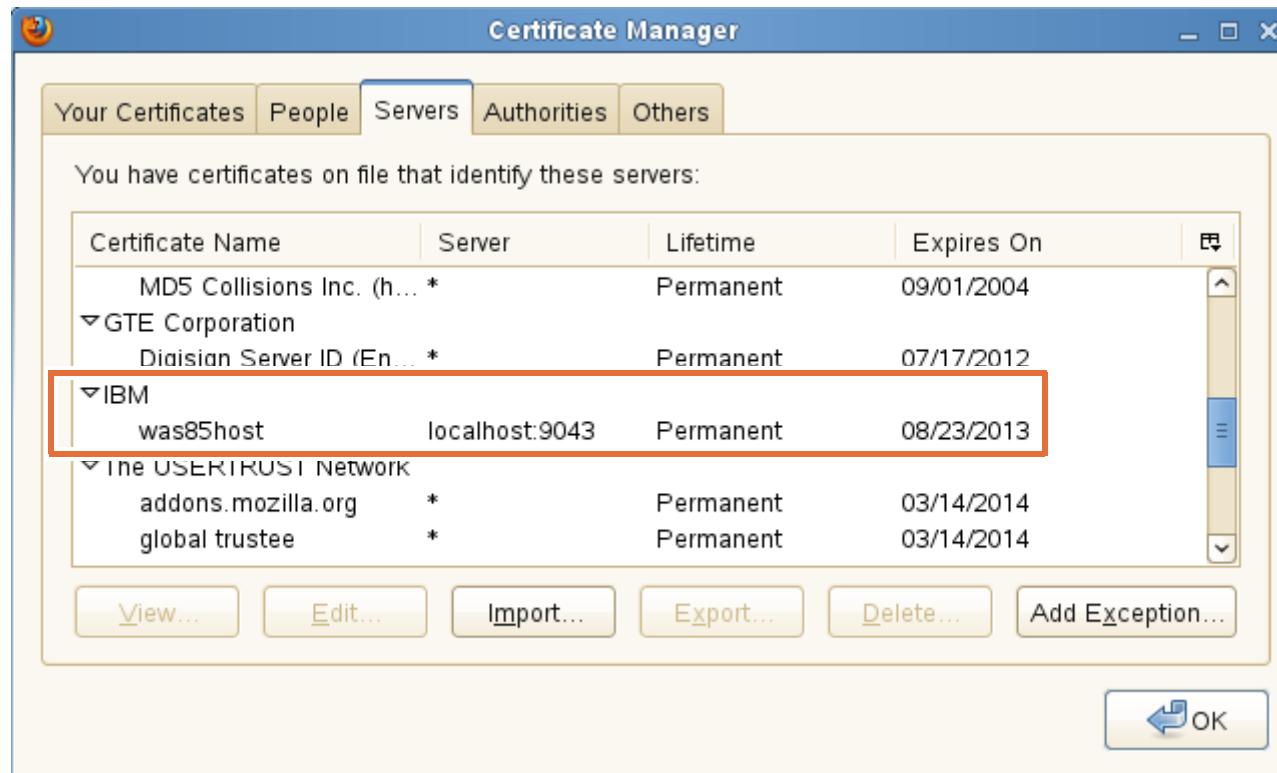


- \_\_\_ f. In the **Certificate Viewer**, notice that the certificate is issued to the host name **was85host**. It is also interesting to note that was85host (and not a known certificate authority) issued it.



- \_\_\_ g. Click **Close** to exit the Certificate Viewer.
- \_\_\_ h. On the Add Security Exception window, click **Confirm Security Exception** to add the certificate to the list of certificates that the browser trusts.
- \_\_\_ i. The console login window is displayed.
- \_\_\_ j. Return to the Firefox preferences by clicking **Edit > Preferences**.
- \_\_\_ k. Select the **Advanced** icon and the **Encryption** tab.

- \_\_ I. Click **View Certificates** and the **Servers** tab. Scroll down and look for the certificate named was85host under IBM. Notice that now an exception is listed for server **localhost**.



- \_\_ m. Close out of the open dialog boxes.  
\_\_ 3. Add a certificate exception for was85host.  
\_\_ a. Open a Firefox window and use the following URL to go to the administrative console:  
`http://was85host:9060/ibm/console`

- \_\_\_ b. Now you see a warning of an untrusted connection. Expand **Technical Details**.



## This Connection is Untrusted

You have asked Firefox to connect securely to **was85host:9043**, but we can't verify that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to let you know that you are going to the right place. However, this site's identity can't be verified.

### What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

[Get me out of here!](#)

#### ▼ Technical Details

was85host:9043 uses an invalid security certificate.

The certificate is not trusted because the issuer certificate is not trusted.

(Error code: sec\_error\_untrusted\_issuer)

- \_\_\_ c. Notice that the connection is coming into **was85host** (not **localhost**) and that there is only one issue listed:

- The certificate is not trusted because the issuer certificate is not trusted.

The second issue that localhost had is no longer there since the host name in the URL matches the host name in the certificate.

- \_\_ d. Expand **I Understand the Risks** and click **Add Exception**.



## This Connection is Untrusted

You have asked Firefox to connect securely to **was85host:9043**, but we can't connect because your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification so that you are going to the right place. However, this site's identity can't be verified.

### What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

**Get me out of here!**

#### ► Technical Details

#### ▼ I Understand the Risks

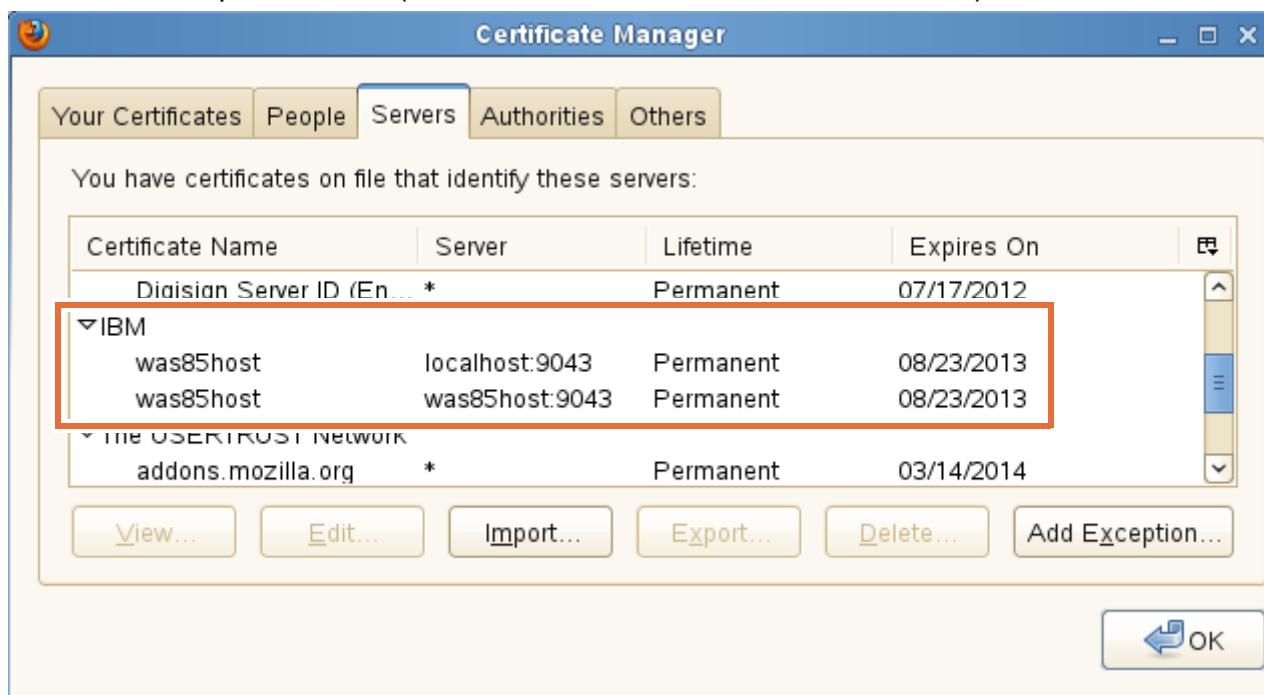
If you understand what's going on, you can tell Firefox to start trusting this site's certificate. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site does not have trusted identification.

**Add Exception...**

- \_\_ e. On the Add Security Exception window, click **Confirm Security Exception** to add the certificate to the list of certificates that the browser trusts.
- \_\_ f. The console login window is displayed.
- \_\_ g. Return to the Firefox preferences by clicking **Edit > Preferences**.
- \_\_ h. Select the **Advanced** icon and the **Encryption** tab.

- \_\_\_ i. Click **View Certificates** and the **Servers** tab. Scroll down and look for the certificates named **was85host** under IBM. Notice that there are now two exceptions listed (one for localhost and one for was85host).



- \_\_\_ j. Close all of the open dialog boxes.

**End of exercise**

## Exercise review and wrap-up

This exercise looked at the installation of WebSphere Application Server V8.5. The Profile Management Tool was used to create an application server profile called profile1.

# Exercise 3. Installing IBM HTTP Server

## What this exercise is about

This exercise covers the installation and configuration of IBM HTTP Server and its WebSphere Application Server plug-ins.

## What you should be able to do

At the end of the exercise, you should be able to:

- Install IBM HTTP Server with IBM Installation Manager
- Confirm and test the installation of IBM HTTP Server
- Install Web Server Plug-ins for WebSphere Application Server
- Install WebSphere Customization Toolbox
- Configure the Web Server Plug-ins for WebSphere Application Server
- Examine the installed directories and files for IBM HTTP Server and the plug-ins

## Introduction

A main theme in this release is the separation of installation from configuration. IBM Installation Manager is the installation technology of choice. Installation Manager is a program that helps you install, import, update, modify, and uninstall packages on your computer. Installation Manager also provides tools for managing licenses for the packages that it installs, and for updating and modifying packages. If Installation Manager is installed on your computer, it is updated to ensure that the computer has the latest version installed.

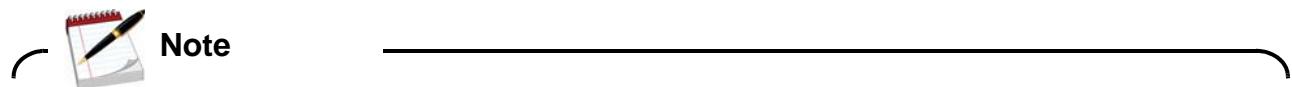
In this exercise, you use IBM Installation Manager to install IBM HTTP Server and WebSphere Application Server plug-ins. The plug-ins are configured with the Web Server Plug-ins Configuration Tool. In WebSphere Application Server V8, installation and configuration are separated into two different steps. The HTTP Server and the Plug-ins installers do not offer the configuration steps that existed in v7.0. Instead, administrators use the Web Server Plug-ins Configuration Tool to complete a plug-in configuration for various web server plug-ins. Administrators can also configure IBM HTTP Server Admin server.

## Requirements

To do this exercise, you must complete the Installing IBM Installation Manager and Installing WebSphere Application Server exercises.

## Exercise instructions

### Section 1: Resetting the WebSphere environment



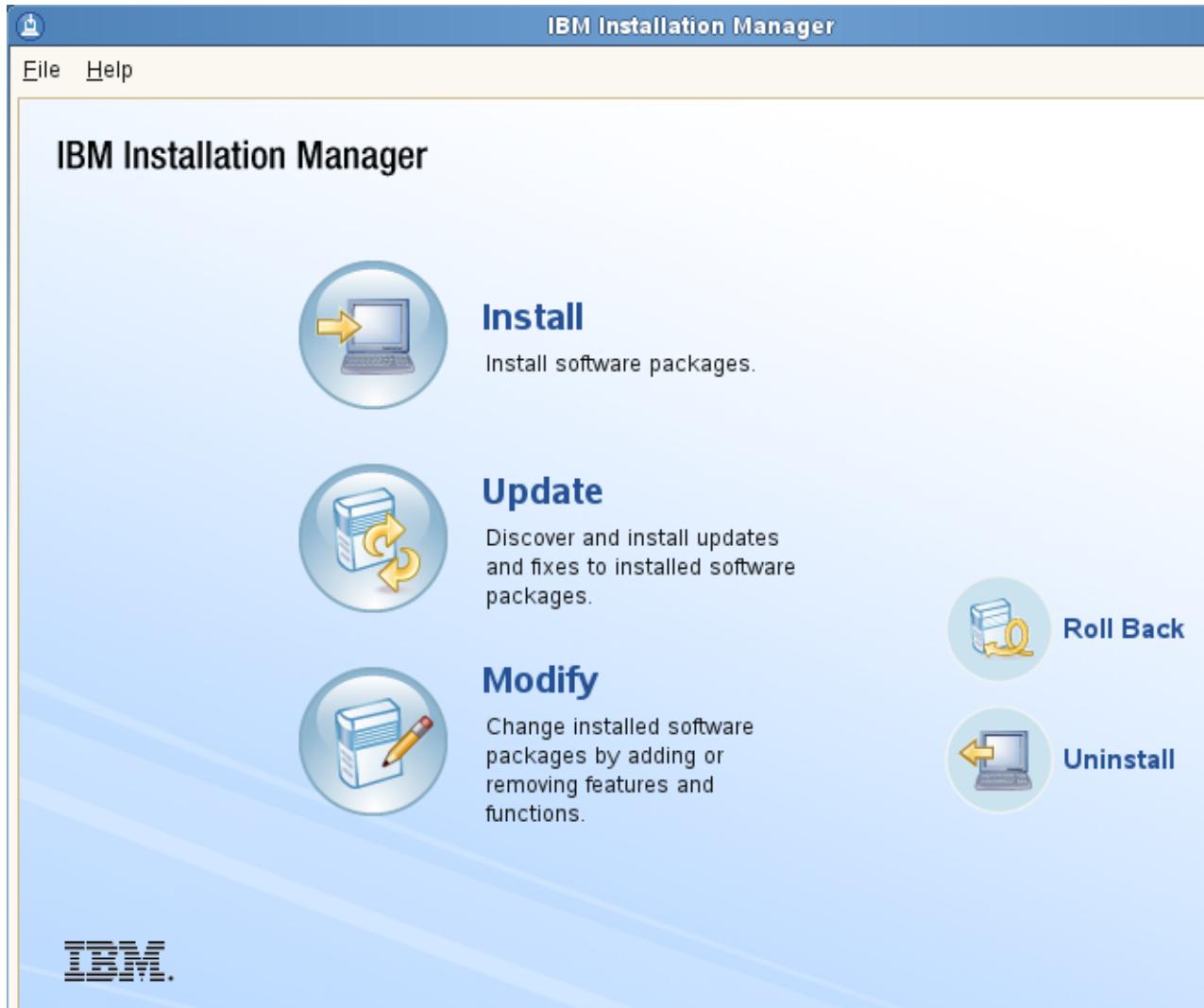
#### Note

To reset your WebSphere environment, read **Appendix A** for instructions on how to complete this procedure.

### Section 2: Installation of the IBM HTTP Server

- \_\_\_ 1. Start IBM Installation Manager.
  - \_\_\_ a. Open a terminal window and navigate to the following directory:  
`/opt/IBM/InstallationManager/eclipse/`
  - \_\_\_ b. Enter `./IBMMIM` to start IBM Installation Manager.

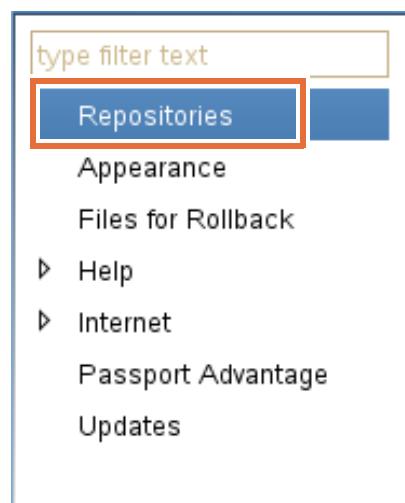
- \_\_ c. Installation Manager opens to the main page.



- \_\_ 2. Configure a repository for the IBM HTTP Server package files.

- \_\_ a. Click **File > Preferences**.

\_\_ b. Select **Repositories**.

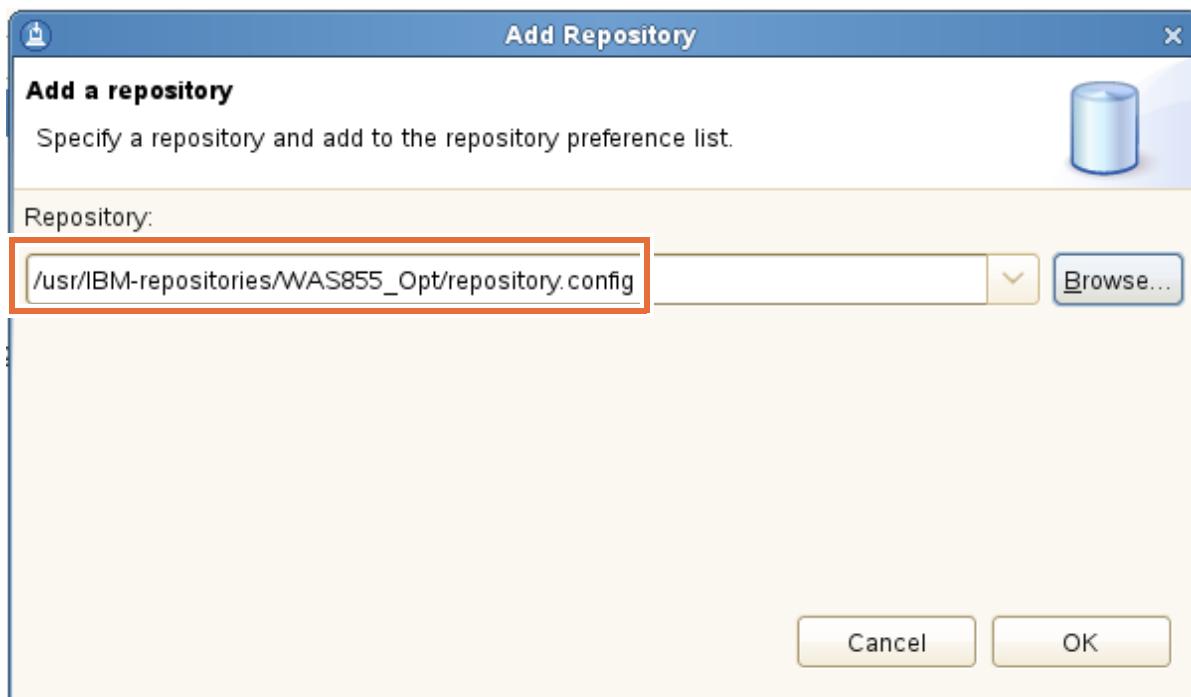


\_\_ c. Click **Add Repository**.



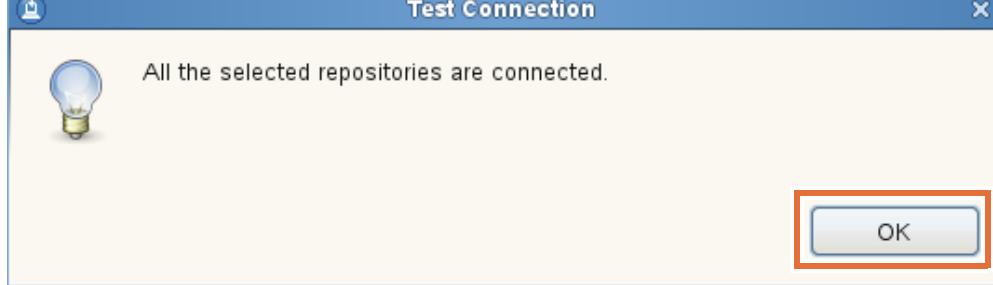
- \_\_ d. Click **Browse** and select the following file:

/usr/IBM-repositories/WAS855\_Opt/repository.config



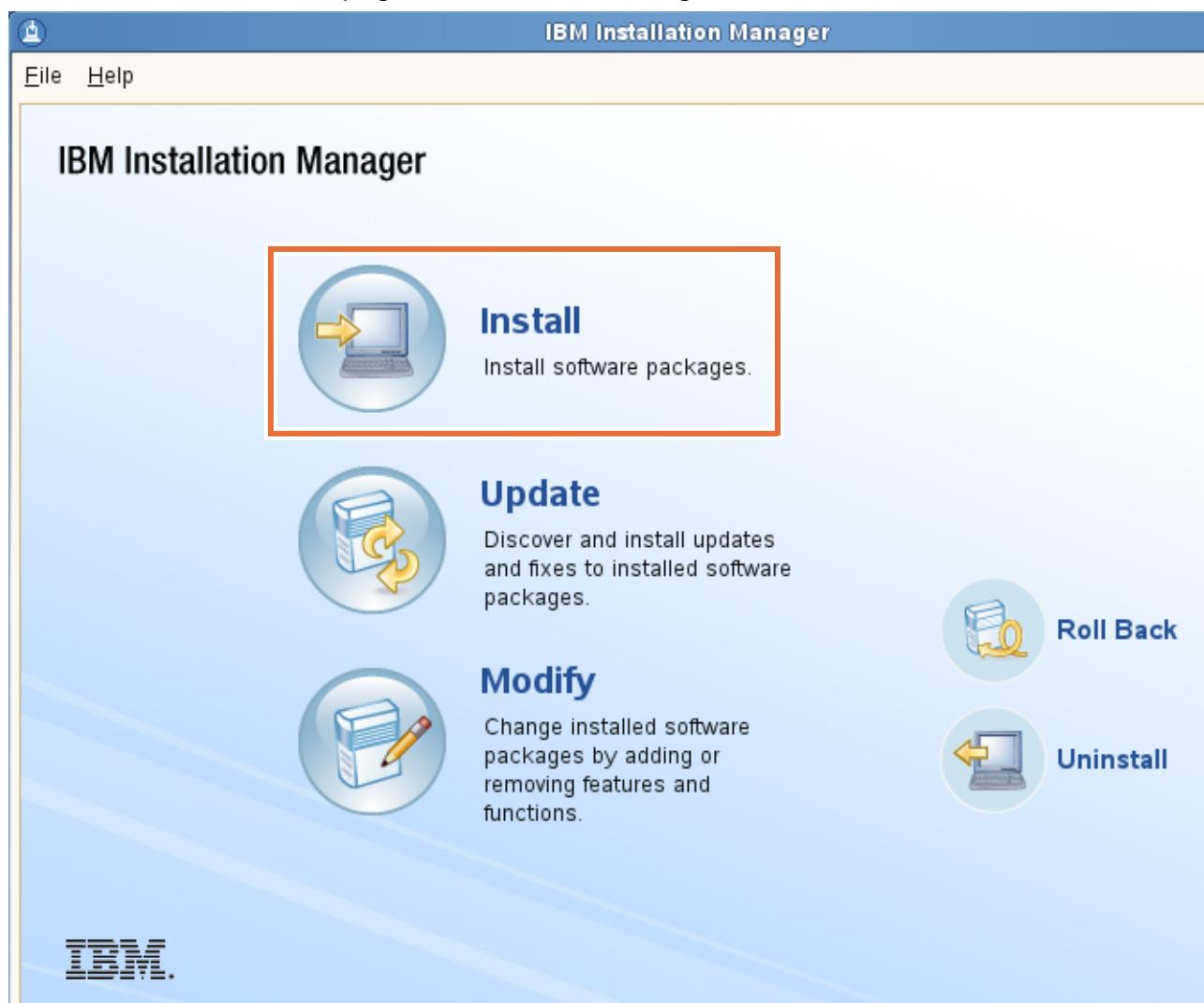
- \_\_ e. Click **OK** to add the repository to the list of repositories.

- \_\_ 3. Test the connection to the local repository. This step is more important when you configure remote repositories and need access to the repository to complete a product installation.
- \_\_ a. Click **Test Connections**.
  - \_\_ b. A message indicates that the repository is connected. Click **OK**.



- \_\_ c. Click **OK** to close the Preferences window.

- 4. Start the installation of IBM HTTP Server packages.
  - a. From the main page of Installation Manager, click **Install**.



- \_\_ b. From the list of installation packages, select only **IBM HTTP Server for WebSphere Application Server**, which also selects Version 8.5.5.0.

| Installation Packages                                                                      | Status            |
|--------------------------------------------------------------------------------------------|-------------------|
| Application Client for IBM WebSphere Application Server                                    |                   |
| <input type="checkbox"/> Version 8.5.5.0                                                   |                   |
| IBM HTTP Server for WebSphere Application Server                                           | Will be installed |
| <input checked="" type="checkbox"/> Version 8.5.5.0                                        |                   |
| IBM WebSphere Application Server Network Deployment                                        | Installed         |
| <input type="checkbox"/> Version 8.5.5.0                                                   | Installed         |
| <input type="checkbox"/> Pluggable Application Client for IBM WebSphere Application Server |                   |
| Web Server Plug-ins for IBM WebSphere Application Server                                   |                   |
| <input type="checkbox"/> Version 8.5.5.0                                                   |                   |
| WebSphere Customization Toolbox                                                            |                   |
| <input type="checkbox"/> Version 8.5.5.0                                                   |                   |

- \_\_ c. Click **Next**. This action retrieves and validates the package files from the repository.  
\_\_ d. Select **I accept the terms in the license agreement** and click **Next**.

- \_\_ e. On the Location panel, accept the default of **Create a new package group** and the default Installation Directory path of /opt/IBM/HTTPServer.

| Package Group Name   | Installation Directory |
|----------------------|------------------------|
| IBM HTTP Server V8.5 | /opt/IBM/HTTPServer    |

Package Group Name: IBM HTTP Server V8.5  
Installation Directory: /opt/IBM/HTTPServer

| Disk Space Information |                 |
|------------------------|-----------------|
| Volume                 | Available Space |
| /                      | 18.29 GB        |

- \_\_ f. Click **Next**.
- \_\_ g. On the next panel, accept the default selection **IBM HTTP Server for WebSphere Application Server 8.5.5.0** as the feature to install.

- \_\_ h. Click **Next**.

- \_\_ i. On the Features panel for Configuration for IBM HTTP Server for WebSphere Application Server 8.5.5.0, enter 80 as the HTTP port.

### Install Packages

Fill in the configurations for the packages.

The screenshot shows the 'Configuration for IBM HTTP Server for WebSphere' interface. The 'Features' tab is active. On the left, there's a tree view with 'IBM HTTP Server for WebSph' expanded, and 'Web Server Configuration' is selected. The main pane is titled 'Configuration for IBM HTTP Server for WebSphere' and 'Web Server Configuration'. It contains a note about specifying a port number for the server. Below the note is an input field labeled 'HTTP port:' with the value '80' entered. A red box highlights this input field.

- \_\_ j. Click **Next**.
- \_\_ k. On the Summary panel, verify that IBM HTTP Server for WebSphere Application Server 8.5.5.0 is the package for installation. Click **Install**.

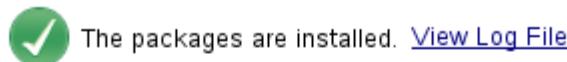
### Install Packages

Review the summary information.



The screenshot shows the 'Summary' panel of the 'Install Packages' wizard. At the top, there's a navigation bar with tabs: 'Install', 'Licenses', 'Location', 'Features', 'Summary' (which is active), and 'Finish'. Below the tabs, there's a section for 'Target Location' with fields for 'Package Group Name' (IBM HTTP Server V8.5), 'Installation Directory' (/opt/IBM/HTTPServer), and 'Shared Resources Directory' (/opt/IBM/IMShared). Under 'Packages', there's a list of packages with 'IBM HTTP Server for WebSphere Application Server 8.5.5.0' selected. In the 'Environment' section, it shows 'English'. In 'Disk Space Information', it shows 'Total Available Space' as 23.27 GB. Under 'Repository Information', it shows 'Total Download Size: 31.49 MB' and 'Total Installation Size: 209.95 MB'. At the bottom, there are buttons for '?', '< Back' (disabled), 'Next >', 'Install' (highlighted with a red box), and 'Cancel'.

- \_\_ I. When the installation of the packages is complete, a message is shown.



The following package was installed:

|   |                                                          |
|---|----------------------------------------------------------|
| ▼ | IBM HTTP Server V8.5                                     |
|   | IBM HTTP Server for WebSphere Application Server 8.5.5.0 |

- \_\_ m. Click **Finish**.

### Section 3: Install IBM HTTP Server plug-in

The plug-ins are necessary for routing requests from the web server and WebSphere Application Server.

- \_\_ 1. Start the installation of the Web Server Plug-ins for IBM WebSphere Application Server.
- \_\_ a. From the main page of Installation Manager, click **Install**.
- \_\_ b. From the list of installation packages, select only **Web Server Plug-ins for IBM WebSphere Application Server**. This action also selects Version 8.5.5.0.

| Installation Packages |                                                                                              | Status            |
|-----------------------|----------------------------------------------------------------------------------------------|-------------------|
| ▼                     | <input type="checkbox"/> Application Client for IBM WebSphere Application Server             |                   |
|                       | <input type="checkbox"/> Version 8.5.5.0                                                     |                   |
| ▼                     | <input type="checkbox"/> IBM HTTP Server for WebSphere Application Server                    | Installed         |
|                       | <input type="checkbox"/> Version 8.5.5.0                                                     | Installed         |
| ▼                     | <input type="checkbox"/> IBM WebSphere Application Server Network Deployment                 | Installed         |
|                       | <input type="checkbox"/> Version 8.5.5.0                                                     | Installed         |
|                       | <input type="checkbox"/> Pluggable Application Client for IBM WebSphere Application Server   |                   |
| ▼                     | <input checked="" type="checkbox"/> Web Server Plug-ins for IBM WebSphere Application Server |                   |
|                       | <input checked="" type="checkbox"/> Version 8.5.5.0                                          | Will be installed |
| ▼                     | <input type="checkbox"/> WebSphere Customization Toolbox                                     |                   |
|                       | <input type="checkbox"/> Version 8.5.5.0                                                     |                   |

- \_\_ c. Click **Next**. This action retrieves and validates the package files from the repository.

- \_\_ d. Select **I accept the terms in the license agreement** and click **Next**.
- \_\_ e. Select **Create a new package group** and accept the default Installation Directory of /opt/IBM/WebSphere/Plugins. Click **Next**.
- \_\_ f. On the Features panel, select **Web Server Plug-ins for WebSphere Application Server 8.5.5.0** as the feature for installation. Click **Next**.
- \_\_ g. On the Summary panel, click **Install**.
- \_\_ h. When the installation of the packages is complete, a message is shown.



The packages are installed. [View Log File](#)

The following package was installed:



- \_\_ i. Click **Finish**.

## Section 4: Install WebSphere Customization Toolbox

The WebSphere Customization Toolbox includes tools for managing, configuring, and upgrading various parts of the WebSphere Application Server environment. Three tools are included:

- Profile Management Tool for z/OS only
- z/OS Migration Tool
- Web Server Plug-ins Configuration Tool

In this part of the exercise, the Web Server Plug-ins Configuration Tool (PCT) is installed as part of the WebSphere Customization Toolbox. The PCT includes both browser-based and command-line options. The other tools are not necessary for the lab exercise environment.

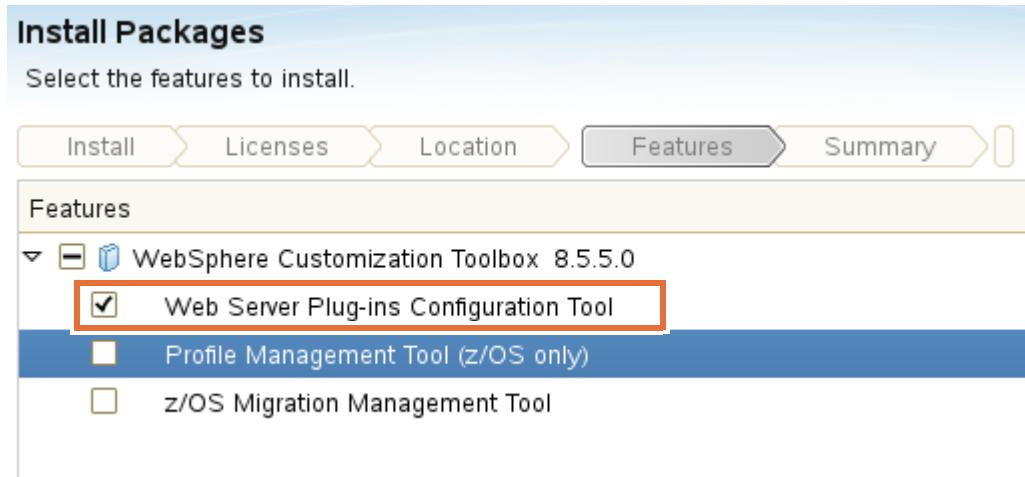
The Web Server Plug-ins Configuration Tool is needed in the next section to configure the web server plug-in. The plug-in configuration allows the web server to connect correctly with the application server.

- \_\_ 1. Install the WebSphere Customization Toolbox.
  - \_\_ a. From the main page of Installation Manager, click **Install**.

- \_\_\_ b. From the list of installation packages, select **WebSphere Customization Toolbox**. This action also selects Version 8.5.5.0.

| Installation Packages                                      | Status            | Vendor |
|------------------------------------------------------------|-------------------|--------|
| Application Client for IBM WebSphere Application Server    |                   |        |
| Version 8.5.5.0                                            |                   | IBM    |
| IBM HTTP Server for WebSphere Application Server           | Installed         |        |
| Version 8.5.5.0                                            | Installed         | IBM    |
| IBM WebSphere Application Server Network Deployment        | Installed         |        |
| Version 8.5.5.0                                            | Installed         | IBM    |
| Pluggable Application Client for IBM WebSphere Application |                   |        |
| Web Server Plug-ins for IBM WebSphere Application Server   | Installed         |        |
| Version 8.5.5.0                                            | Installed         | IBM    |
| <b>WebSphere Customization Toolbox</b>                     |                   |        |
| Version 8.5.5.0                                            | Will be installed | IBM    |

- \_\_\_ c. Click **Next**. This action retrieves and validates the package files from the repository.
- \_\_\_ d. Select **I accept the terms in the license agreement** and click **Next**.
- \_\_\_ e. Accept `/opt/IBM/WebSphere/Toolbox` as the Installation Directory and click **Next**.
- \_\_\_ f. On the Features panel, select **Web Server Plug-ins Configuration Tool**. Clear any other check boxes and click **Next**.



- \_\_ g. On the Summary panel, verify that the only package that is listed under the **WebSphere Customization Toolbox 8.5.5.0** is the Web Server Plug-ins Configuration Tool.

## Install Packages

Review the summary information.

Install    Licenses    Location    Features    **Summary**

**Target Location**

Package Group Name: WebSphere Customization Toolbox V8.5  
 Installation Directory: /opt/IBM/WebSphere/Toolbox  
 Shared Resources Directory: /opt/IBM/IMShared

| <b>Packages</b> |                                         |
|-----------------|-----------------------------------------|
| Packages        |                                         |
| ▼               | WebSphere Customization Toolbox 8.5.5.0 |
| ▼               | Web Server Plug-ins Configuration Tool  |

**Environment**

English

**Disk Space Information**

/

Total Download Size: 53.53 MB  
 Total Installation Size: 251.49 MB

\_\_ h. Click **Install**.

\_\_ i. When the installation of the packages is complete, select **None** under “Which program do you want to start?”

The following package was installed:

- ▼ WebSphere Customization Toolbox V8.5
  - WebSphere Customization Toolbox 8.5.5.0

Which program do you want to start?

WebSphere Customization Toolbox  
 None

- \_\_ j. Click **Finish**.
- \_\_ k. Close IBM Installation Manager.

## Section 5: Configuring the web server plug-in

After the products are installed with the IBM Installation Manager, other tools are used to complete the configuration processes. The Web Server Plug-ins Configuration Tool is available for configuring web server plug-ins. The Web Server Plug-ins Configuration Tool creates one or more configurations for the web server plug-ins that can direct requests from a web client through the web server. The tool edits the configuration files for a web server by creating directives that point to the location of the binary plug-in module. The tool also creates a directive to the plug-in configuration file.

In this part of the exercise, the Web Server Plug-ins Configuration Tool (PCT) is used to configure the web server plug-in.

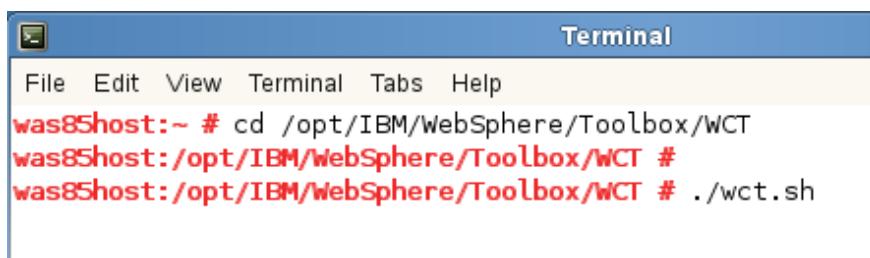
— 1. Start the WebSphere Customization Toolbox.

— a. Using a terminal window, navigate to the following directory:

```
/opt/IBM/WebSphere/Toolbox/WCT
```

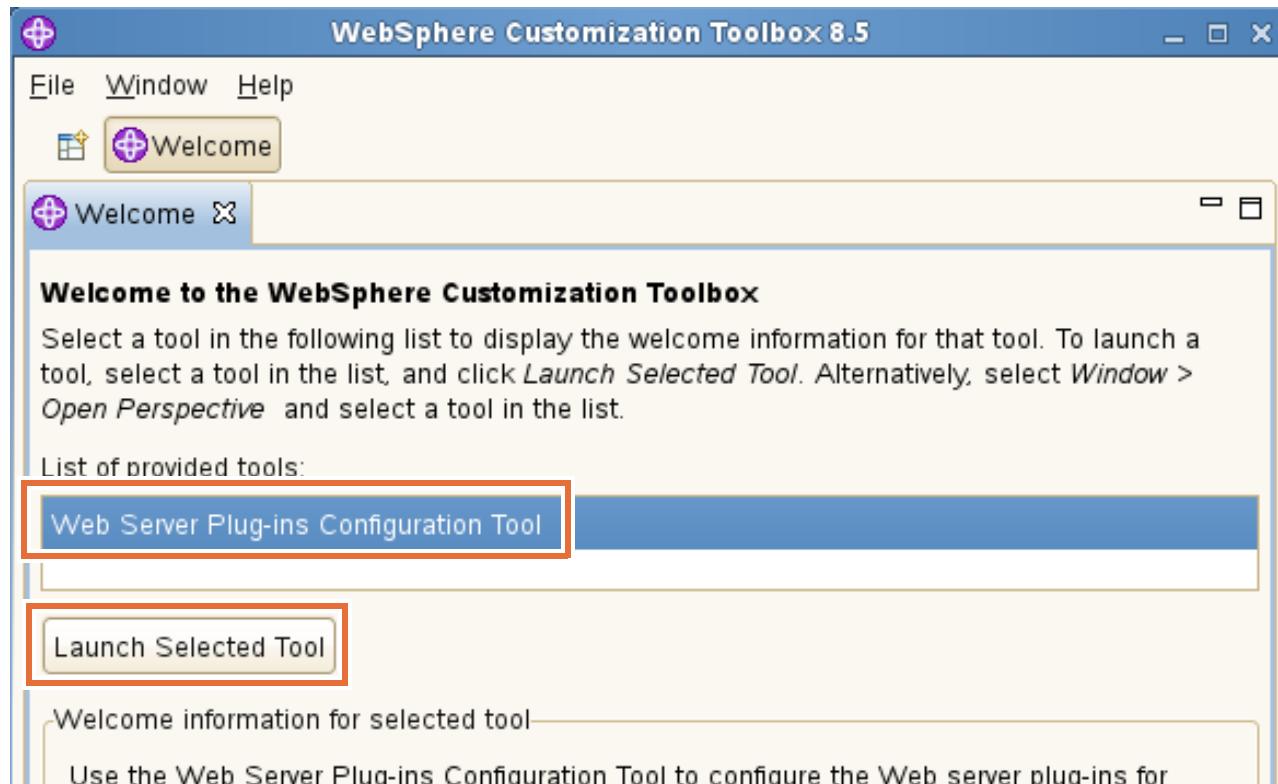
— b. Enter the following command to start the WebSphere Customization Toolbox:

```
./wct.sh
```

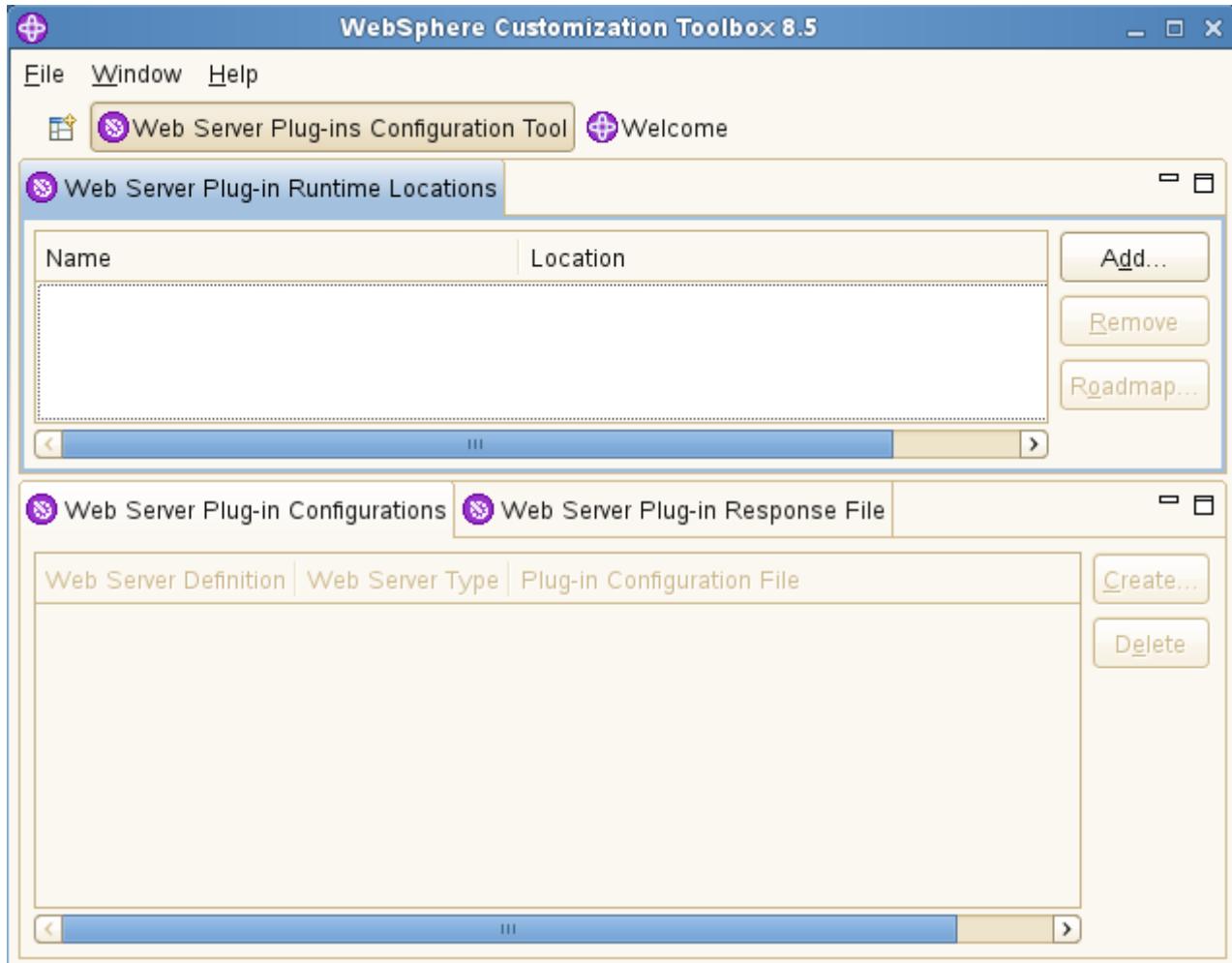


```
Terminal
File Edit View Terminal Tabs Help
was85host:~ # cd /opt/IBM/WebSphere/Toolbox/WCT
was85host:/opt/IBM/WebSphere/Toolbox/WCT #
was85host:/opt/IBM/WebSphere/Toolbox/WCT # ./wct.sh
```

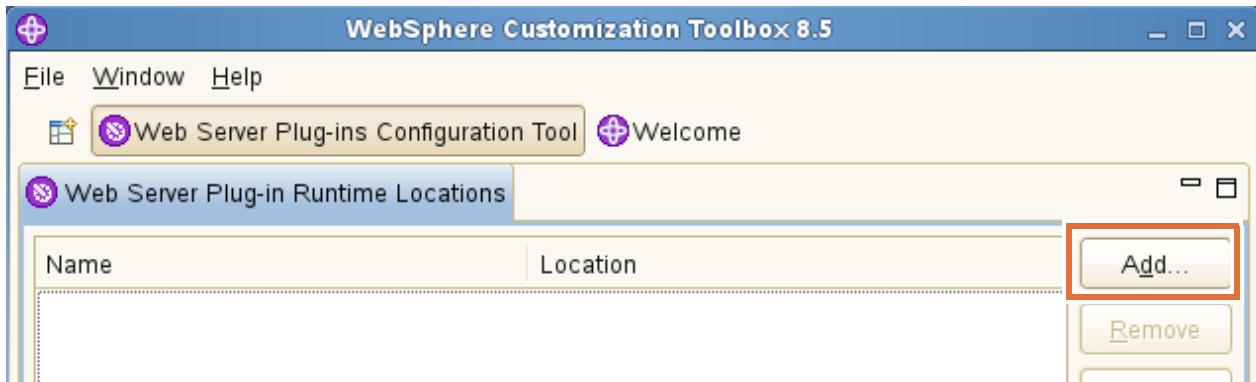
- \_\_ c. Select **Web Server Plug-ins Configuration Tool** and click **Launch Selected Tool**.



- \_\_\_ d. The WebSphere Customization Toolbox opens and looks like the following screen capture:

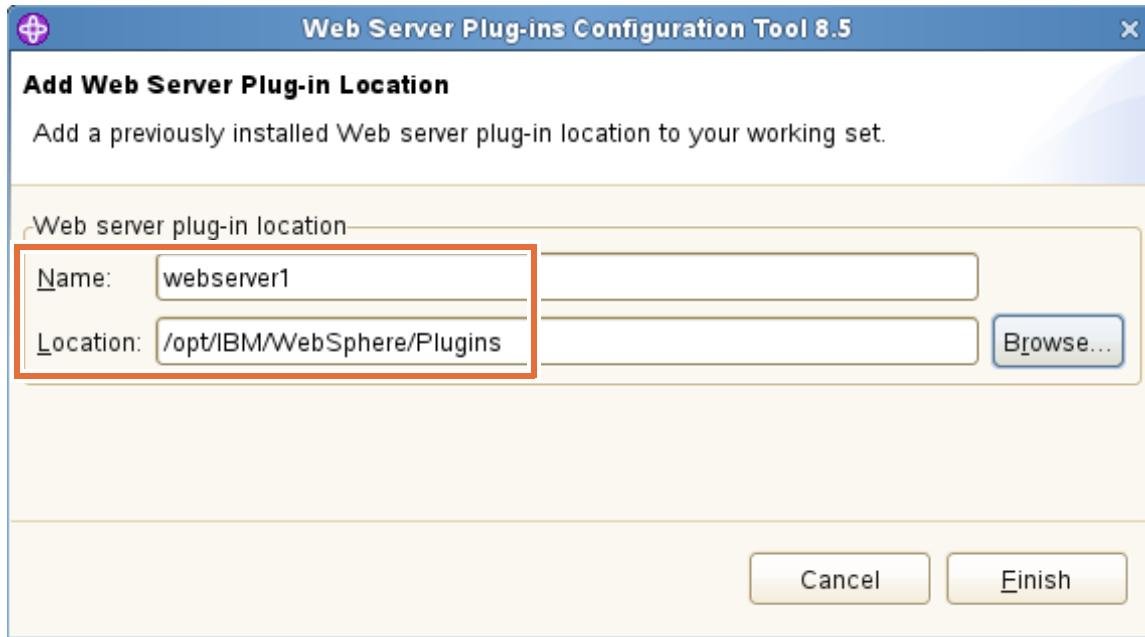


- \_\_\_ 2. Add the location of the web server plug-in runtime files. Since the installed web server plug-in that you want to use is not in the list, you must add it.
- \_\_\_ a. In the Web Server Plug-in Runtime Locations panel, click **Add**.



\_\_ b. In the Add Web Server Plug-in Location panel, enter the following values:

- In the **Name** field, enter: webserver1
- In the **Location** field, **Browse** to: /opt/IBM/WebSphere/Plugins

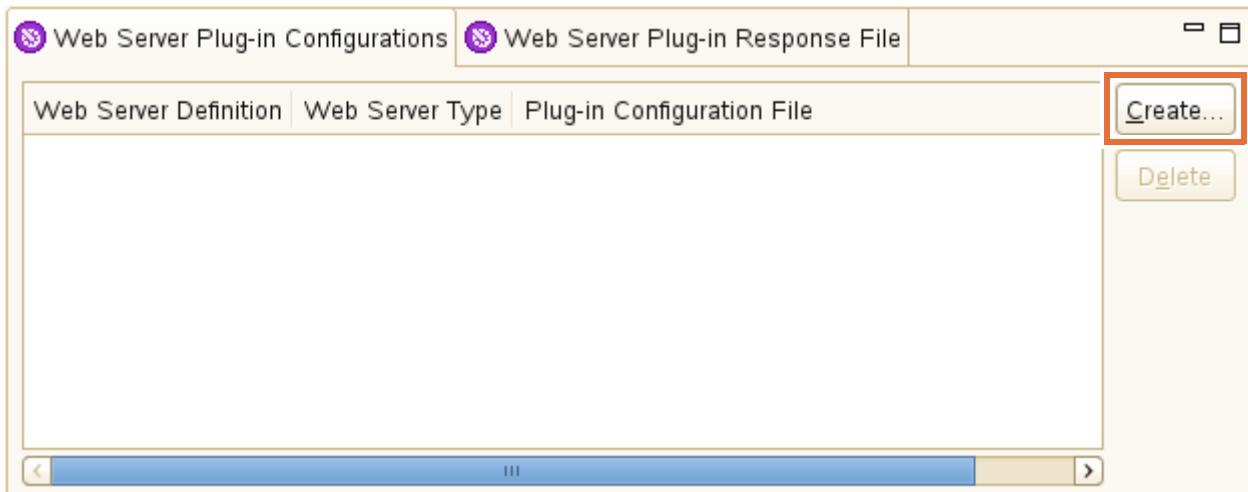


\_\_ c. Verify the settings and click **Finish**.

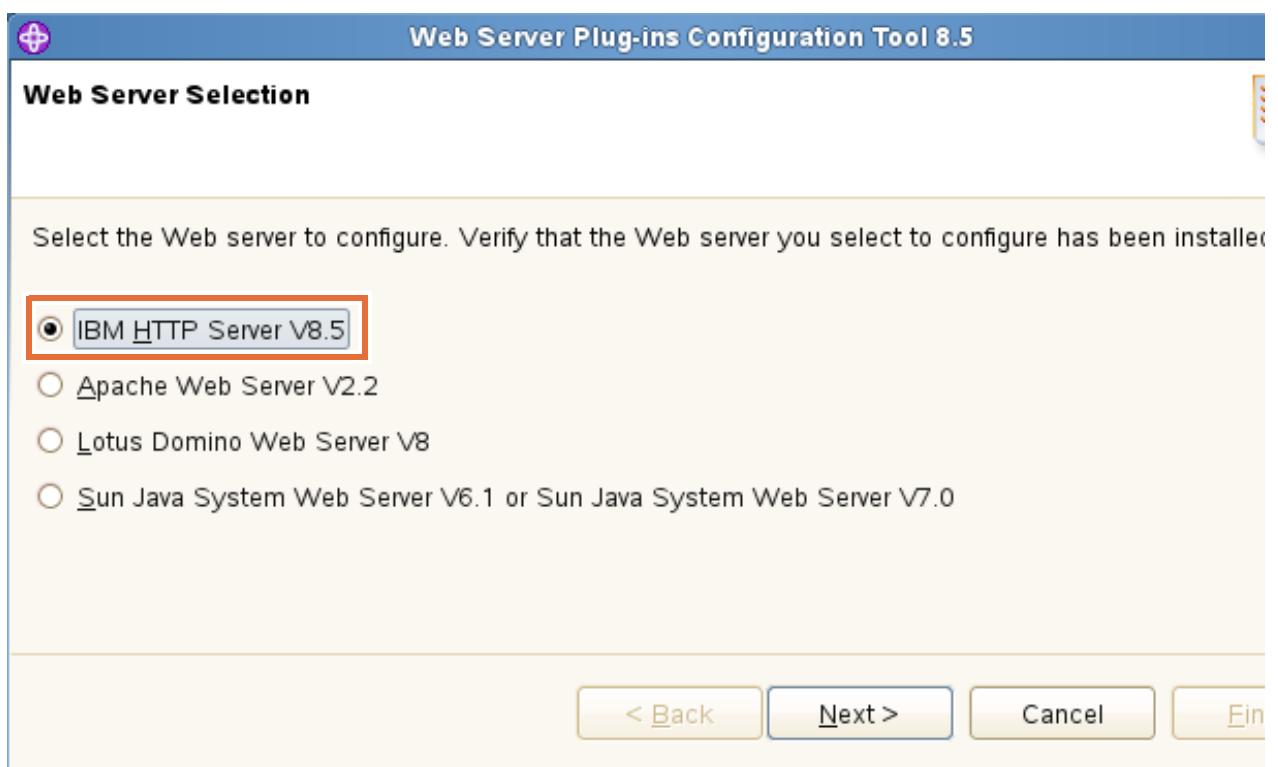
\_\_ d. The web server plug-in runtime is added to the location list.

| Name       | Location                   | Add...                 |
|------------|----------------------------|------------------------|
| webserver1 | /opt/IBM/WebSphere/Plugins | <a href="#">Remove</a> |

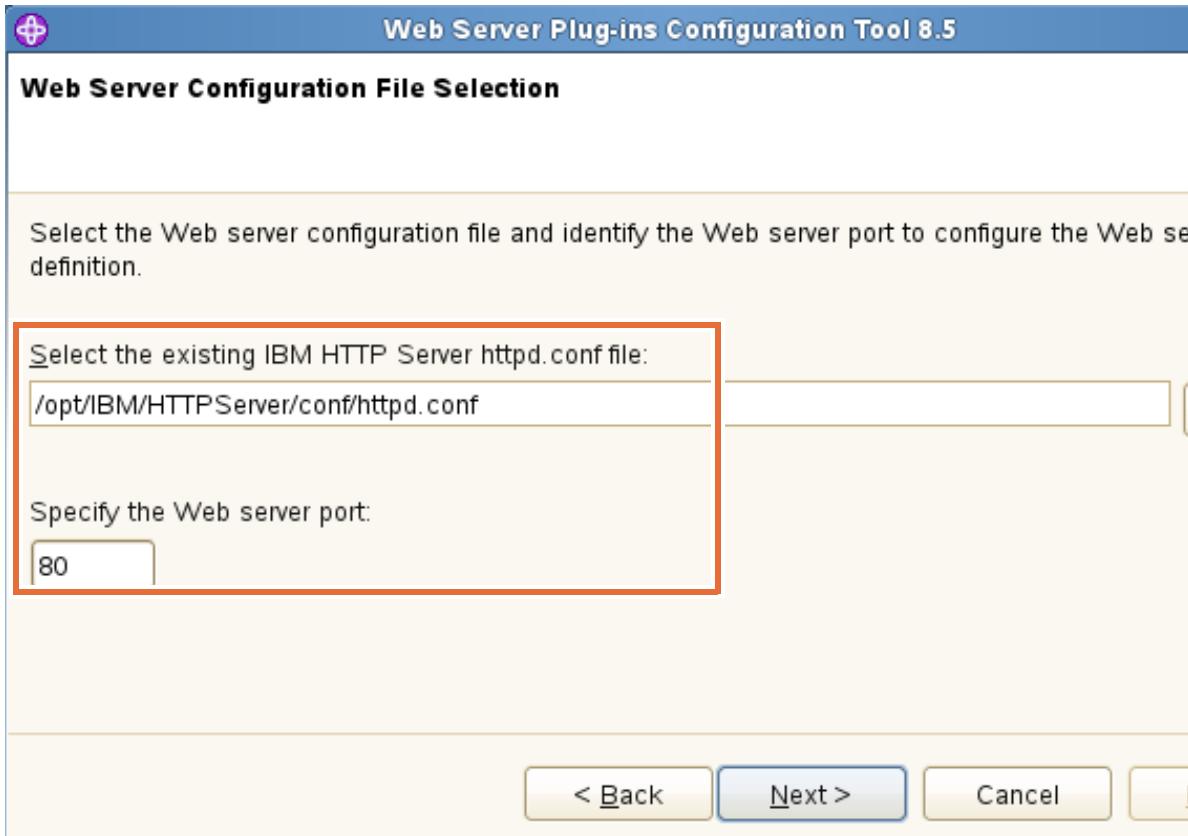
\_\_ e. In the Web Server Plug-in Configurations panel, click **Create**.



\_\_ f. Verify that **IBM HTTP Server V8.5** is selected and click **Next**.

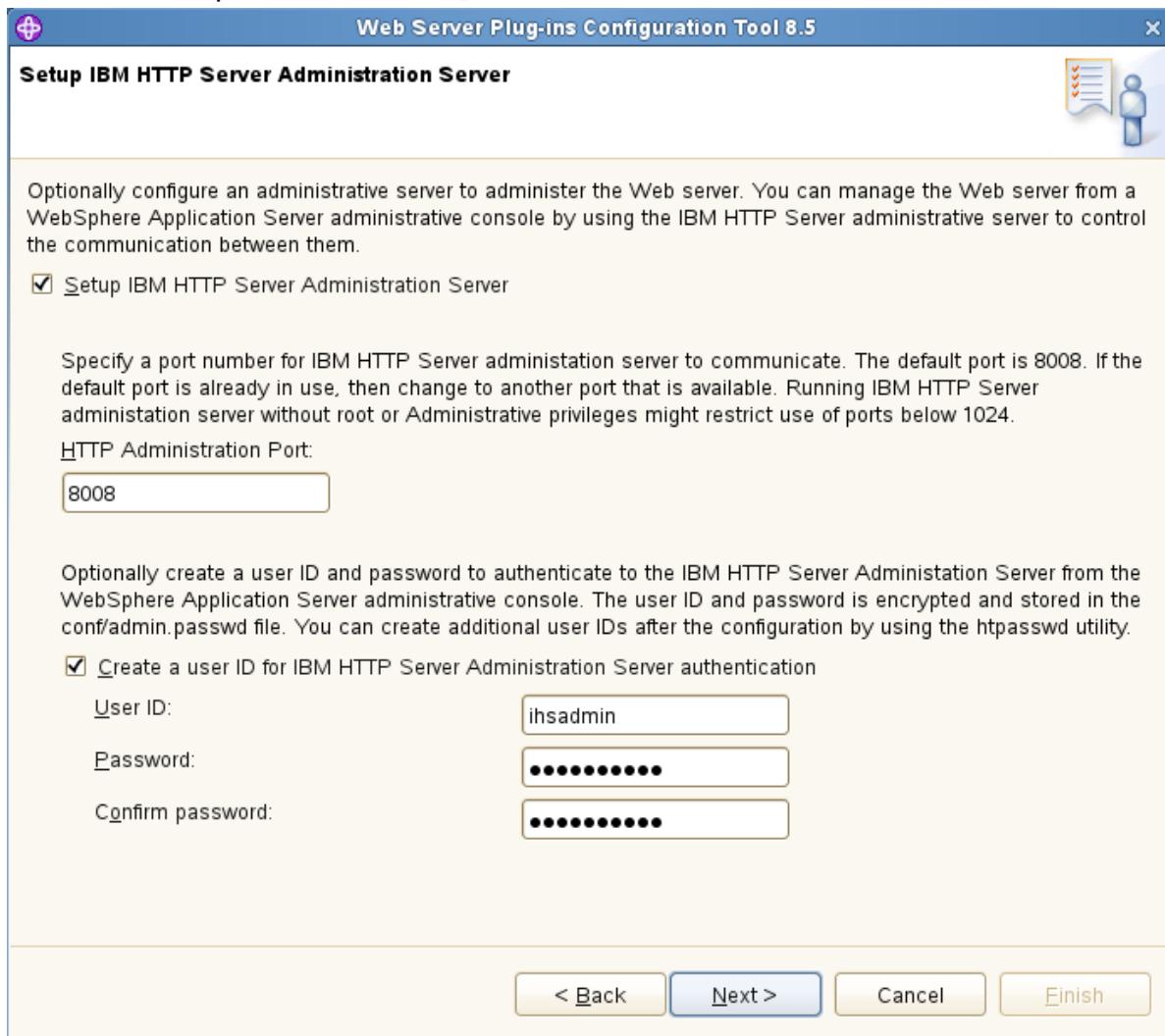


- g. In the Web Server Configuration File Selection panel, accept the following default values:
- Select the existing IBM HTTP Server httpd.conf file:  
/opt/IBM/HTTPServer/conf/httpd.conf
  - Specify the Web server port: 80



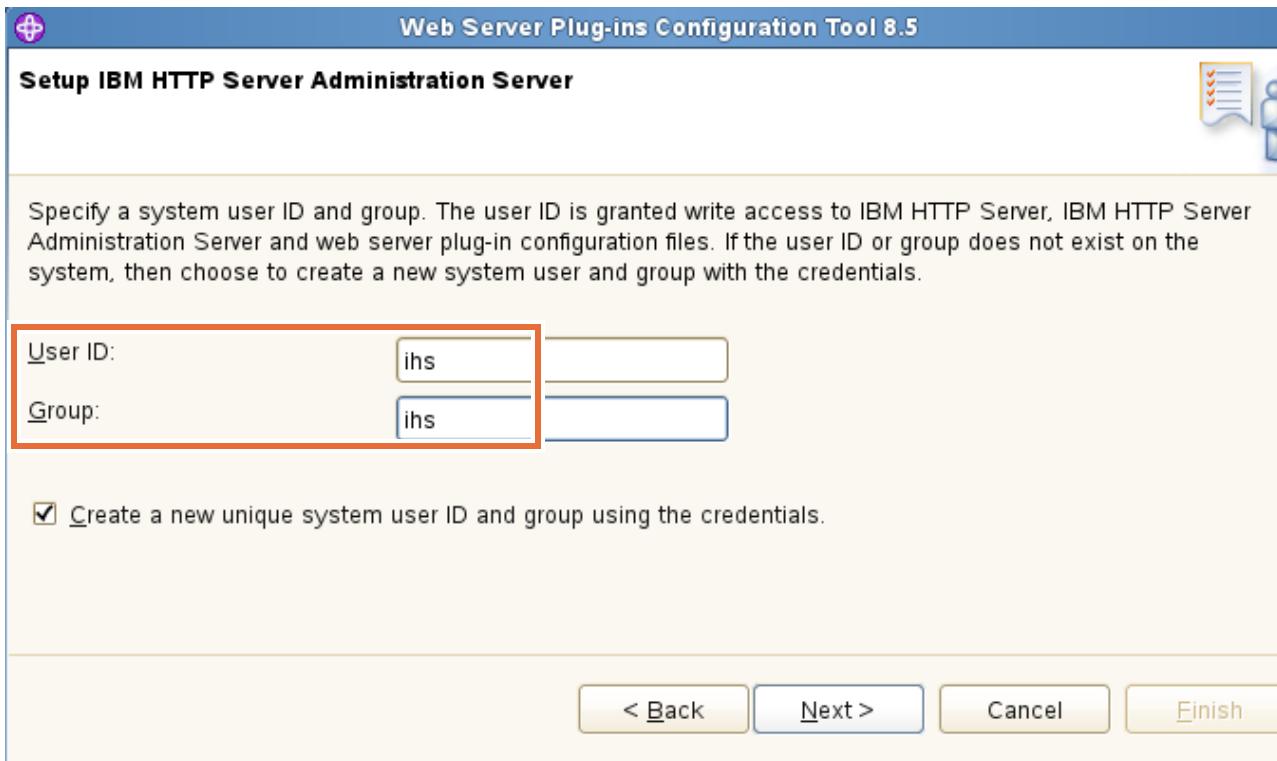
- h. Click **Next**.

- \_\_ i. In the Setup IBM HTTP Server Administration Server panel, enter the following values (note: expand the dialog box to see all the fields):
- Verify that the **Setup IBM HTTP Server Administration Server** check box is selected.
  - HTTP Administration Port: 8008
  - Verify that the **Create a user ID for IBM HTTP Administration Server authentication** check box is selected.
  - User ID: ihsadmin
  - Password: web1sphere
  - Confirm password: web1sphere



- \_\_ j. Click **Next**.

- \_\_ k. On the Setup IBM HTTP Server Administration Server panel, enter `ihs` in the **User ID** and **Group** fields.



- \_\_ l. Click **Next**.

- \_\_\_ m. In the Web Server Definition Name panel, verify that the unique name for the web server is `webserver1` and click **Next**.

**Web Server Definition Name**

Use a Web server definition to manage a Web server through the WebSphere Application Server administrative console or the wsadmin tool. The definition name must be unique because this name is used to identify this V server in the administrative console.

Specify a unique Web server definition name:

`webserver1`

The Web server definition name must not be empty and it must not contain the following special characters or space:

< Back    Next >    Cancel    Finish

- \_\_\_ n. On the Configuration Scenario Selection panel, select the **Remote** option and enter the host name: `was85host`

**Configuration Scenario Selection**

Configure the Web server plug-ins to the computer where the Web server exists. When the Web server and application server are not on the same computer, choose the remote configuration scenario. When both Web and application server are on the same computer, choose the local configuration scenario. In the local scenario Web server definition you create in this wizard is defined automatically in the application server.

Configuration scenario

(Remote) Host name or IP address of the application server  
was85host

(Local) Installation location of WebSphere Application Server

Browse

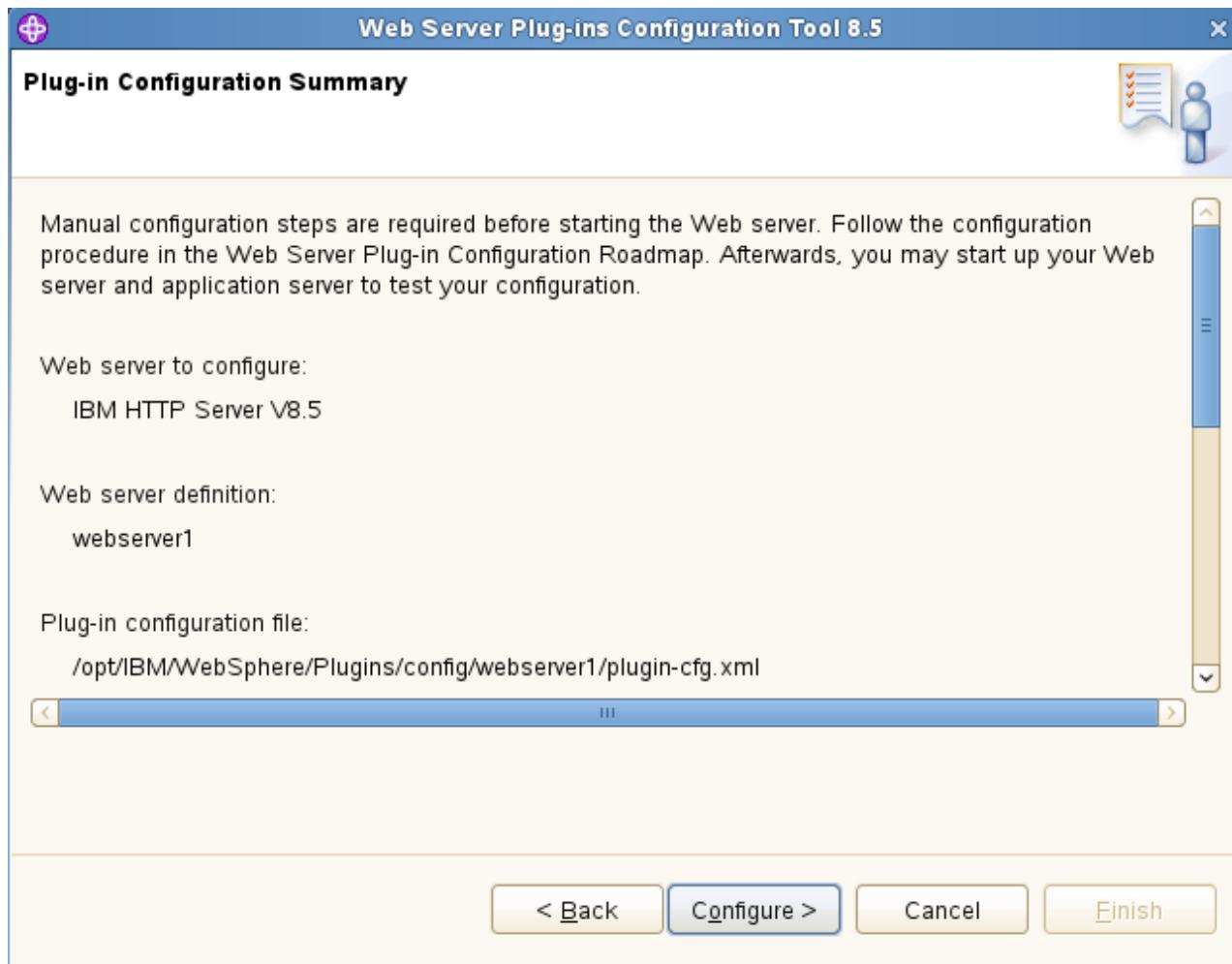
- \_\_\_ o. Click **Next** to continue.

**Note**

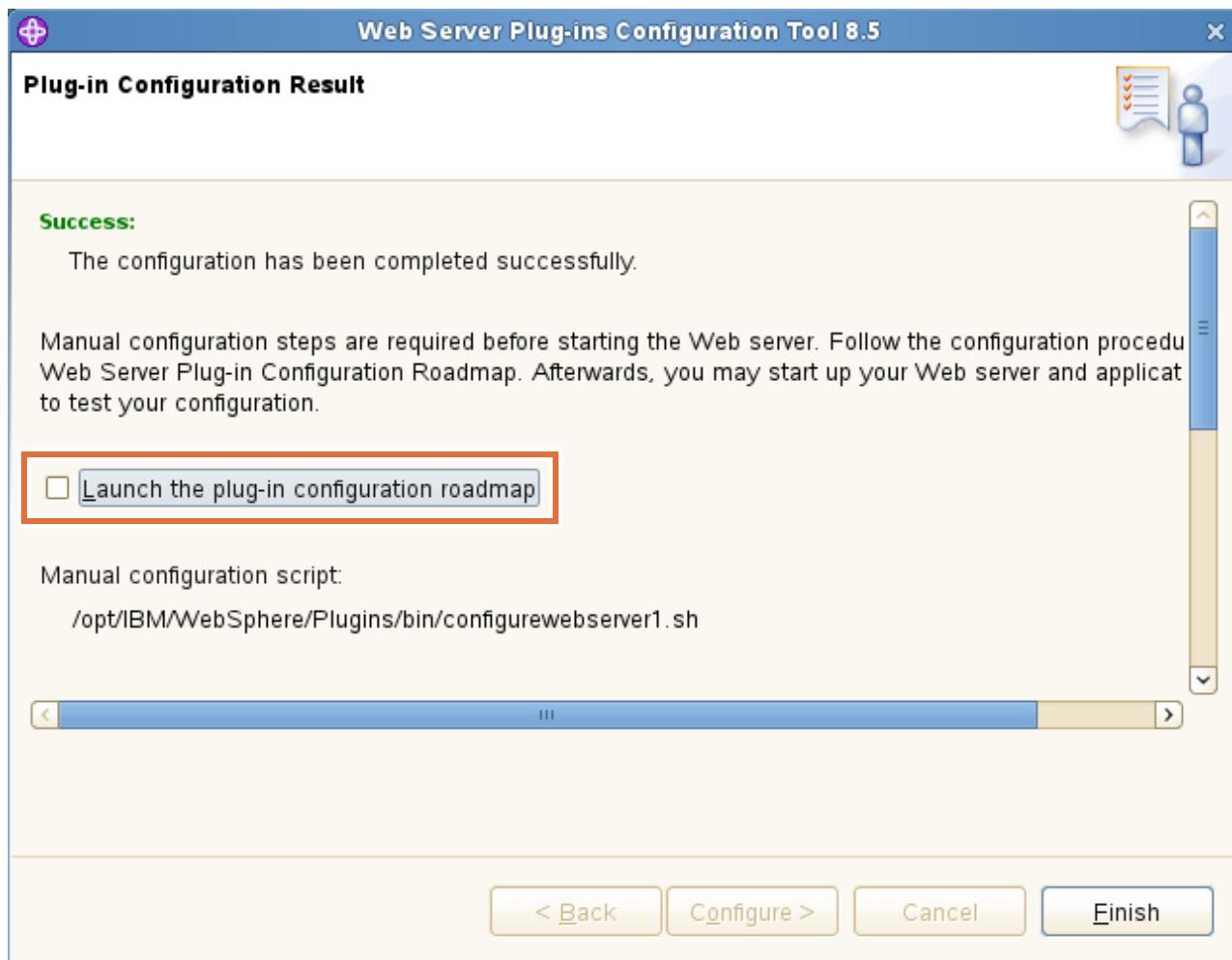
For a **remote** configuration, the Web Server Plug-ins Configuration Tool configures the web server to use the `plugin-cfg.xml` file that is maintained on the web server machine. The file is in the `plugins_root/config/web_server_name` directory. This file requires periodic propagation. Propagation means copying the current `plugin-cfg.xml` file from the application server machine to replace the `plugins_root/config/web_server_name/plugin-cfg.xml` file.

For a **local** configuration, the Web Server Plug-ins Configuration Tool configures the web server to use the `plugin-cfg.xml` file that is within the application server profile. The stand-alone application server regenerates the `profile_root/config/cells/cell_name/nodes/web_server_name_node/servers/web_server_name/plugin-cfg.xml` file whenever a change occurs in the application server configuration that affects deployed applications.

- p. On the Plug-in Configuration Summary panel, review your settings. Click **Configure**.

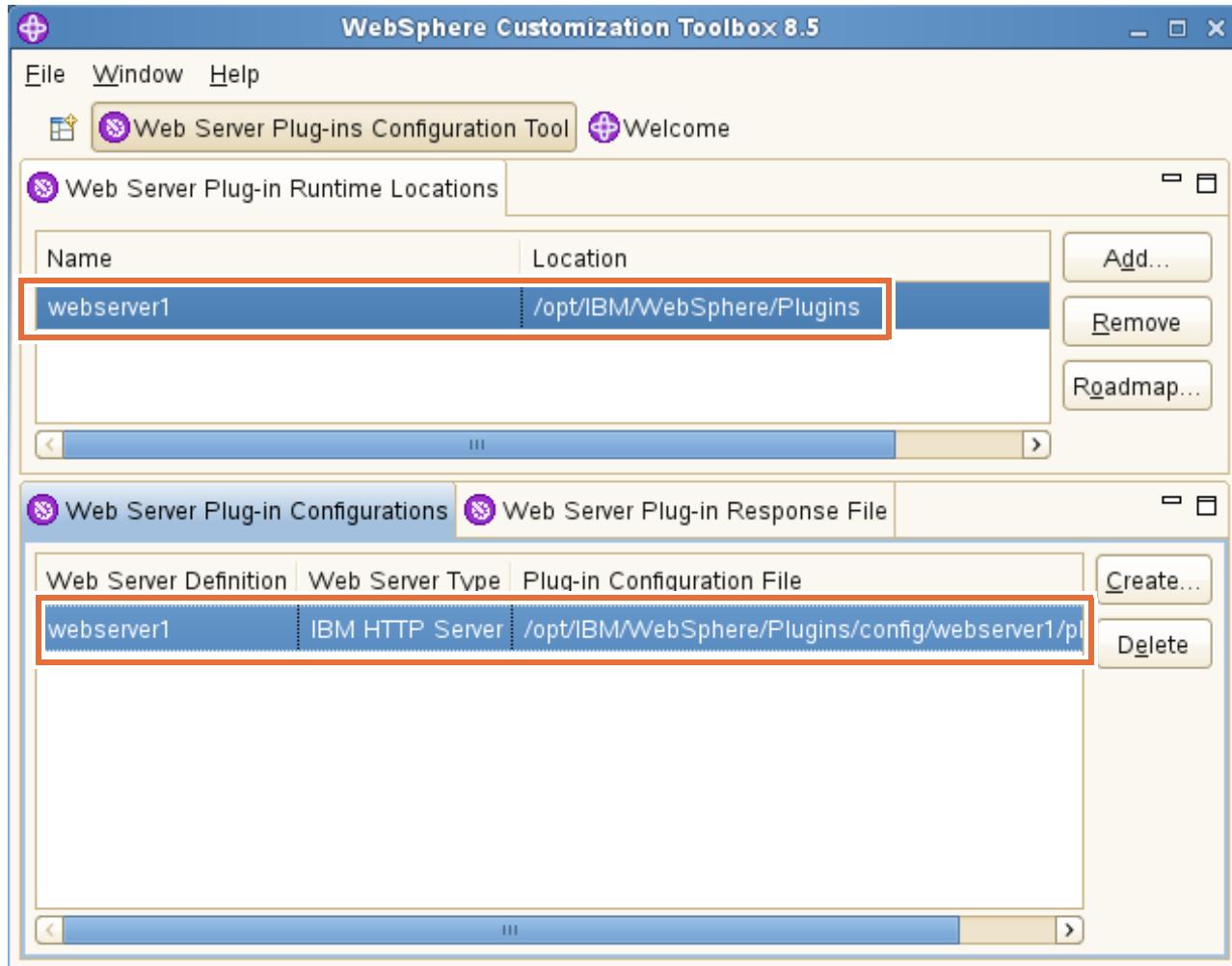


- \_\_\_ q. When the configuration is complete, the Plug-in Configuration Result panel displays the results. Clear the check box next to **Launch the plug-in configuration roadmap**.



- \_\_\_ r. Click **Finish**.

- \_\_ s. The configuration looks like the following screen capture:



- \_\_ t. Exit the WebSphere Customization Toolbox.

## **Section 6: Confirm and test installation of IBM HTTP Server and plug-ins**

- \_\_ 1. Verify that IBM HTTP Server is started.  
\_\_ a. From a terminal window, use the following command to check to see whether any httpd processes are running:

```
ps -ef | grep httpd
```

```
Terminal
File Edit View Terminal Tabs Help
was85host:~ : ps -ef | grep httpd
root 15524 19164 0 20:02 pts/1 00:00:00 grep httpd
was85host:~ #
```

- \_\_\_ b. If no `httpd` processes are started, start both the HTTP Server and the HTTP Administrative Server with the following commands:

```
/opt/IBM/HTTPServer/bin/apachectl start
/opt/IBM/HTTPServer/bin/adminctl start
```

```
Terminal
File Edit View Terminal Tabs Help
was85host:~ # /opt/IBM/HTTPServer/bin/apachectl start
was85host:~ #
was85host:~ # /opt/IBM/HTTPServer/bin/adminctl start
/opt/IBM/HTTPServer/bin/adminctl start: admin http started
was85host:~ #
```

- \_\_\_ 2. Check the status of the IBM HTTP Server with a web browser.
- \_\_\_ a. Open a browser and enter the following URL:
- http://was85host
- \_\_\_ b. If the IBM HTTP Server is started, the default IBM HTTP Server window opens.
- \_\_\_ c. Close the browser.

## **Section 7: Explore the IBM HTTP Server and plug-in installation**

Now that IBM HTTP Server is installed, examine the directory structure and review what you installed.

- \_\_\_ 1. Explore the IBM HTTP Server directory.
- \_\_\_ a. Open a terminal window.
- \_\_\_ b. Navigate to the following directory:
- /opt/IBM/HTTPServer
- \_\_\_ c. Review some of the subdirectories and their contents:
- bin: programs, scripts, and shared libraries
  - conf: configuration files
  - logs: trace, error, and log files
  - properties: messages, versions
- \_\_\_ 2. Explore the WebSphere Application Server plug-ins directory.
- \_\_\_ a. Navigate to the following directory:
- /opt/IBM/WebSphere/Plugins
- \_\_\_ b. Review some of the subdirectories and their contents:
- bin: programs, scripts, and DLLs
  - config: configuration directory for plug-in configuration file for your web server
  - logs: configuration and web server files

- plugins: JAR files
- properties: messages, versions

## **Section 8: Check installation log files**

A number of log files are created during the installation process. It is useful to check these files to verify that the installation completed successfully.

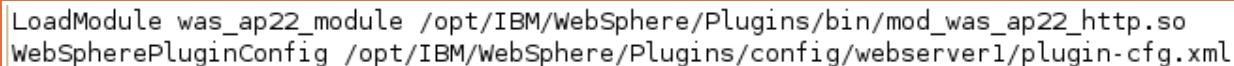
- \_\_\_ 1. Navigate to the installation logs directory under the IBM HTTP Server installation directory.
  - \_\_\_ a. Navigate to the following directory:  
`/opt/IBM/HTTPServer/logs/postinstall`
  - \_\_\_ b. Using a text editor (such as gedit, vi, or emacs), open the `postinstall.log` file. This file records installation status messages.
  - \_\_\_ c. Look for the log message “INSTCONFSUCCESS” at the end of the log to verify that the installation was successful.
  - \_\_\_ d. Close the log file.
- \_\_\_ 2. Change to the `Plugins` directory under the WebSphere directory.
  - \_\_\_ a. Navigate to the following directory:  
`/opt/IBM/WebSphere/Plugins/logs/config`
  - \_\_\_ b. Open the `installIHSPlugin.log` file. This log records WebSphere Application Server plug-ins installation status messages.
  - \_\_\_ c. Look for the log message “Install complete” to verify that the installation of the plug-ins was successful.
  - \_\_\_ d. Close the log file.

## **Section 9: Review configuration and error files**

No further configuration is required for this lab exercise. However, more configuration of the IBM HTTP Server and administration server is possible through configuration files that are in the `conf` directory of the web server.

- \_\_\_ 1. Examine the configuration directory under the IBM HTTP Server installation directory.
  - \_\_\_ a. Navigate to the following directory:  
`/opt/IBM/HTTPServer/conf`
  - \_\_\_ b. Open the `httpd.conf` file with an editor. This file contains configuration data for the IBM HTTP Server.

- \_\_\_ c. Scroll to the bottom of this file and notice the last two lines. They define the module that is loaded as the WebSphere plug-in and define the path to the plugin-cfg.xml file.



```
LoadModule was_ap22_module /opt/IBM/WebSphere/Plugins/bin/mod_was_ap22_http.so
WebSpherePluginConfig /opt/IBM/WebSphere/Plugins/config/webserver1/plugin-cfg.xml
```

Ln 1, Col 1

INS



### Information

The WebSpherePluginConfig line at the bottom of the `httpd.conf` is important as the entry is how the web server understands which configuration file to use for the WebSphere plug-in.

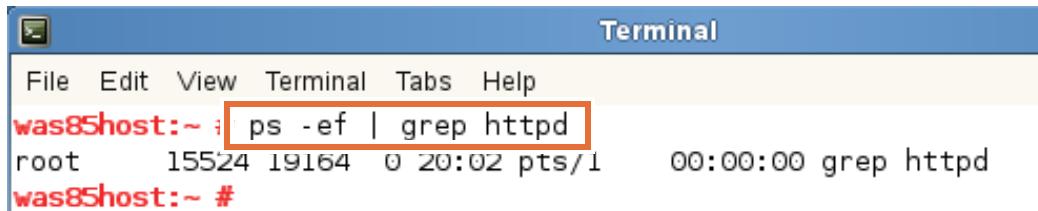
- \_\_\_ d. Close the file when you are finished with your review.
- \_\_\_ e. Notice the `httpd.conf.default` file. This file contains the original configuration parameters. It can be copied, if you want to restore or use the original configuration parameters.
- \_\_\_ f. Open the `admin.conf` file. This file contains configuration data for the administration server. Close the file when you are finished with your review.
- \_\_\_ g. Notice the `admin.conf.default` file. This file contains the original configuration parameters. It can be copied, if you want to restore or use the original administrative configuration parameters.

## **Section 10: Adding a web server as an IBM Installation Manager repository (optional)**

IBM Installation Manager uses repositories to store product package files. Repositories can be located locally or remotely. In this part of the exercise, you configure a web server as a remote repository. Using a web server to host installation repositories can be useful. This approach enables enterprises to share single points of management for installation binary files.

- \_\_\_ 1. Verify that the web server is started.
- \_\_\_ a. From a terminal window, use the following command to check for the `httpd` process:

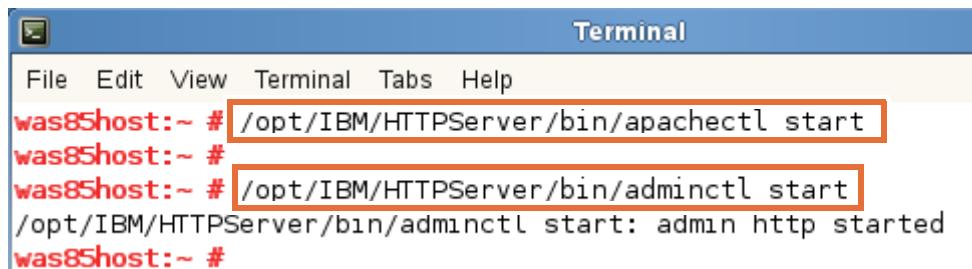
```
ps -ef | grep httpd
```



```
Terminal
File Edit View Terminal Tabs Help
was85host:~ : ps -ef | grep httpd
root 15524 19164 0 20:02 pts/1 00:00:00 grep httpd
was85host:~ #
```

- \_\_ b. If no `httpd` processes are started, start both the HTTP Server and the HTTP Administrative Server with the following commands:

```
/opt/IBM/HTTPServer/bin/apachectl start
/opt/IBM/HTTPServer/bin/adminctl start
```



A screenshot of a terminal window titled "Terminal". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The main area shows a command-line session:

```
was85host:~ # /opt/IBM/HTTPServer/bin/apachectl start
was85host:~ # /opt/IBM/HTTPServer/bin/adminctl start
/opt/IBM/HTTPServer/bin/adminctl start: admin http started
was85host:~ #
```

The first two commands are highlighted with a red box.

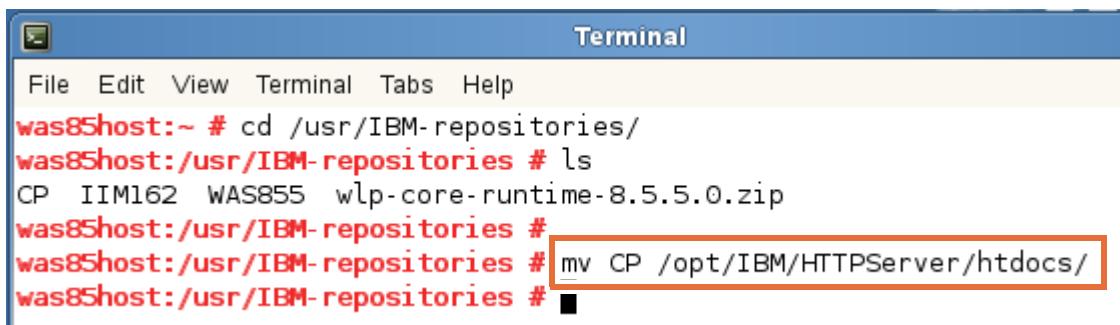
- \_\_ 2. Copy the Caching Proxy installation files to the web server.

- \_\_ a. In a terminal window, navigate to the following directory:

```
/usr/IBM-repositories
```

- \_\_ b. Move the CP (Caching Proxy) installation repository over to the `htdocs` directory for the web server:

```
mv CP /opt/IBM/HTTPServer/htdocs/
```



A screenshot of a terminal window titled "Terminal". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The main area shows a command-line session:

```
was85host:~ # cd /usr/IBM-repositories/
was85host:/usr/IBM-repositories # ls
CP IIM162 WAS855 wlp-core-runtime-8.5.5.0.zip
was85host:/usr/IBM-repositories # mv CP /opt/IBM/HTTPServer/htdocs/
was85host:/usr/IBM-repositories #
```

The last command is highlighted with a red box.

- \_\_ 3. Start IBM Installation Manager.

- \_\_ a. Open a terminal window and navigate to the following directory:

```
/opt/IBM/InstallationManager/eclipse/
```

- \_\_ b. Enter `./IBMIM` to start IBM Installation Manager.

- \_\_ 4. Add the web server as a repository.

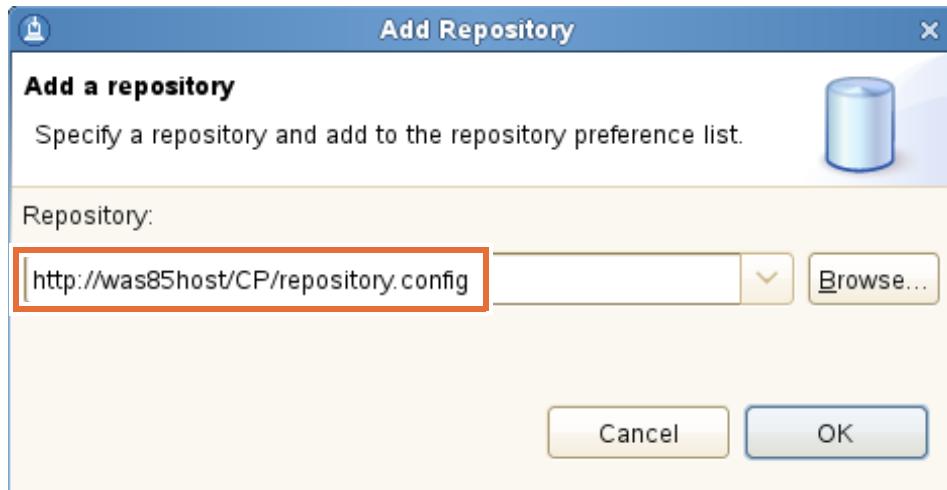
- \_\_ a. Click **File > Preferences**.

- \_\_ b. Select **Repositories**.

- \_\_ c. Select each of the existing repositories and click **Remove Repository**.

- \_\_ d. Click **Add Repository**.

- \_\_ e. Enter `http://was85host/CP/repository.config` and click **OK**.



- \_\_ f. Click **Test Connections**. Verify that you receive a message that indicates the selected repositories are connected.
- \_\_ g. Click **OK**.
- \_\_ h. Click **OK** again.



#### Information

The new repository is now accessible not just through the local file system. It is also accessible to any client that has network access to the local web server.

This feature is significant as it allows administrators to share installation files through one central point on the network.

- \_\_ 5. Explore the Caching Proxy installation. In a normal environment, since the Caching Proxy package is added to the list of repositories, you can install it. For this exercise, you prove that the package is *ready* to install. You do not actually install the package since it is installed.
- \_\_ a. From the main IBM Installation Manager window, click **Install**.

- \_\_\_ b. Although the local repositories are all removed, Installation Manager still displays the contents of the optional repository because the HTTP repository is defined and is being used.

The screenshot shows the 'IBM Installation Manager' interface. At the top, it says 'IBM Installation Manager'. Below that, the title 'Install Packages' is displayed. A sub-header 'Select packages to install:' follows. A table lists packages under the heading 'Installation Packages'. The first item is 'IBM WebSphere Edge Components: Caching Proxy' (Version 8.5.0.0) by IBM. At the bottom left, there's a checkbox for 'Show all versions'. On the right, a button says 'Check for Other Versions, Fixes, ...'. Navigation buttons at the bottom include '?', '< Back', 'Next >', and 'Install'.

- \_\_\_ c. Click **Cancel** to cancel the installation.  
\_\_\_ d. Close IBM Installation Manager.



### Information

The IBM Packaging Utility was used to create the Caching Proxy repository. Not all repositories are usable through an HTTP transport without a minor conversion. The IBM Packaging Utility can modify repositories so that they are usable over an HTTP connection.

The IBM Packaging Utility can also be used to split or join existing repositories for your specific needs. For example, customers removed portions of repositories for unnecessary operating systems. In other cases, they joined repositories to include all the packages for specific environments.

## End of exercise

## Exercise review and wrap-up

In the first part of the exercise, you installed IBM HTTP Server with Installation Manager and confirmed the installation. In the second part of the exercise, you installed the Web Server Plug-ins for WebSphere Application Server. Finally, you examined the installed directories, log files, and configuration files for IBM HTTP Server and plug-ins.



# Exercise 4. Exploring the administrative console

## What this exercise is about

In this exercise, you explore the WebSphere Application Server V8.5.5 administrative console. You examine the configuration structure, and explore the contents of the WebSphere administrative console. The administrative console is used to manage server configuration content, to gain knowledge, and to become familiar with what is available.

## What you should be able to do

At the end of this exercise, you should be able to:

- Verify that WebSphere Application Server is started
- Start the administrative console
- Explore the navigation and functions of the administrative console
- Use the administrative console to examine configuration information, resources, and properties

## Introduction

In this exercise, you explore the WebSphere Application Server V8.5.5 configuration. Exploration includes starting the server and browsing through some of the configuration files. You also start and explore the WebSphere administrative console.

The application server must already be installed and tested, and you must be able to successfully start the server.

The application server runs as a single JVM, including all shared services and the containers to run applications.

The WebSphere administrative console provides a graphical view of the configuration and includes forms and wizards to make it easier to complete configuration tasks.

## Requirements

To do this exercise, you must have a working server and WebSphere administrative console.

## Exercise instructions

### Section 1: Resetting the WebSphere environment



#### Note

To reset your WebSphere environment, read **Appendix A** for instructions on how to complete this procedure.

### Section 2: Start the server

- 1. Before you can configure the application server environment, you must start all the required processes. For this exercise, you use profile1 on server1.
- 2. Determine how many JVMs are running on your system.
  - a. Use the command `ps -ef | grep java` to show the running Java processes.



#### Information

Take note of how many **Java** processes are running. There are no Java processes running, unless the server is running from a previous exercise. Start the server if it is not already running.

- b. In a terminal window, navigate to: `<profile_root>/profile1/bin`
- c. Enter `./startServer.sh server1` to start the server.
- d. After a successful startup of the server you see the message “Server server1 open for e-business”, which indicates that the server is ready and shows its process ID.

The screenshot shows a terminal window titled "Terminal". The window contains the following text:

```
File Edit View Terminal Tabs Help
was8host01:/opt/IBM/WebSphere/AppServer/profiles # cd profile1/bin/
was8host01:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./startServer.sh se
ADMU0116I: Tool information is being logged in file
 /opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1/startServer.
ADMU0128I: Starting tool with the profile1 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 13931
was8host01:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```



## Information

If the server does not start, look at the `startServer.log` in the `<profile_root>/profile1/logs/server1` directory.

- \_\_\_ 3. Examine the running Java processes. Use the command `ps -ef | grep java` to show the running Java processes.

A screenshot of a terminal window titled "Terminal". The menu bar includes "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal window displays the following command and its output:

```
was8host01:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ps -ef | grep java
0 S root 13931 1 5 80 0 - 111097 futex_ 15:44 pts/0 00:01:05 /opt/IBM
erver/java/bin/java -Decclipse.security -Dwas.status.socket=51134 -Dosgi.install.are
phere/AppServer -Dosgi.configuration.area=/opt/IBM/WebSphere/AppServer/profiles/pro
erver1/configuration -Djava.awt.headless=true -Dosgi.framework.extensions=com.ibm.c
clipse.adaptors -Xshareclasses:name=websphrev80_%g,groupAccess,nonFatal -Xbootclas
M/WebSphere/AppServer/java/jre/lib/ext/ibmorb.jar:/opt/IBM/WebSphere/AppServer/java
mext.jar -classpath /opt/IBM/WebSphere/AppServer/profiles/profile1/properties:/opt/
ppServer/properties:/opt/IBM/WebSphere/AppServer/lib/startup.jar:/opt/IBM/WebSphere
bootstrap.jar:/opt/IBM/WebSphere/AppServer/lib/jsf-nls.jar:/opt/IBM/WebSphere/AppSe
y.jar:/opt/IBM/WebSphere/AppServer/lib/urlprotocols.jar:/opt/IBM/WebSphere/AppSe
```

- \_\_\_ a. Compare the process ID for the server that the `startServer` command shows with the one shown by the `ps` command. Yours are going to be different from the process IDs that are shown in the screen captures.

## Section 3: Start the administrative console

The administrative console is the graphical user interface for managing WebSphere Application Server configuration settings for servers, applications, and other resources. The administrative console is a browser-based web application that uses HTML and JavaScript.



## Information

In a federated cell, you always use the administrative console that is connected to the deployment manager so that changes are synchronized across the cell. In a stand-alone application server, you connect directly to the administrative console on the server.

- \_\_\_ 1. Open the administrative console.
  - \_\_\_ a. Open a Firefox browser and enter the following URL:  
`http://was85host:9060/ibm/console`

The browser might show a message that the server connection is untrusted. (This process is reviewed in an optional section of a previous exercise.) This screen shows how Firefox shows this message. Click “**I understand the Risks**”.



## This Connection is Untrusted

You have asked Firefox to connect securely to **was85host:9043**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

### What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

[Get me out of here!](#)

#### ► Technical Details

#### ► **I Understand the Risks**

- \_\_\_ b. Firefox shows this screen. It is fine to click **Add Exception**.

#### ▼ **I Understand the Risks**

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

[Add Exception...](#)

The next screen provides details about this security exception. Examine this screen before you accept this certificate.

- c. Firefox allows you to view the details of the certificate before you confirm the security exception. It is important that you view the certificate before you confirm this security exception. Click **View**.



- \_\_\_ d. The details of the certificate are shown. Why does the browser view this certificate as an error? Why is this issue not a concern for this WebSphere server? Close the certificate window and accept the security exception.



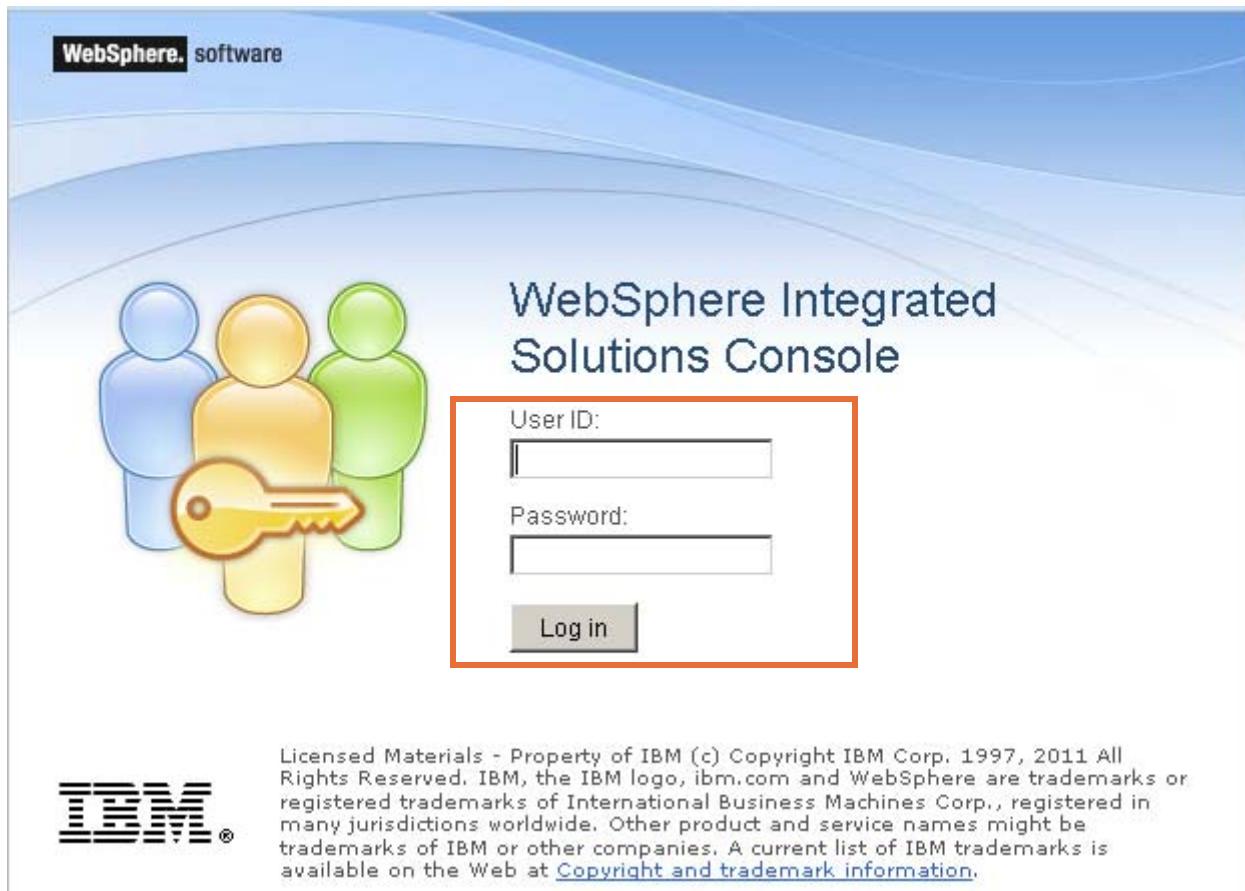
- \_\_\_ e. You must view the certificate, and you must understand the issue here. **Close** the certificate window and click **Confirm Security Exception**. **Permanently store this exception** must be checked.



### Information

It is important that you understand why the certificate for the administrative console generates this error in the browser. The browser is doing a useful security check. WebSphere has a valid certificate and is secure. It is incumbent upon the WebSphere administrator to understand this message and ensure that the certificate is correct.

The administrative console opens.



2. Enter **wasadmin** for the **User ID** and **websphere** for the **Password**. Click **Log in**.



### Information

If security is active, you must log in using a valid user ID and password. If administrative security is disabled, the user ID you enter here does not matter, as it is used to track configuration changes.

A workspace is saved for each user, which includes unsaved configuration changes and navigation preferences.

The administrative console contains three frames:

The screenshot shows the WebSphere Application Server V8.5.5 Administrative Console interface. It consists of three main frames:

- Banner:** The top frame displays a welcome message for the user "wasadmin" and includes links for "Help" and "Logout".
- Navigation tree:** The left frame contains a navigation tree with 13 items: Welcome, Guided Activities, Servers, Applications, Services, Resources, Security, Environment, System administration, Users and Groups, Monitoring and Tuning, Troubleshooting, Service integration, and UDDI.
- Work area:** The right frame is titled "Welcome" and displays information about the Integrated Solutions Console. It shows the version "8.5.5.0" of "WebSphere Application Server". A table provides details about licensed materials, listing "PROPERTY OF IBM" and build numbers: 5724-J08, 5724-I63, 5724-H88, 5724-H89, 5655-W65.

- **Banner:** This area is the top of the administrative console. It shows a welcome message for your user ID. It shows links for logging out of the administrative console and accessing product information.
- **Navigation tree:** This area is the left frame of the administrative console. It shows the types of information you can configure. There are 13 areas:
  - Guided Activities
  - Servers
  - Applications
  - Services
  - Resources
  - Security
  - Environment
  - System administration
  - Users and Groups
  - Monitoring and Tuning
  - Troubleshooting
  - Service integration
  - UDDI

There is also a Welcome link, which takes you back to the main work area home page.

- **Work area:** This area is the right frame of the administrative console. It shows the pages to create or change configuration information. The work area of home shows the installed product version.

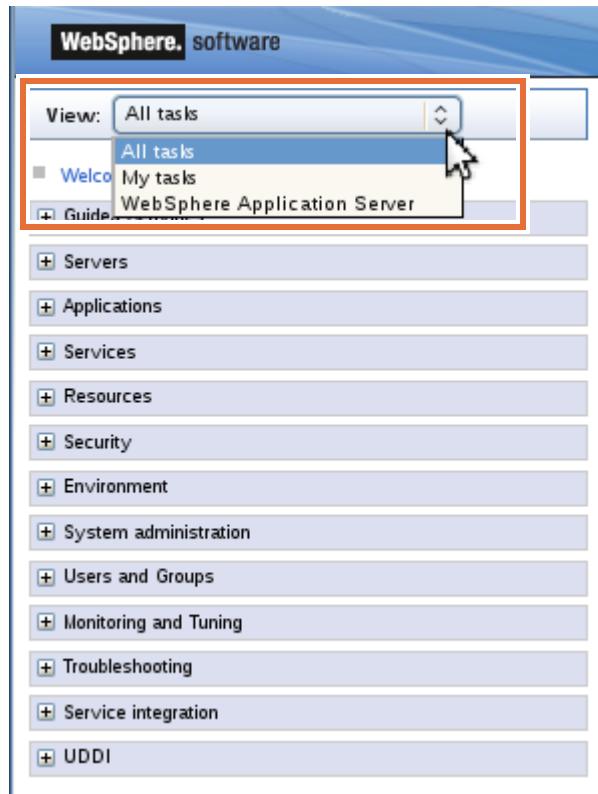
## Section 4: Explore the navigation tree

The administrative console navigation tree lists the tasks available in the administrative console. Tasks are grouped into organizational nodes that represent categories of tasks.

- 1. View the categories of tasks in the navigation tree. When you click a task in the navigation tree, the work area contains one or more modules for completing the task.

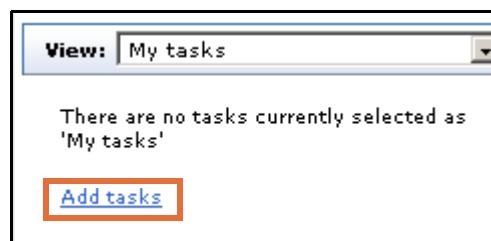
Use the **View** menu at the top of the navigation tree to modify the lists of tasks according to your preferences.

- \_\_\_ 2. Click the **View** menu in the navigation tree.



You can organize the tasks as follows:

- **All tasks:** Shows all tasks in the administrative console.
  - **My tasks:** Shows only the tasks that you added to the view. This list is initially empty, but provides a link to the My Tasks module.
  - **WebSphere Application Server:** Shows only the tasks for this particular product, WebSphere Application Server.
- \_\_\_ 3. Using My tasks allows you to create and edit a list of tasks to view in the navigation tree. The “My tasks” selection is especially useful to customize the navigation to show only the tasks that you use most often.
- \_\_\_ a. Select **My tasks** from the menu in View.
  - \_\_\_ b. No tasks are currently selected. Click **Add tasks** to add a task to the view.



- \_\_\_ c. The work area shows the tasks that you can select to customize the My tasks view in the navigation tree.

\_\_ d. Check the boxes for **Servers**, **Applications**, and **Resources**. Click **Apply**.

My Tasks

Select the tasks you wish to add to the 'My tasks' list.

Welcome  
 Guided Activities  
 Servers  
 Applications  
 Services  
 Resources  
 Security  
 Environment  
 System administration  
 Users and Groups  
 Monitoring and Tuning  
 Troubleshooting  
 Service integration  
 UDDI

- \_\_\_ e. After applying your selections, your customized task list is seen in the navigation tree.

The screenshot shows the Integrated Solutions Console interface. On the left, there's a sidebar with a 'View' dropdown set to 'My tasks'. Below it is a link 'Edit 'My tasks''. The main area lists various administrative tasks. On the right, a 'My Tasks' panel displays a subset of these tasks, each with a checkbox. Some checkboxes are checked (e.g., 'Servers', 'Applications', 'Resources'), while others are unchecked. At the bottom of the right panel are 'Apply' and 'Reset' buttons.

| Available Tasks                                | Selected Tasks |
|------------------------------------------------|----------------|
| WebSphere application servers                  |                |
| WebSphere MQ servers                           |                |
| Web servers                                    |                |
| New Application                                |                |
| WebSphere enterprise applications              |                |
| Business-level applications                    |                |
| Assets                                         |                |
| Schedulers                                     |                |
| Object pool managers                           |                |
| JMS providers                                  |                |
| Connection factories                           |                |
| Queue connection factories                     |                |
| Topic connection factories                     |                |
| Queues                                         |                |
| Topics                                         |                |
| Activation specifications                      |                |
| JDBC providers                                 |                |
| Data sources                                   |                |
| Data sources (WebSphere Application Server V4) |                |
| Resource adapters                              |                |
| J2C connection factories                       |                |
| J2C activation specifications                  |                |
| J2C administered objects                       |                |
| Timer managers                                 |                |
| Work managers                                  |                |
| Object cache instances                         |                |
| Servlet cache instances                        |                |
| Mail providers                                 |                |
| Mail sessions                                  |                |
| URL providers                                  |                |
| URLs                                           |                |
| Resource Environment Providers                 |                |
| Resource environment entries                   |                |

- \_\_\_ f. Continue to explore and customize the **My tasks** view and add more tasks.  
 \_\_\_ g. Select **All tasks** from the View menu.

## Section 5: Explore guided activities

In this part of the exercise, you look at the guided activities for WebSphere Application Server V8.5. Guided activities lead you through common administrative tasks that require you to go to multiple administrative console pages.

- \_\_\_ 1. In the administrative console navigation tree, expand **Guided Activities**.  
 In the stand-alone environment, the guided activities include:
  - Connecting to a database

- Routing requests through a web server to an application server



### Information

In the federated environment, the guided activities include:

- Connecting to a database
- Routing requests through a web server to an application server
- Configuring a cluster and configuring highly available applications

- \_\_ a. Click **Connecting to a database** to view the first activity.

The work area has information about the activity to help you complete this task successfully. It contains an introduction to the task, and details other tasks to do before and after completing this task. The work area also provides hints and tips to help you avoid and recover from problems and other tasks.

The screenshot shows the WebSphere Application Server administrative console interface. The left sidebar lists various categories: Welcome, Guided Activities, Servers, Applications, Services, Resources, Security, Environment, System administration, Users and Groups, Monitoring and Tuning, Troubleshooting, Service integration, and UDDI. The 'Guided Activities' section is expanded, showing two items: 'Connecting to a database' (which is selected and highlighted in blue) and 'Routing requests through a Web server to an application server'. The main content area is titled 'Connecting to a database' and contains the following sections:

- Introduction:** A text block explaining that the guided activity leads through steps to configure database access for an application, assuming database software is installed and configured. It states that after completion, the application will be able to access data from a database.
- To continue, click **Start**.**
- Start** button (with a gear icon) and **Help** link.
- A list of tasks with plus signs:
  - Configure credentials for secure database access
  - Configure a JDBC provider
  - Configure WebSphere variables
  - Configure a data source
  - Save and synchronize configuration
  - Test database connection

On the right side of the main content area, there is a diagram illustrating a three-tier architecture with three server icons connected to a central database icon. Below the diagram, the heading 'Assumptions' is followed by a note stating it is assumed that you can securely access data information on this topic. There is also a 'Before you begin' section with a bulleted list:

- [Configure](#)
- [Deploy](#)
- [Learn about](#)

- \_\_ b. Continue to explore the details for connecting to a database, or select another guided activity to explore.

## Section 6: Explore server settings

In this part of the exercise, you look at some of the settings that can be configured with the administrative console. You begin by looking at the server section.

- \_\_ 1. Explore the server settings.
- \_\_ a. In the administrative console navigation tree, expand **Servers** and **Server Types**.

In the stand-alone environment, the only server types are:

- WebSphere application servers
- WebSphere MQ servers
- Web servers



### Information

In the federated environment, you can also manage:

- On demand routers
- PHP servers
- WebSphere Application Server Community Edition servers
- Generic servers
- Proxy servers
- Apache servers
- Custom HTTP servers
- Clusters
- Cluster topology
- Generic server clusters
- Core groups

\_\_ b. Click **WebSphere application servers**.

The work area is a table that lists the application servers. You have one server, **server1**.

This page is known as a collection page because the list is a collection of objects. The page has two options for controlling the amount of information that is shown, Filter and Preferences.

| Application servers                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                 |           |            |      |      |           |         |         |                 |           |            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-----------|------------|------|------|-----------|---------|---------|-----------------|-----------|------------|
| <b>Application servers</b><br>Use this page to view a list of the application servers in your environment and the status of each of these servers. You can also use this page to change the status of a specific application server.<br><input checked="" type="checkbox"/> Preferences                                                                                                                                                                                |                 |           |            |      |      |           |         |         |                 |           |            |
|  <br><table border="1"> <thead> <tr> <th>Name</th> <th>Node</th> <th>Host Name</th> <th>Version</th> </tr> </thead> <tbody> <tr> <td>server1</td> <td>was85hostNode01</td> <td>was85host</td> <td>ND 8.5.5.0</td> </tr> </tbody> </table> You can administer the following resources:<br>Total 1 |                 |           |            | Name | Node | Host Name | Version | server1 | was85hostNode01 | was85host | ND 8.5.5.0 |
| Name                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Node            | Host Name | Version    |      |      |           |         |         |                 |           |            |
| server1                                                                                                                                                                                                                                                                                                                                                                                                                                                                | was85hostNode01 | was85host | ND 8.5.5.0 |      |      |           |         |         |                 |           |            |



## Information

For some other collection pages, an extra option, Scope, is presented. An example of scope is seen later.

- \_\_\_ c. Click the **Filter** icon.

The screenshot shows the 'Application servers' page. At the top left is a blue header bar with the title 'Application servers' and a question mark icon. Below the header is a section titled 'Application servers' with a descriptive paragraph. A 'Preferences' button is visible. The main area contains a table with columns: Name, Node, Host Name, and Version. Below the table is a filter section with a 'Filter' dropdown set to 'Name', a 'Search terms:' input field containing '\*', and a 'Go' button. A message says 'To filter the following table, select the column by which to filter, then enter filter criteria (wildcards: \*,?,%).'. At the bottom, there's a section for administering resources with a table showing 'server1' and its details: 'was85hostNode01', 'was85host', and 'ND 8.5.5.0'. A total count of '1' is shown.

The filter feature allows you to use wildcards to match only the objects you want to work with if there are many objects of the same type. You can select a table column and specify the text to match.



## Information

This option is rarely necessary unless there are many items.

- \_\_\_ d. Click **server1**.

The configuration of server1 is seen. This page is known as a details page. There are two pages, each with a tab at the top:

- **Runtime**

Runtime lists the current information that the running server uses.

## - Configuration

Configuration lists the saved settings that are used when the server is next started.

The screenshot shows the 'Application servers' configuration page for 'server1'. The 'Runtime' tab is selected. The 'General Properties' section includes fields for Name (server1), Node name (was85host01 Node01), and checkboxes for Run in development mode, Parallel start, and Start components as needed. The 'Access to internal server classes' dropdown is set to 'Allow'. The 'Server-specific Application Settings' section includes dropdowns for Classloader policy (Multiple) and Class loading mode (Classes loaded with parent class loader first). On the right, there are links for Container Settings, Applications, Server messaging, and Server Infrastructure, each with expandable sections.

| General Properties                                  |                    | Container Settings        |                                           |
|-----------------------------------------------------|--------------------|---------------------------|-------------------------------------------|
| Name                                                | server1            | Session management        | <a href="#">Session management</a>        |
| Node name                                           | was85host01 Node01 | SIP Container Setting     | <a href="#">SIP Container Setting</a>     |
| <input type="checkbox"/> Run in development mode    |                    | Web Container Setting     | <a href="#">Web Container Setting</a>     |
| <input checked="" type="checkbox"/> Parallel start  |                    | Portlet Container Setting | <a href="#">Portlet Container Setting</a> |
| <input type="checkbox"/> Start components as needed |                    | EJB Container Setting     | <a href="#">EJB Container Setting</a>     |
| Access to internal server classes                   | Allow              | Container Services        | <a href="#">Container Services</a>        |
|                                                     |                    | Business Process Ser      | <a href="#">Business Process Ser</a>      |
| Applications                                        |                    |                           |                                           |
| <a href="#">Installed applications</a>              |                    |                           |                                           |
| Server messaging                                    |                    |                           |                                           |
| <a href="#">Messaging engines</a>                   |                    |                           |                                           |
| <a href="#">Messaging engine inb</a>                |                    |                           |                                           |
| <a href="#">transports</a>                          |                    |                           |                                           |
| <a href="#">WebSphere MQ link in</a>                |                    |                           |                                           |
| <a href="#">transports</a>                          |                    |                           |                                           |
| <a href="#">SIB service</a>                         |                    |                           |                                           |
| Server Infrastructure                               |                    |                           |                                           |
| <a href="#">Java and Process</a>                    |                    |                           |                                           |
| <a href="#">Management</a>                          |                    |                           |                                           |
| <a href="#">Administration</a>                      |                    |                           |                                           |

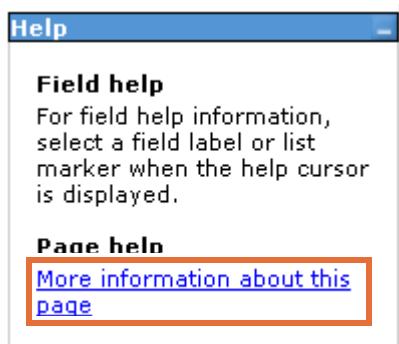
Classloader policy: **Multiple**

Class loading mode: **Classes loaded with parent class loader first**

Buttons: Apply, OK, Reset, Cancel

Some basic configuration settings are shown under **General Properties**, including the **Classloader policy** and the **Class loading mode**.

For a description of any of the settings, click **More information about this page** in the **Help** box. The Help section is on the far right of the screen. On some screens, it is necessary to scroll right to see the Help section.



- \_\_\_ e. Under Server Infrastructure, click to expand **Java and Process Management**. Click **Process definition**.

Use this page to view or change settings for a process definition. This page provides command-line information for starting or initializing a process.

The Working directory entry starts with a \$ and is a **WebSphere variable** ( `${USER_INSTALL_ROOT}`). These variables allow for substitutions to the absolute paths. You explore **WebSphere variables** later in this exercise.

- \_\_\_ f. Click **Java Virtual Machine** under Additional Properties.

The advanced JVM settings for server1 are seen. Scroll down and examine the settings. Use the Help box to get default values for these settings.

- What is the value of Maximum heap size? \_\_\_\_\_
- Is Debug Mode used? \_\_\_\_\_
- Is the JIT (just-in-time compiler) set? \_\_\_\_\_

- \_\_\_ g. Click **Cancel** to return to the Process Definition page.

- \_\_\_ h. Click **Process Logs** under Additional Properties.

The process log settings for server1 are shown. These logs are the native stdout and stderr log files for the JVM process.



### Note

These logs are different from the `SystemOut.log` and `SystemErr.log` files, which capture most output from the JVM, and support log file rotation to prevent the files from growing too large.

2. Explore the server details.
- a. At the top of the page, a breadcrumb trail shows the pages that you went to.

### Application servers

#### Application servers > **server1** > Process definition > Process Logs

Use this page to view or modify settings to specify the files to which standard out and standard error stream process logs are created by redirecting the standard out and standard error streams of a process to independent Native code writes to the process logs. These logs can also contain information that relates to problems in native diagnostic information written by the JVM. One set of process logs is created for each application server and applications. Process logs are also created for the deployment manager and each node manager. Changes on the Configuration panel apply when the server is restarted. Changes on the Runtime panel apply immediately.

Configuration      **Runtime**

General Properties

\* Stdout File Name  
\${SERVER\_LOG\_ROOT}/native\_stdout.log

\* Stderr File Name  
\${SERVER\_LOG\_ROOT}/native\_stderr.log

- b. Click **server1** from the breadcrumb trail to return to the server1 configuration page.

- \_\_ c. Under Communications, expand **Ports**. The TCP/IP ports that server1 uses are listed.

| Communications                        |       |         |
|---------------------------------------|-------|---------|
| Ports                                 |       | Details |
| Port Name                             | Port  |         |
| BOOTSTRAP_ADDRESS                     | 2809  |         |
| SOAP_CONNECTOR_ADDRESS                | 8880  |         |
| ORB_LISTENER_ADDRESS                  | 9100  |         |
| SAS_SSL_SERVERAUTH_LISTENER_ADDRESS   | 9401  |         |
| CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS | 9403  |         |
| CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS | 9402  |         |
| WC_adminhost                          | 9060  |         |
| WC_defaulthost                        | 9080  |         |
| DCS_UNICAST_ADDRESS                   | 9353  |         |
| WC_adminhost_secure                   | 9043  |         |
| WC_defaulthost_secure                 | 9443  |         |
| SIP_DEFAULTHOST                       | 5060  |         |
| SIP_DEFAULTHOST_SECURE                | 5061  |         |
| SIB_ENDPOINT_ADDRESS                  | 7276  |         |
| SIB_ENDPOINT_SECURE_ADDRESS           | 7286  |         |
| SIB_MQ_ENDPOINT_ADDRESS               | 5558  |         |
| SIB_MQ_ENDPOINT_SECURE_ADDRESS        | 5578  |         |
| IPC_CONNECTOR_ADDRESS                 | 9633  |         |
| OVERLAY_UDP_LISTENER_ADDRESS          | 11003 |         |
| OVERLAY_TCP_LISTENER_ADDRESS          | 11004 |         |

**[+] Messaging**

\_\_ d. Click Details to get more information about these ports.

**Application servers**

**Application servers > server1 > Ports**

**[+] Preferences**

| <b>Select</b>            | <b>Port Name</b>                                      | <b>Host</b>        | <b>Port</b> | <b>Transport Details</b>                   |
|--------------------------|-------------------------------------------------------|--------------------|-------------|--------------------------------------------|
| <input type="checkbox"/> | <a href="#">BOOTSTRAP ADDRESS</a>                     | was85host          | 2809        | No associated transports                   |
| <input type="checkbox"/> | <a href="#">CSIV2 SSL MUTUALAUTH LISTENER ADDRESS</a> | was85host          | 9402        | No associated transports                   |
| <input type="checkbox"/> | <a href="#">CSIV2 SSL SERVERAUTH LISTENER ADDRESS</a> | was85host          | 9403        | No associated transports                   |
| <input type="checkbox"/> | <a href="#">DCS UNICAST ADDRESS</a>                   | *                  | 9353        | <a href="#">View associated transports</a> |
| <input type="checkbox"/> | <a href="#">IPC CONNECTOR ADDRESS</a>                 | \${LOCALHOST_NAME} | 9633        | No associated transports                   |
| <input type="checkbox"/> | <a href="#">ORB LISTENER ADDRESS</a>                  | was85host          | 9100        | No associated transports                   |
| <input type="checkbox"/> | <a href="#">OVERLAY TCP LISTENER ADDRESS</a>          | *                  | 11004       | No associated transports                   |
| <input type="checkbox"/> | <a href="#">OVERLAY UDP LISTENER ADDRESS</a>          | *                  | 11003       | No associated transports                   |
| <input type="checkbox"/> | <a href="#">SAS SSL SERVERAUTH LISTENER ADDRESS</a>   | was85host          | 9401        | No associated transports                   |
| <input type="checkbox"/> | <a href="#">SIB ENDPOINT ADDRESS</a>                  | *                  | 7276        | <a href="#">View associated transports</a> |
| <input type="checkbox"/> | <a href="#">SIB ENDPOINT SECURE ADDRESS</a>           | *                  | 7286        | <a href="#">View associated transports</a> |
| <input type="checkbox"/> | <a href="#">SIB MQ ENDPOINT ADDRESS</a>               | *                  | 5558        | <a href="#">View associated transports</a> |
| <input type="checkbox"/> | <a href="#">SIB MQ ENDPOINT SECURE ADDRESS</a>        | *                  | 5578        | <a href="#">View associated transports</a> |
| <input type="checkbox"/> | <a href="#">SIP DEFAULTHOST</a>                       | *                  | 5060        | <a href="#">View associated transports</a> |
| <input type="checkbox"/> | <a href="#">SIP DEFAULTHOST SECURE</a>                | *                  | 5061        | <a href="#">View associated transports</a> |
| <input type="checkbox"/> | <a href="#">SOAP CONNECTOR ADDRESS</a>                | was85host          | 8880        | No associated transports                   |
| <input type="checkbox"/> | <a href="#">WC adminhost</a>                          | *                  | 9060        | <a href="#">View associated transports</a> |
| <input type="checkbox"/> | <a href="#">WC adminhost secure</a>                   | *                  | 9043        | <a href="#">View associated transports</a> |
| <input type="checkbox"/> | <a href="#">WC defaulthost</a>                        | *                  | 9080        | <a href="#">View associated transports</a> |
| <input type="checkbox"/> | <a href="#">WC defaulthost secure</a>                 | *                  | 9443        | <a href="#">View associated transports</a> |

Total 20

- \_\_ e. Click **SOAP\_CONNECTOR\_ADDRESS** to configure the port.

The host and port that are associated with the SOAP listener are seen in the Port column. SOAP clients, like wsadmin, use this port to connect to the server to do administrative tasks. On a single server installation, the default SOAP port is 8880.

- \_\_ f. Click **server1** in the breadcrumb trail to return to the server1 details page.  
 \_\_ g. Under Server Infrastructure, expand **Administration**. Click **Server Components**.

The internal components of server1 are listed. The resource that you can administer is **Name Server**.

- \_\_ h. Click **Name Server**.

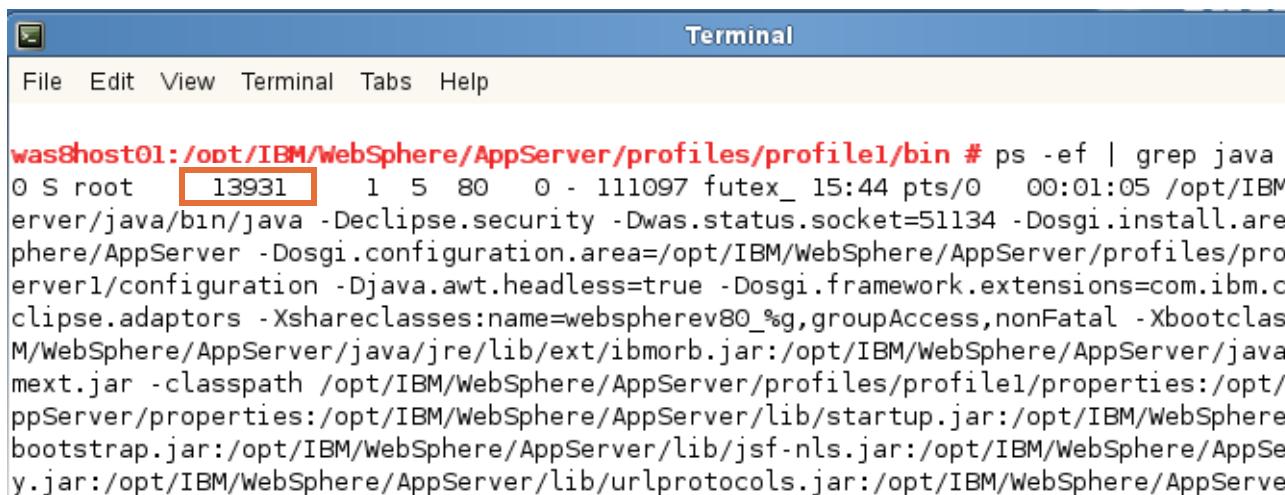
The name service settings for the application server are seen.

- \_\_ i. Click **server1** in the breadcrumb trail to return to the server1 details page.  
 \_\_ j. Click the **Runtime** tab.

The properties of the currently running instance of server1 are shown:

- Process ID (PID). Record your process ID: \_\_\_\_\_
- Cell name
- Node name
- State (Started)

- \_\_ k. From a terminal window, use the command `ps -ef | grep java` and verify that the process ID shown matches the PID for server1.



```
was8host01:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ps -ef | grep java
0 S root 13931 1 5 80 0 - 111097 futex_ 15:44 pts/0 00:01:05 /opt/IBM
ervert/java/bin/java -Decclipse.security -Dwas.status.socket=51134 -Dosgi.install.are
phere/AppServer -Dosgi.configuration.area=/opt/IBM/WebSphere/AppServer/profiles/pro
file1/configuration -Djava.awt.headless=true -Dosgi.framework.extensions=com.ibm.c
lipse.adaptors -Xshareclasses:name=websphrev80_%g,groupAccess,nonFatal -Xbootclas
M/WebSphere/AppServer/java/jre/lib/ext/ibmorb.jar:/opt/IBM/WebSphere/AppServer/java
mext.jar -classpath /opt/IBM/WebSphere/AppServer/profiles/profile1/properties:/opt/
ppServer/properties:/opt/IBM/WebSphere/AppServer/lib/startup.jar:/opt/IBM/WebSphere
bootstrap.jar:/opt/IBM/WebSphere/AppServer/lib/jsf-nls.jar:/opt/IBM/WebSphere/AppSe
y.jar:/opt/IBM/WebSphere/AppServer/lib/urlprotocols.jar:/opt/IBM/WebSphere/AppServe
```

## Section 7: Examine application settings

- \_\_ 1. Go to the administrative console navigation tree. Expand **Applications** and **Application Types**.

There are three application types:

- WebSphere enterprise applications
- Business-level applications
- Assets

2. Click **WebSphere enterprise applications**.

A collection page lists the applications in the configuration and their status. If you move the mouse cursor over a status icon and click, a window opens and shows the status.

| Select                   | Name                               | Application Status |
|--------------------------|------------------------------------|--------------------|
| <input type="checkbox"/> | <a href="#">DefaultApplication</a> |                    |
| <input type="checkbox"/> | <a href="#">ivtApp</a>             |                    |
| <input type="checkbox"/> | <a href="#">query</a>              |                    |

Total 3

The applications that are installed by default include:

- **DefaultApplication**: includes the snoop servlet.
- **ivtApp**: installation verification test.
- **query**: for information about Enterprise JavaBeans.
- If the sample applications were installed, they would also be listed here.

At the moment you have only one server, but potentially the list includes applications that are installed on multiple servers on the same computer or in a network deployment cell.



## Information

Tasks that can be done on an application include:

- Start
- Stop
- Install
- Uninstall
- Update
- Rollout Update
- Remove File
- Export
- Export DDL
- Export File

— 3. Click **DefaultApplication**.

This page shows the general properties of the application with links to a number of properties pages such as Details, Web Module, and Enterprise JavaBeans Properties. It also shows a References page.

— 4. Under Modules, click **Manage Modules**.

You see one web module, **Default Web Application**; one EJB module, **Increment EJB module**; and the servers with which they are associated.

— 5. Click the **Default Web Application** module.

A detail page shows the general properties that are associated with the deployment of the web module.

— 6. Click **Manage Modules** in the breadcrumb trail and select the **Increment EJB module**.

A detail page shows the general properties that are associated with the deployment of the EJB module.

## **Section 8: Examine environment settings**

— 1. In the administrative console navigation tree, expand **Environment**. The Environment options include:

- Virtual hosts
- Update global web server plug-in configuration
- WebSphere variables
- Shared libraries
- SIP application routers
- Replication domains
- Naming
- OSGi bundle repositories

— 2. Click **Virtual Hosts**.

The work area of a collection page lists the virtual hosts that are defined for the cell.

— 3. Click **default\_host**.

A details page shows the details for the virtual host. You can directly change only the virtual host name. Under Additional Properties, you find links to other properties pages.

\_\_\_ 4. Click **Host Aliases** under Additional Properties.

The host name and port combinations that are associated with this virtual host are shown. For the default\_host, the default values are:

- \*:9080 (any host on the internal HTTP transport port)
- \*:80 (any host on the external HTTP transport port)
- \*:9443 (any host on the internal SSL transport port)
- \*:5060 (any host on the SIP transport port)
- \*:5061 (any host on the SIP transport port)
- \*:443 (any host on the external SSL transport port).

You can define more virtual hosts or modify default\_host to support more host-port combinations.

\_\_\_ 5. Click **Virtual Hosts** in the breadcrumb trail to return to the Virtual Hosts page.

\_\_\_ 6. Click **admin\_host** and then click **Host Aliases**.

\_\_\_ 7. Examine the admin\_host virtual host and write down the port numbers that are associated with this virtual host: \_\_\_\_\_



**Note**

The browser accesses the administrative console on one of the ports that are associated with the admin\_host virtual host.

\_\_\_ 8. In the administrative console navigation tree, under the Environment section, click **WebSphere variables**.

The work area of a WebSphere variables collection page is displayed. This page includes the scope feature because variables can be defined for a cell, node, or

server. A menu of all available scopes is provided to narrow the list of variables that are based on scope.

**WebSphere Variables**

**WebSphere Variables**

Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values such as file system root directories. Variables have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable has conflicting scope values, the more granular scope values override values at greater scope levels. Therefore, server variables override node variables, which override cluster variables, which override cell variables.

Scope: **All scopes**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

- \_\_\_ 9. From the scope menu, select scope **Cell=<cellname>**. How many variables are defined for the cell? \_\_\_\_\_
- \_\_\_ 10. From the scope menu, select scope **Node=<nodename>**. How many variables are defined for the node? \_\_\_\_\_
- If there are more than the maximum rows (20 by default), click **Next** to allow you to see the additional entries.
- Notice that many variable values include references to other variables, for example, \${USER\_INSTALL\_ROOT}.
- \_\_\_ 11. From the scope menu, select scope **Node=<nodename>, Server=<servername>**. How many variables are defined for the server? \_\_\_\_\_

## Section 9: Examine resource settings

- \_\_\_ 1. In the administrative console navigation tree, expand **Resources**. The resources options are:
- Schedulers
  - Object pool managers
  - JMS
  - JDBC
  - Resource adapters
  - Asynchronous beans
  - Cache instances
  - Mail
  - URL
  - Resource environment
- \_\_\_ 2. Expand **JDBC**; click **JDBC providers**.

A collection page lists the JDBC providers in the configuration. In a later exercise, you configure a JDBC driver and data source for an application.

- \_\_\_ 3. Expand **JMS** and click **JMS providers**. Click an instance of **Default messaging provider**.

A details page shows some basic properties of the internal JMS provider. Under Additional Properties, there are links for:

- Connection factories (for configuring a JMS connection factory)
- Queue connection factories
- Topic connection factories
- Queues
- Topics
- Activation specifications

These settings can be defined at the cell, node, or server level so there is a scope selection option available.

A collection page lists queue connection factories (if there are any defined). A queue connection factory is used to create connections to the associated JMS provider of JMS queue destinations, for point-to-point messaging.

A collection page lists topic connection factories (if there are any defined). A topic connection factory is used to create connections to the associated JMS provider of JMS topic destinations, for publish/subscribe messaging.

## **Section 10:Examine troubleshooting**

The Troubleshooting area shows messages about runtime events and configuration problems. This area automatically refreshes, and you can view either the runtime messages or configuration problem totals.

- \_\_\_ 1. In the administrative console navigation tree, expand **Troubleshooting**.  
\_\_\_ 2. Expand **Runtime Messages**. You see entries for:

- Runtime error
- Runtime warning
- Runtime information



### Note

Runtime Events are disabled by default (“None”). To enable an event level, select from the list. “Error” would enable only Error runtime events. “Warning” would enable both Error and Warning runtime events. “Info” would enable all runtime events.

- \_\_\_ 3. The total number of errors, warnings, and information messages are shown when you select one of the options. Click **Runtime information** to view all of the messages.
- \_\_\_ 4. Select **Info** from the list and click **Apply**.
- \_\_\_ 5. Click **Save** to save the configuration.
- \_\_\_ 6. In the **Message** column, click one of the messages (if there are any listed) to see the message detail.
- \_\_\_ 7. In the navigation tree, expand **Configuration Validation**. You see entries for:
  - Configuration error
  - Configuration warning
  - Configuration information

The total number of errors, warnings, and information messages are shown when you select one of the options. Click **Configuration error** to view all of the error messages. If you do not have any error messages, click the **Configuration information** messages.

- \_\_\_ 8. If you have a configuration problem, click the link to it. The problem detail is seen. On the next window, you see general properties information about the configuration problem.
- \_\_\_ 9. Click **Back** to return to the Configuration Validation list and view other problems, if any exist.

## **Section 11:Modify the administrative console session timeout**

When you are working with the administrative console, the session expires if it is idle for more than 30 minutes. To continue working, you must log in again. Many administrators find the default session idle duration too short. You can change the session idle duration to a time that works best for you. The session idle duration time cannot be modified from the administrative console. The timeout must be modified by running a script.



### **Information**

The administrative console session expiration script that you use in this exercise is found in the information center by searching for: changing the console session expiration

- \_\_\_ 1. Review the sample administrative console session expiration script.
  - \_\_\_ a. From a terminal window, navigate to the `/usr/software/wsadmin/` directory.
  - \_\_\_ b. Open `consoleTimeout.py` with a text editor.

- \_\_ c. Look at this script. Notice the comments at the top that show how to use this script.

```
consoleTimeout.py ✘
```

```
#-----
Name: consoleTimeout.py
Role: Display or change the AdminConsole inactivity timeout value.
Author: Robert A. (Bob) Gibson
Note: Based upon the Jacl script in the online documentation
http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.v7r0.admin.doc
#-----
import re, sys;

#-----
Name: Usage()
Role: Display script usage information, and exit (terminate script)
#-----
def Usage(cmdName):
 print """
Command: %(cmdName)s\n
Purpose: WebSphere (wsadmin) script used to display, or modify the Admin
 Console timeout value.\n
Usage: %(cmdName)s [value]\n
Where:
 value = An optional numeric value representing the number of minutes of
 inactivity that are allowed. If no value is specified, the
 current invalidationTimeout value is displayed.\n
Examples:
 wsadmin -lang jython -f %(cmdName)s.py\n
 wsadmin -lang jython -f %(cmdName)s.py 30''' % locals()
 sys.exit(1)
```

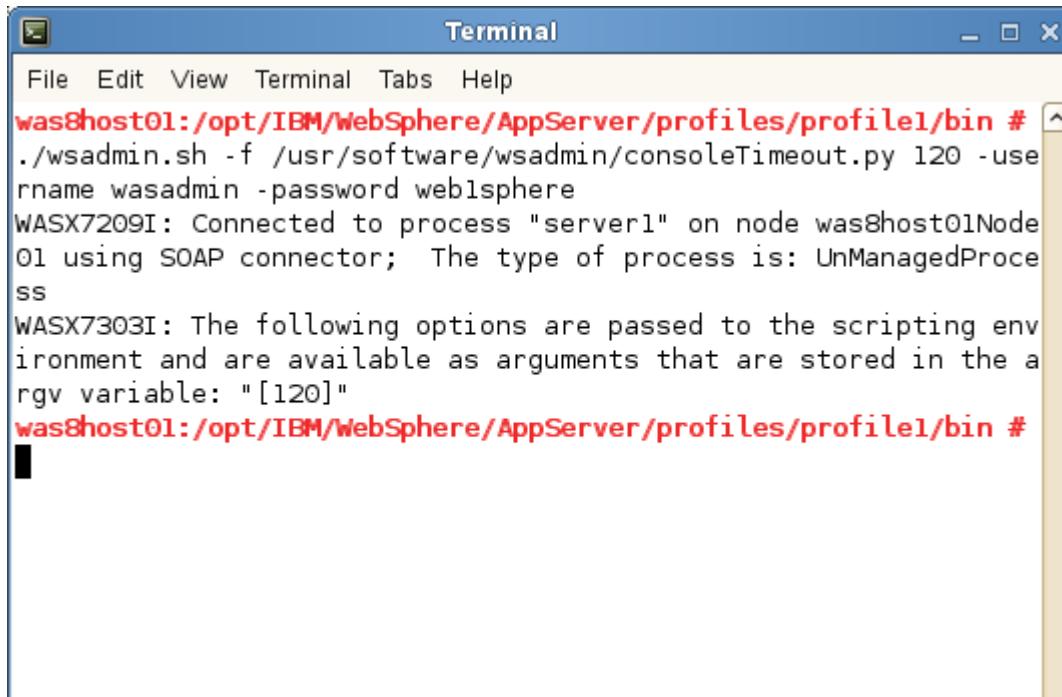
---

- \_\_ d. Close the file when you are done examining it. Do not save any changes that are made to the file.

2. Run the script to change the session expiration.

- a. From the command line, run the `consoleTimeout.py` script with `wsadmin` from the `/opt/IBM/WebSphere/AppServer/profiles/profile1/bin` directory. Type in the following command string on one line.

```
./wsadmin.sh -f /usr/software/wsadmin/consoleTimeout.py 120
-username wasadmin -password websphere
```



The screenshot shows a terminal window with the title "Terminal". The menu bar includes "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal prompt is "was8host01:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #". The user has typed the command:

```
./wsadmin.sh -f /usr/software/wsadmin/consoleTimeout.py 120 -use
rname wasadmin -password websphere
```

Output from the command:

```
WASX7209I: Connected to process "server1" on node was8host01Node
01 using SOAP connector; The type of process is: UnManagedProce
ss
WASX7303I: The following options are passed to the scripting env
ironment and are available as arguments that are stored in the a
rgv variable: "[120]"
```

The terminal prompt appears again at the bottom:

```
was8host01:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```

**Note**

The first time a wsadmin command is started, the environment is set up. There are several messages about “processing new jar”. These messages are expected the first time that wsadmin is run.



Terminal

```
File Edit View Terminal Tabs Help
was8host01:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./wsadmin.sh -f
onsoleTimeout.py 120 -username wasadmin -password web1spHERE
sys-package-mgr: processing new jar, '/opt/IBM/WebSphere/AppServer/lib/startup
sys-package-mgr: processing new jar, '/opt/IBM/WebSphere/AppServer/lib/bootstrap
sys-package-mgr: processing new jar, '/opt/IBM/WebSphere/AppServer/lib/lmproxy
sys-package-mgr: processing new jar, '/opt/IBM/WebSphere/AppServer/lib/urlprot
sys-package-mgr: processing new jar, '/opt/IBM/WebSphere/AppServer/java/lib/to
sys-package-mgr: processing new jar, '/opt/IBM/WebSphere/AppServer/deploytool/
sys-package-mgr: processing new jar, '/opt/IBM/WebSphere/AppServer/deploytool/
sys-package-mgr: processing new jar, '/opt/IBM/WebSphere/AppServer/optionalLib
r'
sys-package-mgr: processing new jar, '/opt/IBM/WebSphere/AppServer/plugins/com
r'
sys-package-mgr: processing new jar, '/opt/IBM/WebSphere/AppServer/plugins/com
sys-package-mgr: processing new jar, '/opt/IBM/WebSphere/AppServer/plugins/org
jar'
```

**Information**

In the commands above, the **user name** and **password** are specified on the command line. Administrative scripts, such as wsadmin, support specifying the user name and password on the command line, in the properties file, or through prompting (GUI or command line).

- \_\_\_\_\_ b. The session expiration timeout is now set for 120 minutes.

**Information**

The timeout session expiration must be set for each profile’s administrative console. Later in the class you create more profiles. To change the session timeout, you must rerun the timeout script for each profile that you want.

**Note**

The wsadmin tool can be used to run scripts. You learn more about wsadmin later in the course.

### **Section 12: Log out of the administrative console**

When you are working in the administrative console, a work area is saved which includes all configuration changes you make in the session. When you log out, you can save or discard these changes. If you close the browser, the session work area is preserved. The next time you log in, you can recover the work area from the previous session.

**Information**

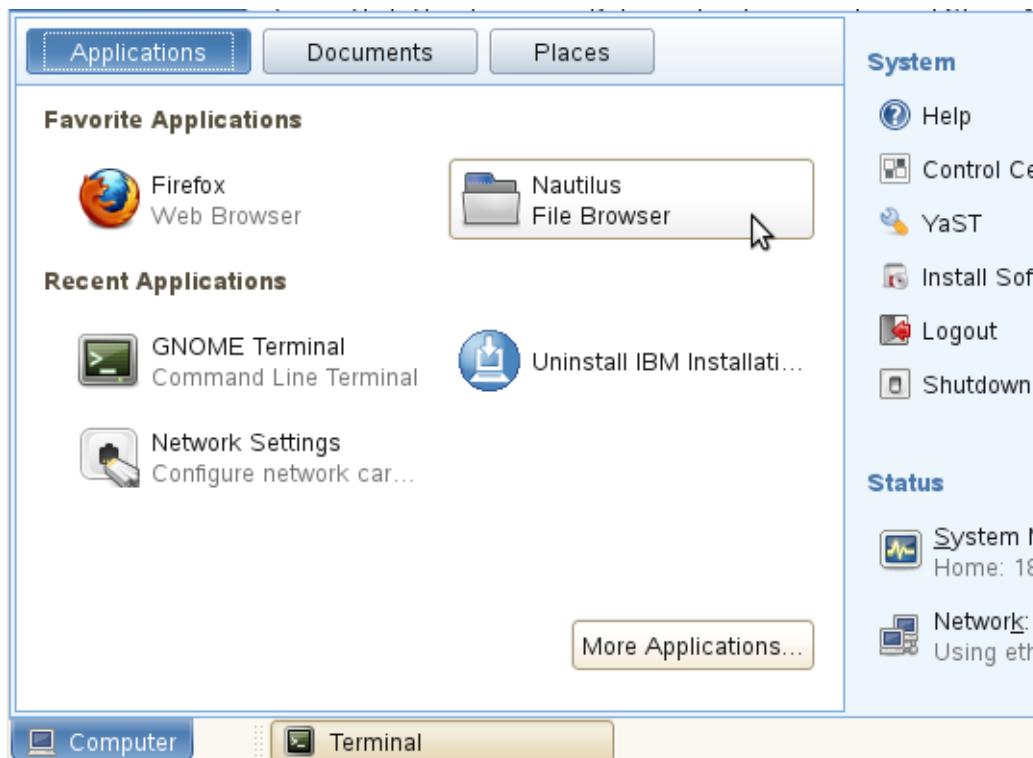
Any configuration changes that you want to keep must be saved to the master configuration; otherwise the new settings are not used.

- 1. Click **Logout** at the top of the page in the taskbar.
- 2. If no resources are changed, then you are returned to the login page. If changes were made, the Save page is seen. Click **Discard** so that you do not overwrite the configuration.  
The Discard WorkSpace Changes page asks you to confirm the discard. Click **Yes**, and you are returned to the login page.
- 3. Close the browser.

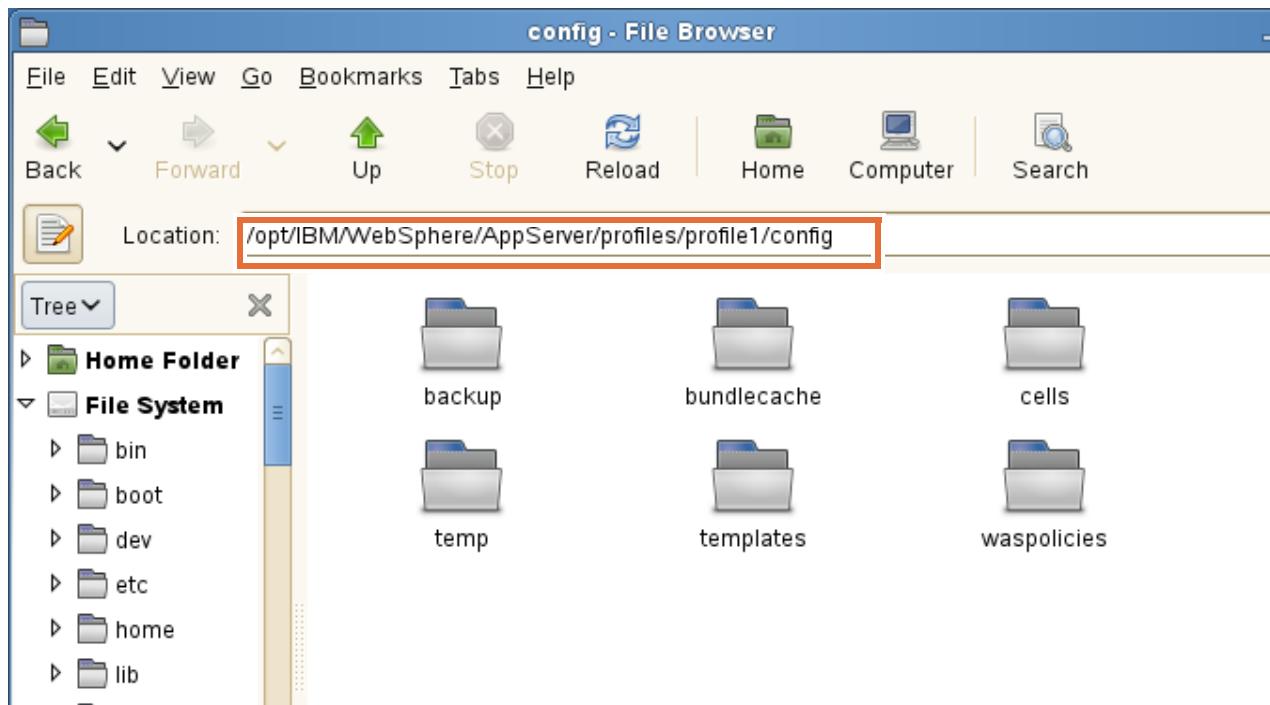
## Section 13:Explore configuration files

Examine some of the configuration files for the WebSphere Application Server.

- 1. Explore the configuration directory structure and view some of the configuration files.
  - a. To explore files on Linux, use Nautilus: **Computer > Applications > Nautilus File Browser.**



- \_\_ b. Examine the files within the profile config directory, which can be found under /opt/IBM/WebSphere/AppServer/profiles/profile1.



\_\_ c. A few important files that are contained in the config directory are:

- <cell> is the cell name
  - <node> is the node name
  - <server> is the server name
- Cell-wide resources  
config/cells/<cell>/resources.xml
  - Node-specific resources  
config/cells/<cell>/nodes/<node>/resources.xml
  - Server-specific resources such as JDBC and JMS providers  
config/cells/<cell>/nodes/<node>/servers/<server>/resources.xml
  - Global security settings  
config/cells/<cell>/security.xml
  - Virtual hosts  
config/cells/<cell>/virtualhosts.xml
  - Applications and endpoints for a node  
config/cells/<cell>/nodes/<node>/serverindex.xml
  - Configuration of a server  
config/cells/<cell>/nodes/<node>/servers/<server>/server.xml



### Information

You must not edit these XML files manually; instead you must use the administrative console or wsadmin command-line tool to make configuration changes that affect these files.

\_\_ 2. Examine the SOAP client configuration file:

<profile\_root>/profile1/properties/soap.client.props

- \_\_ a. From a terminal window, navigate to: <profile\_root>/profile1/properties
- \_\_ b. Open soap.client.props in an editor (such as gedit).

This file contains security configuration information that clients use to authenticate to the security service. The wsadmin client uses this file. Important parameters are:

- loginUserId and loginPassword: ID and password are specified when the parameter loginSource=properties is used. If they are not set here or on the command line, the user is prompted for them interactively.

- `loginSource`: specifies how the authentication information is obtained. The default is `prompt`, which means the user is prompted for a user ID and password.

```
#-----
com.ibm.SOAP.authenticationTarget=BasicAuth

com.ibm.SOAP.loginUserId=
com.ibm.SOAP.loginPassword=
```

#-----  
# SOAP Login Prompt  
#  
# The auto prompting will happen only if all of the following  
#  
# - Running from a SOAP client  
# - Server is reachable and server security is enabled  
# - Username and password are not provided either on command l  
# file  
# - com.ibm.SOAP.loginSource below is set to one of the follow  
#  
# + stdin: Prompt in command window  
# + prompt: GUI dialog box; falls back to stdin if GUI not a  
# + krb5Ccache: (valid only when authenticationTarget=KRB5)  
# data from a cache file (see below)  
#  
# (So to disable auto prompting, set loginSource to nothing)  
#-----

```
com.ibm.SOAP.loginSource=prompt
```



3. **Close** any editor windows still open.

## End of exercise

## Exercise review and wrap-up

This exercise examined many of the features of the administrative console. You looked at the properties of servers, applications, environment settings, and resources.

# Exercise 5. Assembling an application

## What this exercise is about

This exercise covers the steps necessary to assemble Java archive (.jar) files, and web archive (.war) files that combine an enterprise application into an enterprise archive (EAR) file. This file can be deployed to a Java EE 6 compliant application server. IBM Assembly and Deploy Tools for WebSphere Administration are used to complete this exercise.

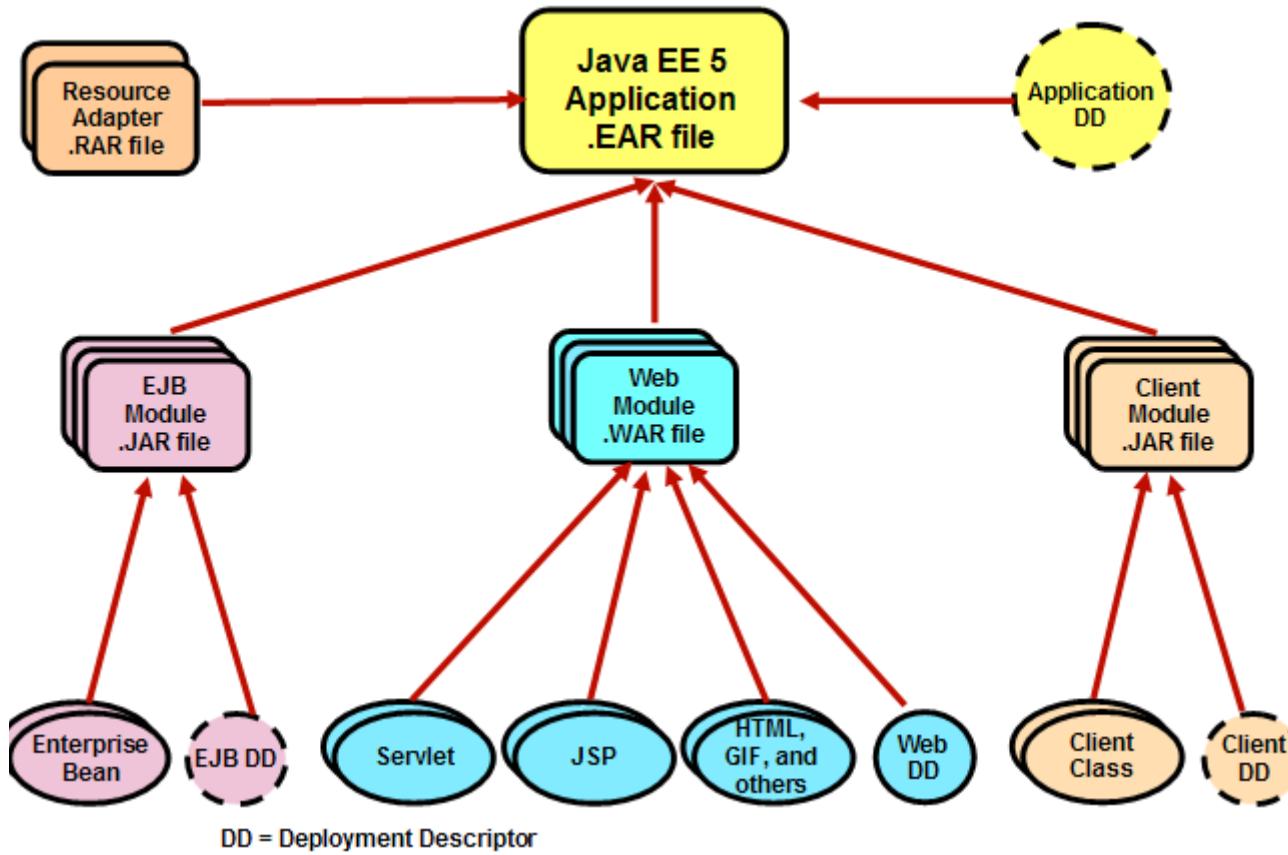
## What you should be able to do

At the end of this exercise, you should be able to:

- Explore the Assembly and Deploy tool
- Import and examine enterprise application components
- Define application-scoped resources: data source and authentication alias
- Export an enhanced EAR file that is ready for deployment

## Introduction

In this exercise, you assume the role of the application assembler. The developers responsible for bean development provided you with the .jar files that contain the code for the Enterprise JavaBeans. The developers responsible for the presentation design gave you the .war files that contain the HTML pages, JSPs, and servlets. It is now your task to take these pieces and assemble them into an EAR file that can be installed in the WebSphere Application Server.



The application developers provided the files for this application. The application consists of the following components.

- One Java utility.jar file. The application uses the code in this file.
- One web module .war file that contains the servlets, JavaServer Faces (JSF) files, Enterprise JavaBeans (EJB) files, and presentation (HTML and graphics) files, along with a deployment descriptor.

In some cases, you might also be given a resource archive (RAR) module. For this exercise, however, there are no resource archive modules.

## Requirements

To complete this exercise, you need the IBM Assembly and Deploy Tools for WebSphere Administration. This tool is used to complete the exercise. You also need the following files, which are in /usr/software/assemble:

- PlantsByWebSphere.war

- pbw-lib.jar

## Exercise instructions

### Section 1: Resetting the WebSphere environment

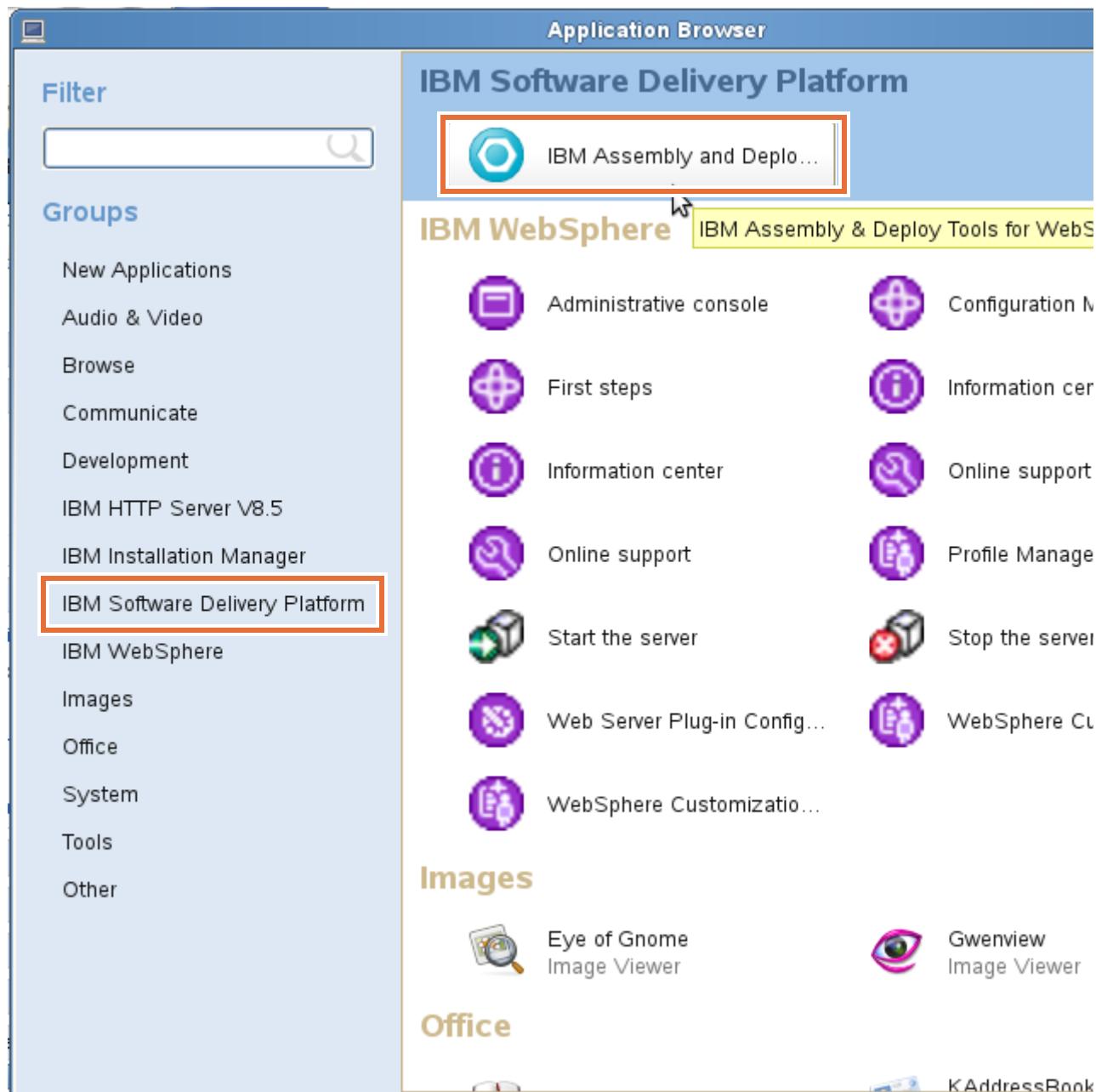


#### Note

To reset your WebSphere environment, read **Appendix A** for instructions on how to complete this procedure.

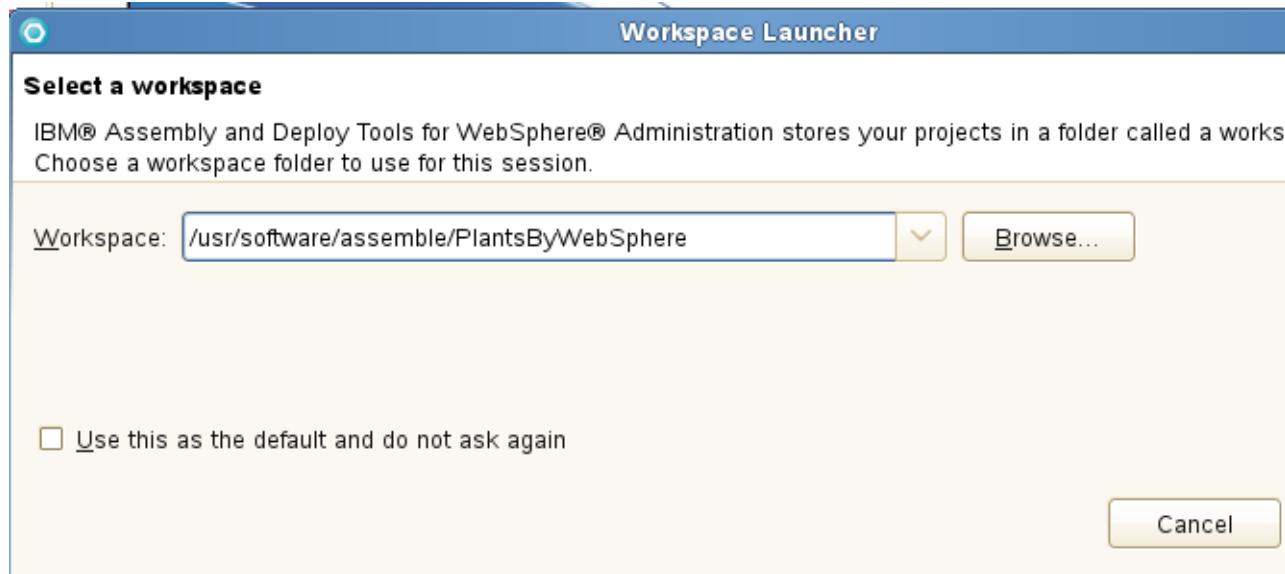
## Section 2: Start the IBM Assembly and Deploy Tools for WebSphere Administration

1. Click Computer > More Applications > IBM Software Delivery Platform > IBM Assembly and Deploy Tools for WebSphere Administration to start IBM Assembly and Deploy Tools.



IBM Assembly and Deploy Tools starts by asking you which directory to use for its workspace. This workspace is used during the life of the project. Each project can have its own separate workspace.

- 2. Enter /usr/software/assemble/PlantsByWebSphere for the **workspace** and click **OK**.



- 3. Wait several seconds for the workspace to load. When the workspace opens, you are in the **Java EE** perspective to begin assembling the application. This perspective is the default, and it is the title of the workbench.
- 4. The **Enterprise Explorer** view shows all the modules that are currently loaded in the workspace (none currently). As you add modules, they show in the corresponding folders.

The empty area on the upper middle part of the window is where the different editors open and show the contents of the selected items on the other views. This area is called the editor pane.

Below the editor pane is a multipurpose pane that contains several views. One of the views in the multipurpose pane is the **Markers** view, which is in the foreground of the multipurpose pane. It contains any outstanding errors that must be resolved. This space is shared with several stacked views. You can select a view by its tab at the top of the pane. These additional views are required during this exercise.

### **Section 3: Create an enterprise application project**

- 1. Create an enterprise application project named `PlantsByWebSphere`.
- a. Select the **File > New > Enterprise Application Project**.
- b. Name the project: `PlantsByWebSphere`



## Information

This new workspace has no servers that are defined. An installed server must be defined in the workspace so its runtime libraries are added to new projects created in the workspace. In addition, the selected **Target runtime** defines where applications are deployed to, when asking assembly and deploy to run an application on a server.

- c. Click **New Runtime** to the right of Target runtime to define a new server.

**New EAR Application Project**

**EAR Application Project**  
Create a EAR application.

Project name: PlantsByWebSphere

Project location

Use default location

Location: /usr/software/assemble/PlantsByWebSphere/PlantsBy

Target runtime

<None>

New Runtime...

EAR version

6.0

Configuration

Default Configuration

The default configuration provides a good starting point. Additional facets can later be installed to add new functionality to the project.

Working sets

Add project to working sets

Working sets:

?

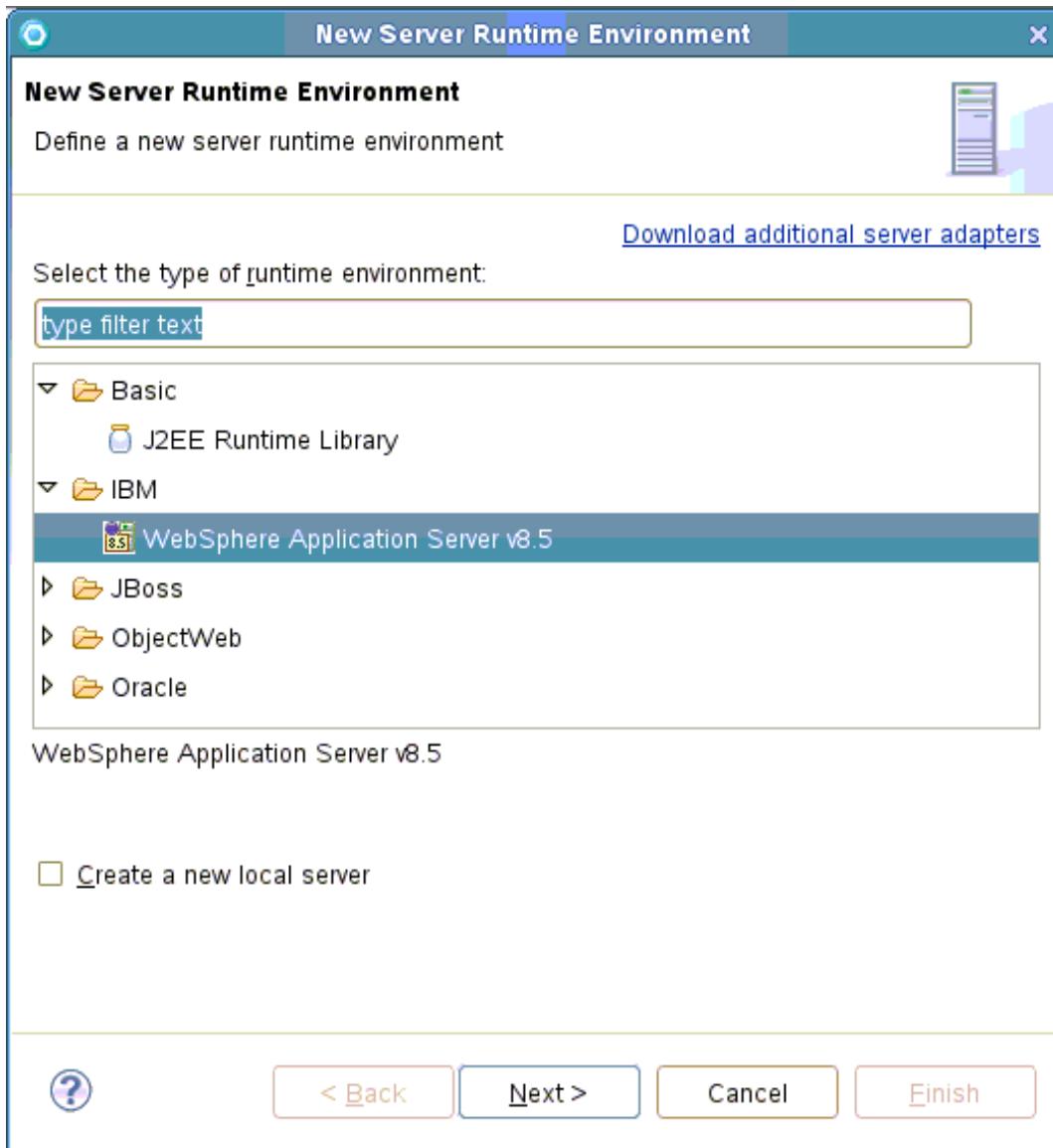
< Back

Next >

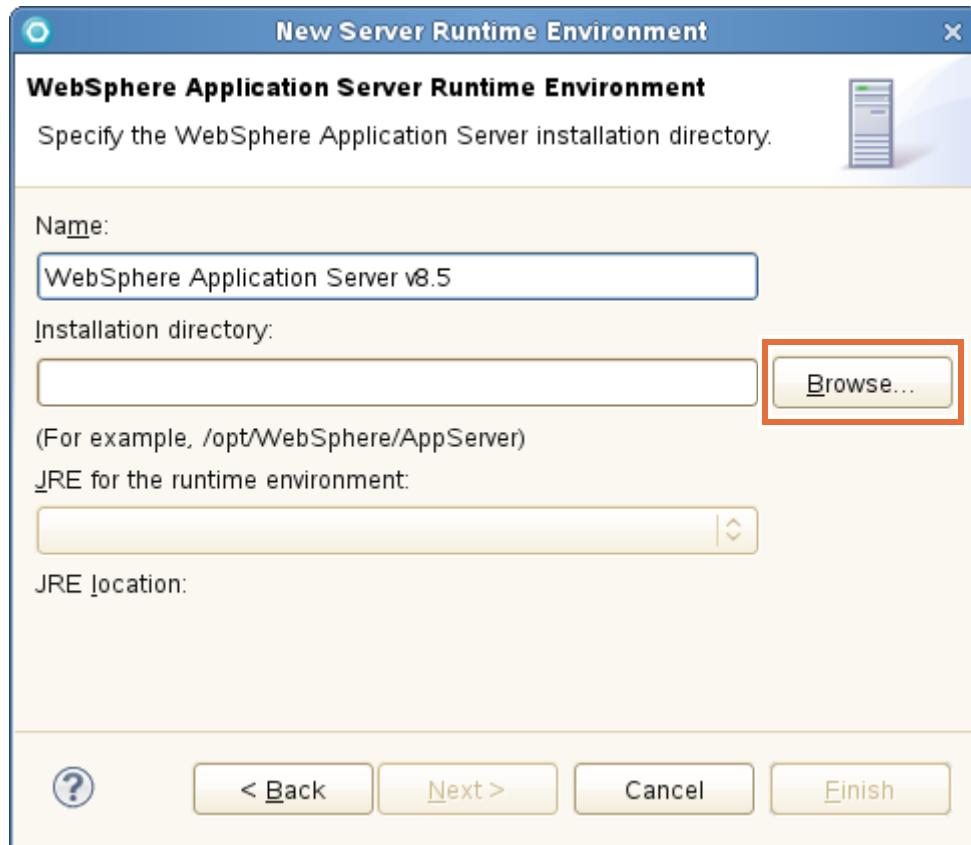
Cancel

Finish

- \_\_ d. Select **IBM > WebSphere Application Server v8.5**. Click **Next**.

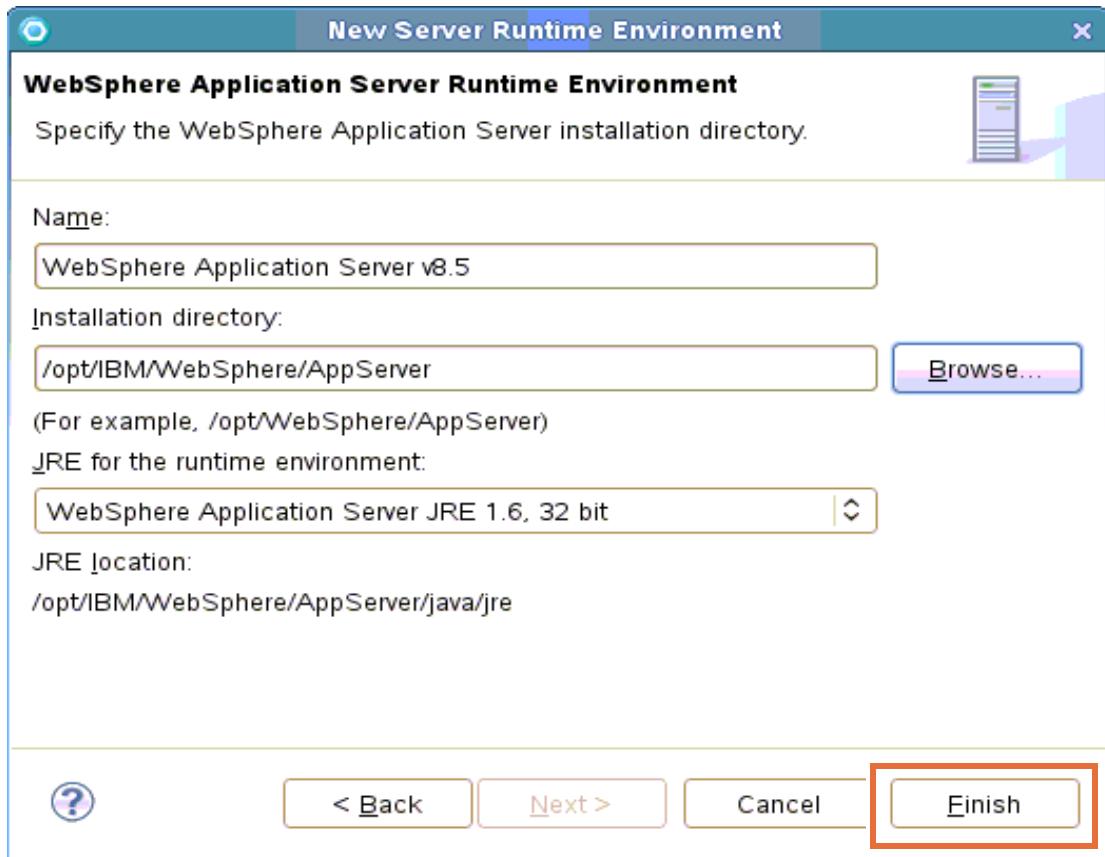


- \_\_\_ e. Click **Browse** for the Installation directory.

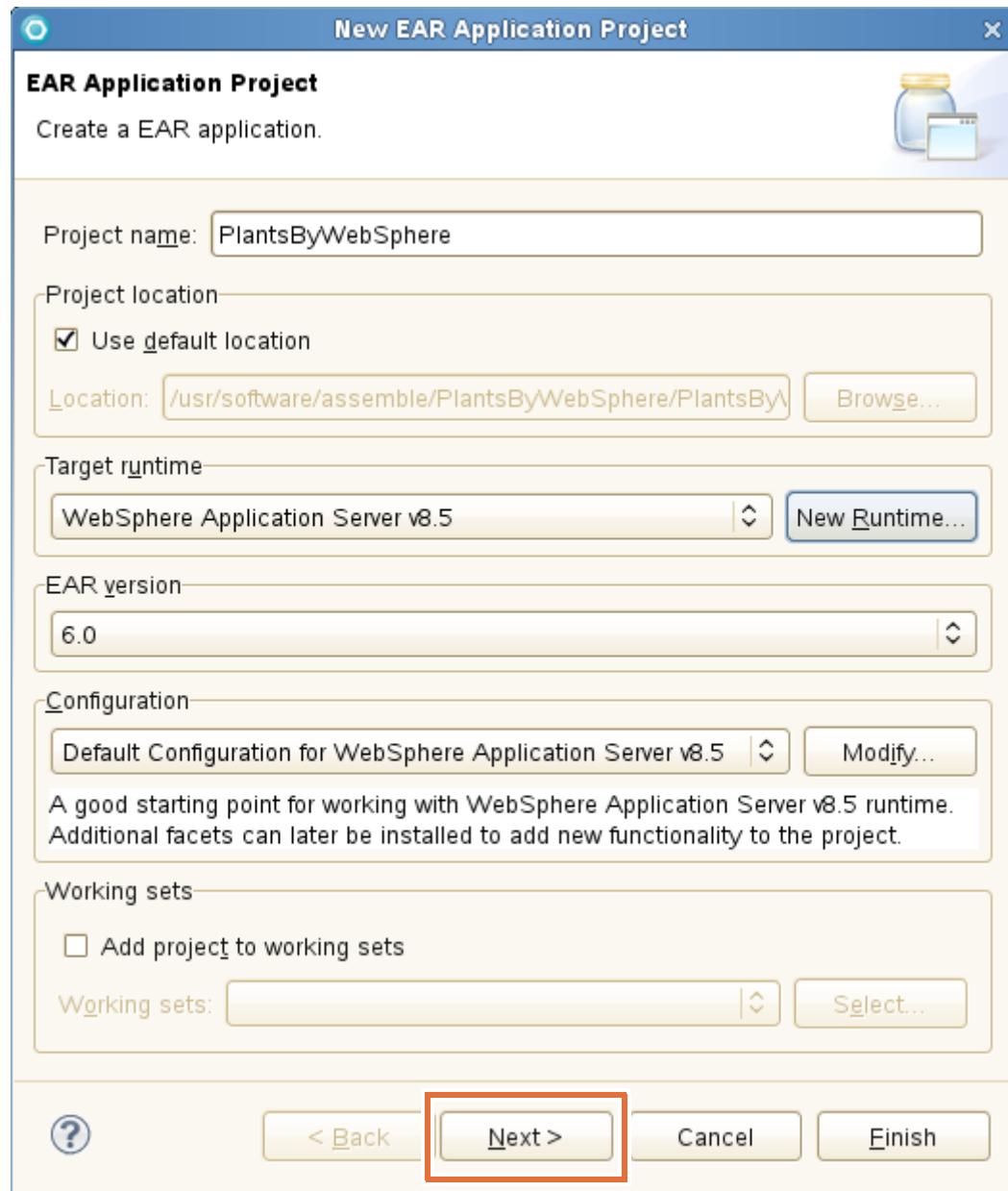


- \_\_\_ f. Go to the `<was_root>` folder and click **OK**.

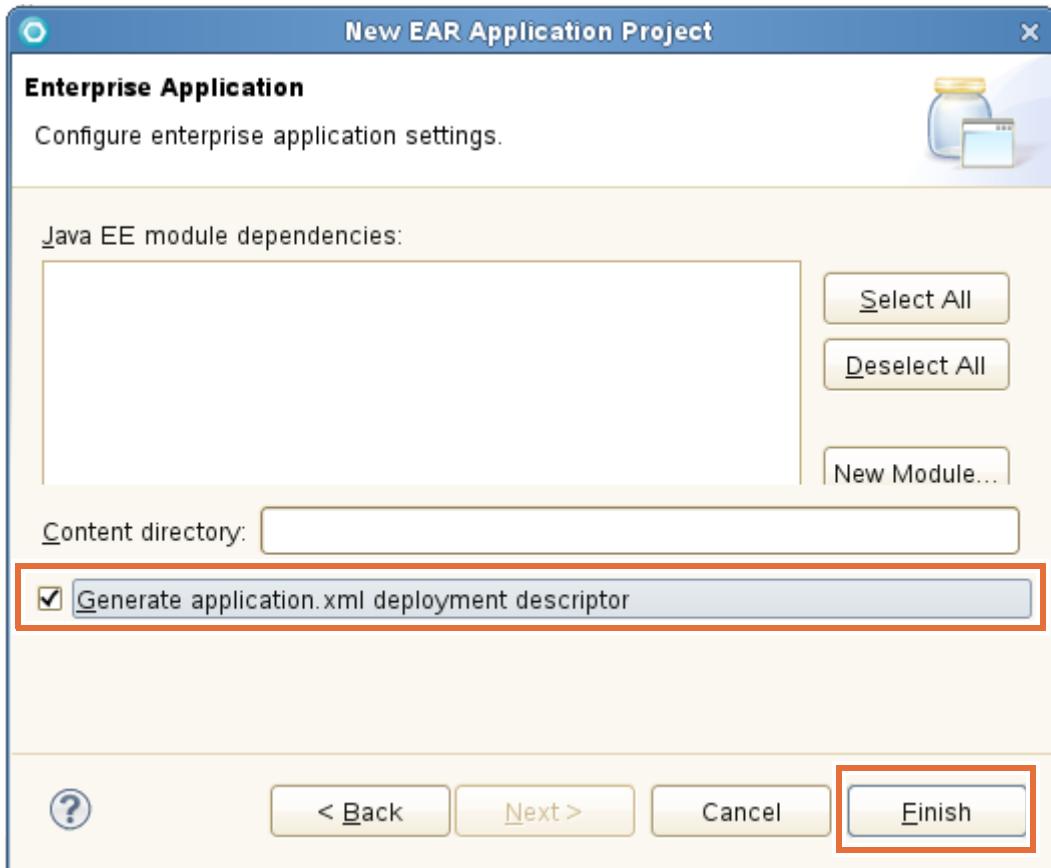
- g. Click **Finish**.



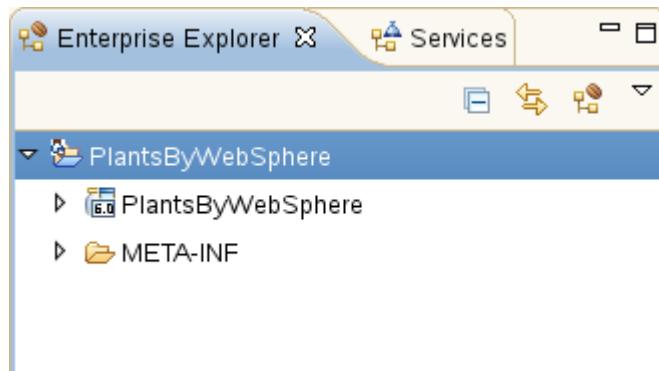
- h. Make sure that the Project name is still set to PlantsByWebSphere. Click **Next**.



- \_\_ i. Check **Generate application.xml deployment descriptor**. Click **Finish**.



- \_\_ j. Verify that the `PlantsByWebSphere` project is now found in the Enterprise Explorer.



## Import the application modules

In your case, you have a set of modules, .war and .jar files, by development. These modules must be assembled into a running application.



### Information

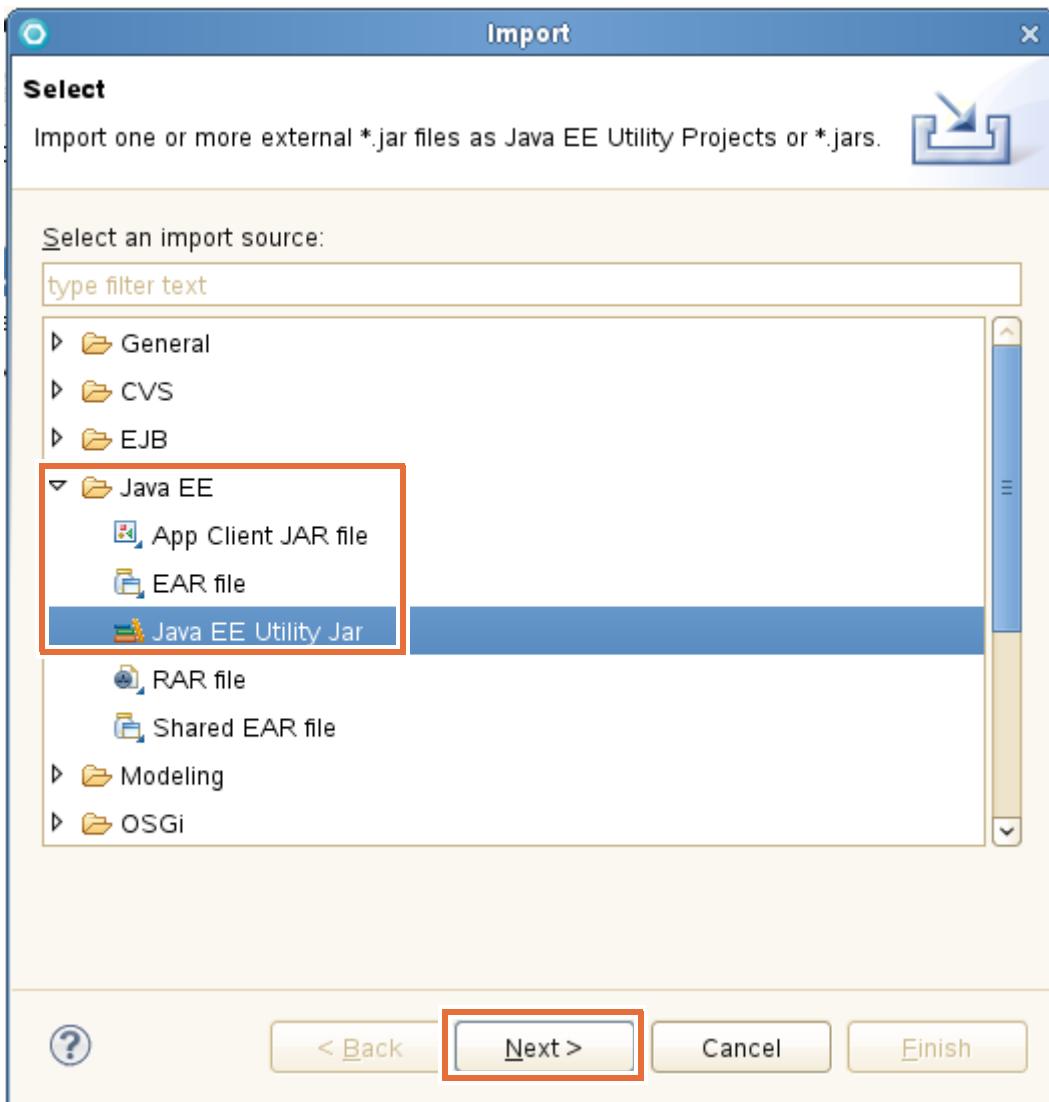
Handing over individual application modules is the most common way to receive application components, especially when more than one development team is involved. Each team is responsible for one or more modules. Another way to hand over an application for deployment is to receive an EAR file for the enterprise application.

### ***Section 4: Add the Plants By WebSphere utility module***

The application developers provide the utility .jar file for this application, and it is named pbw-lib.jar. The JAR file contains the reusable code for multiple applications.

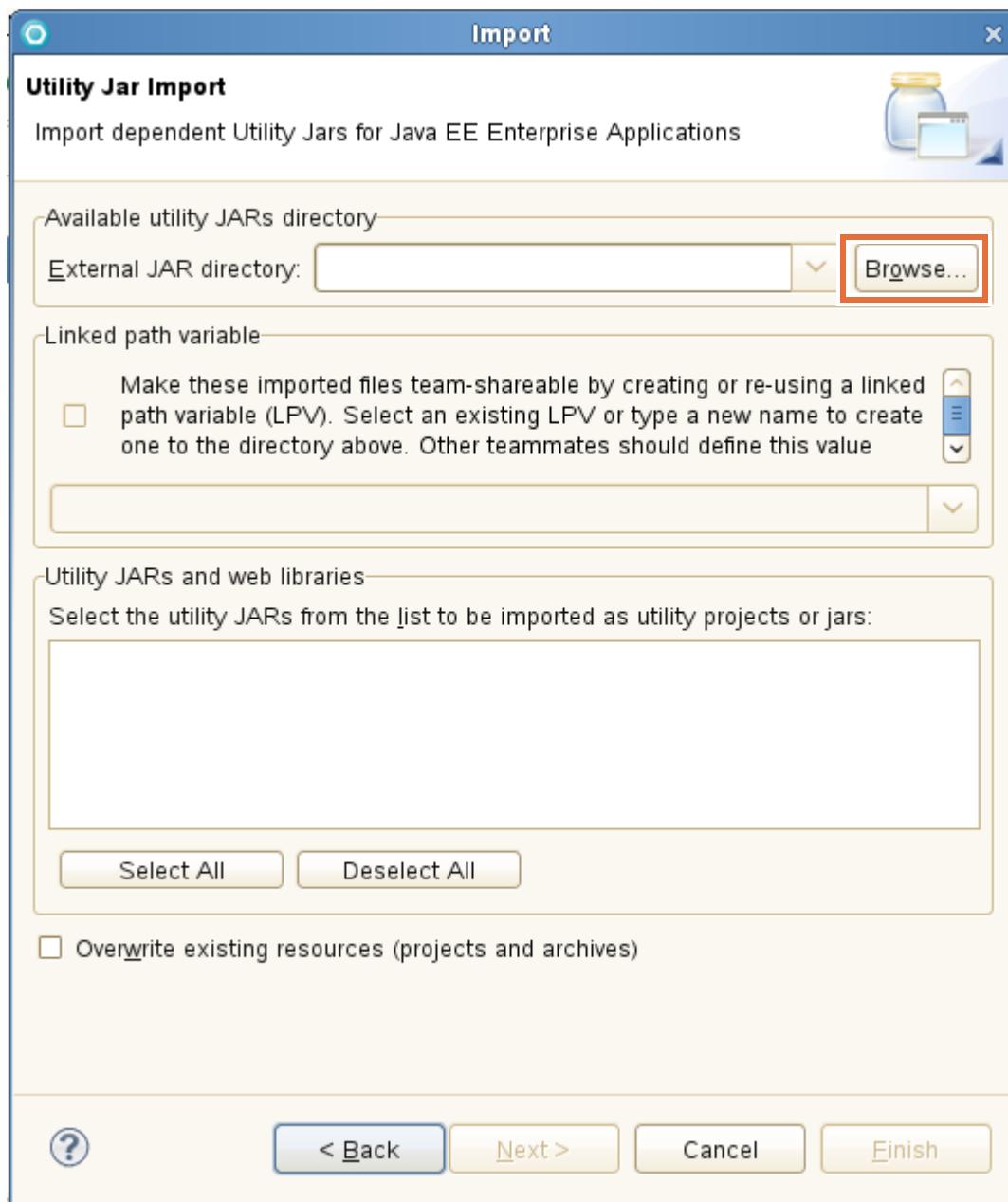
- 1. From the menu, select **File > Import**.

2. From the **Import** dialog, expand the **Java EE** folder, select **Java EE Utility Jar**, and click **Next**.



3. Verify that the EAR project is **PlantsByWebSphere**, and check **Copy utility JARs into an existing EAR from an external location**. Click **Next**.

4. In the **Import** dialog, click **Browse**, navigate to /usr/software/assemble, and click **OK**.



5. Check the utility JAR file named pbw-lib.jar. Click **Finish**.



### Information

Double-clicking a module (the second entry) on the Enterprise Explorer view opens its deployment descriptor (if it exists) in a specialized editor. Deployment descriptor editors have tabs along the bottom of the pane to give you access to the various sections of the file. Using these editors makes working with deployment descriptors much easier.

If you change anything on a deployment descriptor, or any other file, you see an asterisk on the tab where its name is shown at the top of the editor pane. The asterisk indicates that the file has changes and must be saved. Do not save any changes that you make, and close the EJB deployment descriptor in the editor.

## **Section 5: Add the web module**

Next, add the web module that the application references. A web module consists of the JSPs, HTML pages, and servlets that are contained within the `.war` file.

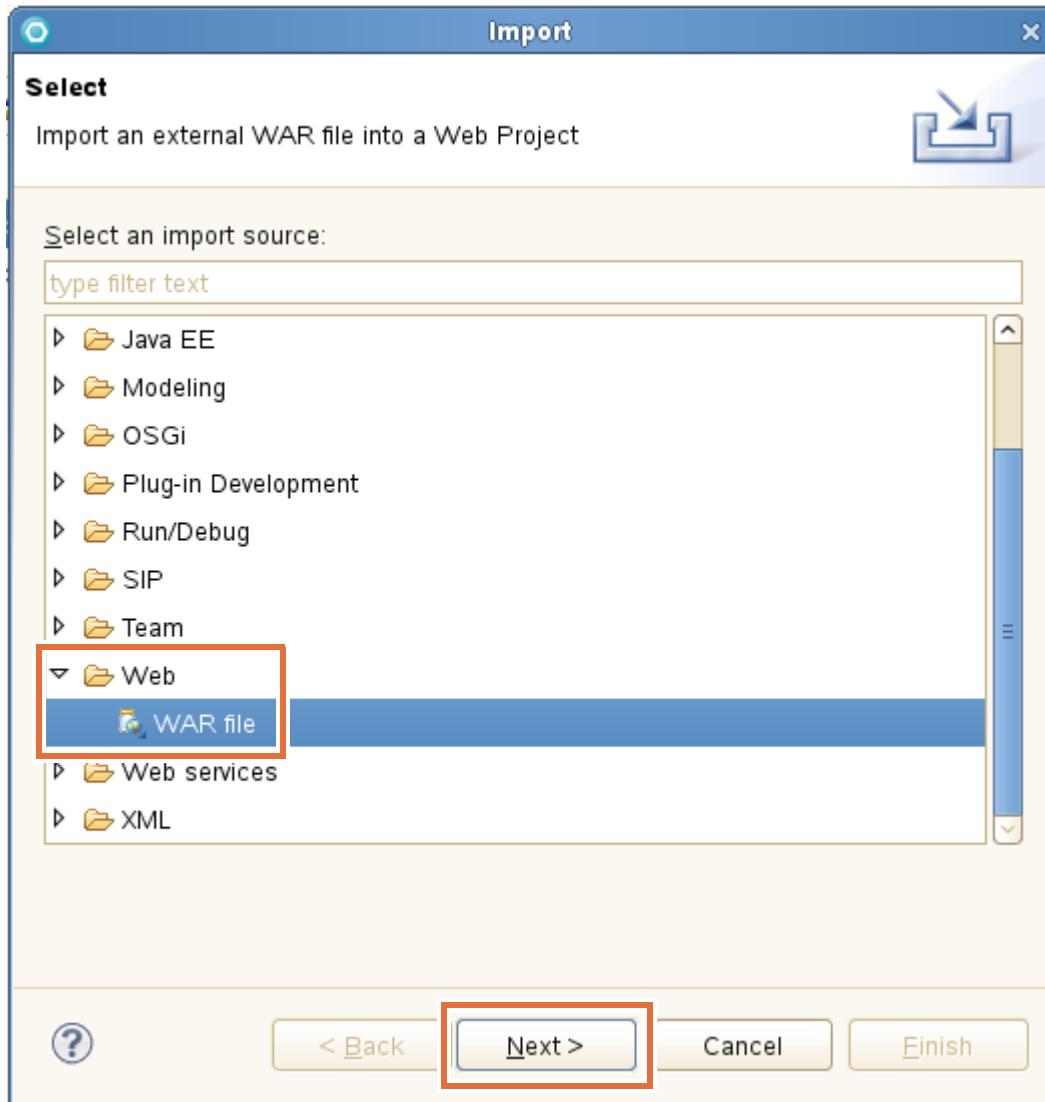


### Note

In addition, this `.war` file contains EJBs. One of the changes in the Java EE 6 specification is that EJBs can be put into a web archive file.

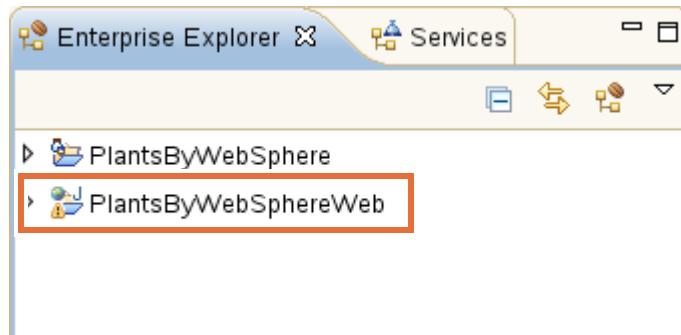
- 
- 1. From the menu, select **File > Import**.

- 2. In the **Import** dialog, expand the Web folder and select **WAR file**.



- 3. Click **Next**.
- 4. In the **Import** dialog, click **Browse**, go to /usr/software/assemble, and select **PlantsByWebSphere.war**. Click **OK**.
- 5. Change the web project name to: PlantsByWebSphereWeb  
The enterprise application project already has the name **PlantsByWebSphere**.
- a. Click **Finish** to add the PlantsByWebSphere web module to the enterprise application.
- b. Click **No** in the dialog that asks to change to the Web perspective.

- \_\_ c. Verify that the web module is in the Enterprise Explorer.



- \_\_ d. Look for errors or warnings in the **Markers** view.

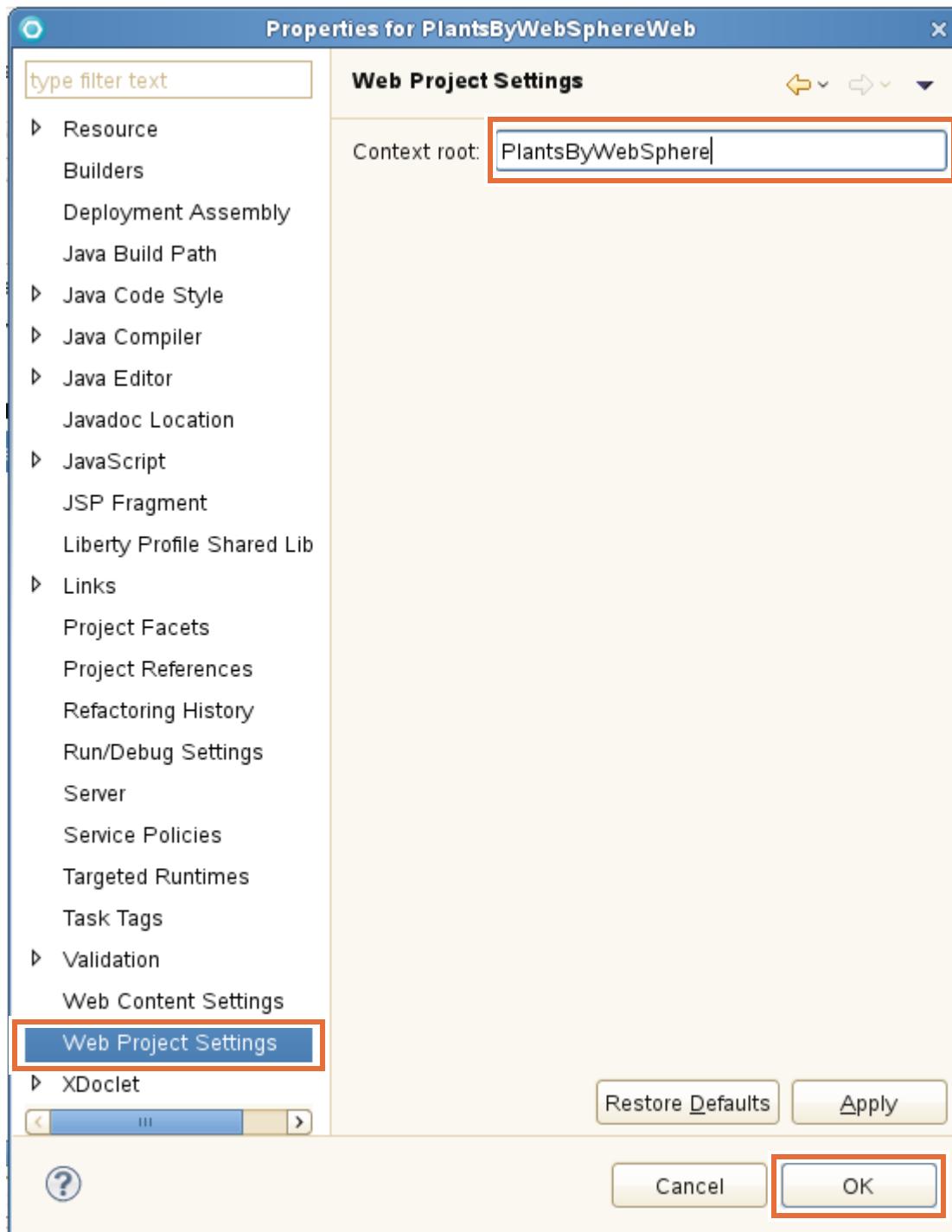


### Information

It takes a minute or two for the project to finish rebuilding. Ignore any errors until the rebuild process is complete.

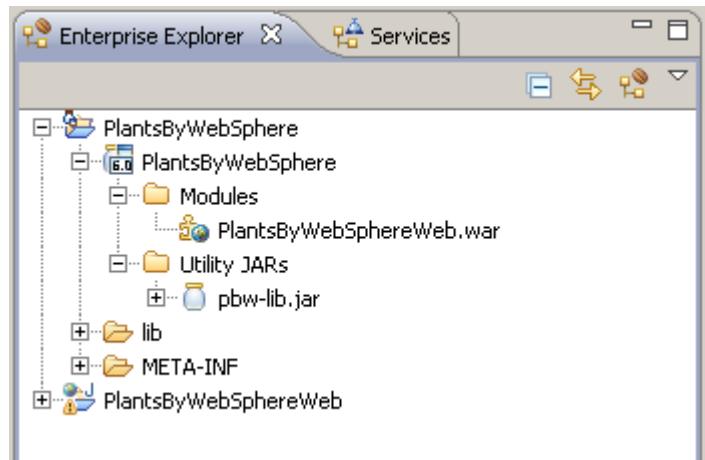
- \_\_ 6. The **PlantsByWebSphere** application must have the Uniform Resource Identifier (URI) modified.
- \_\_ a. In the Enterprise Explorer view, select **PlantsByWebSphereWeb**, right-click, and select **Properties** from the menu.

- \_\_ b. Select **Web Project Settings**. Change the Context root to: PlantsByWebSphere  
Click **OK**.



- \_\_ c. A dialog box might open to confirm this change. Click **OK**.

- \_\_\_ d. The web and utility modules are added to PlantsByWebSphere. You can see these modules in the Enterprise Explorer by expanding **PlantsByWebSphere > PlantsByWebSphere > Modules**.



### Information

#### Warnings in the Markers view

You are going to see several warnings such as JSP problems, XML problems, and possibly others. It is always a good practice to investigate any warnings you see in the Markers view. However, for the purposes of this exercise, you can ignore these warnings.

The types of warnings that are displayed can be configured by setting the preference feature to ignore certain types of warnings.

For example, if you see the XML warning: No grammar constraints (DTD or XML schema) detected for the document, you can remove the warning by clicking **Window > Preferences > XML > XML Files > Validation**. Change the “**Indicate when no grammar is specified**” option to “**Ignore**”. Next, click **Project > Clean**.

## Section 6: Add a test server

Earlier you specified that the target runtime environment is WebSphere Application Server V8.5. Now you add a WebSphere Application Server V8.5 that can be used to test the application if one does not exist.

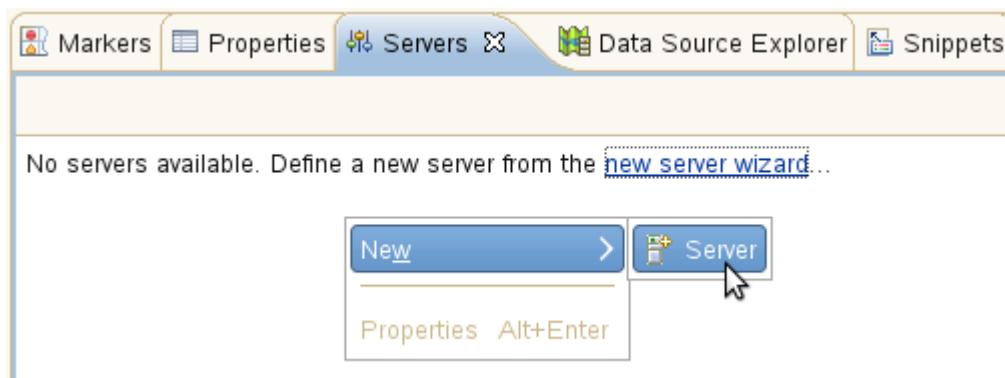


## Information

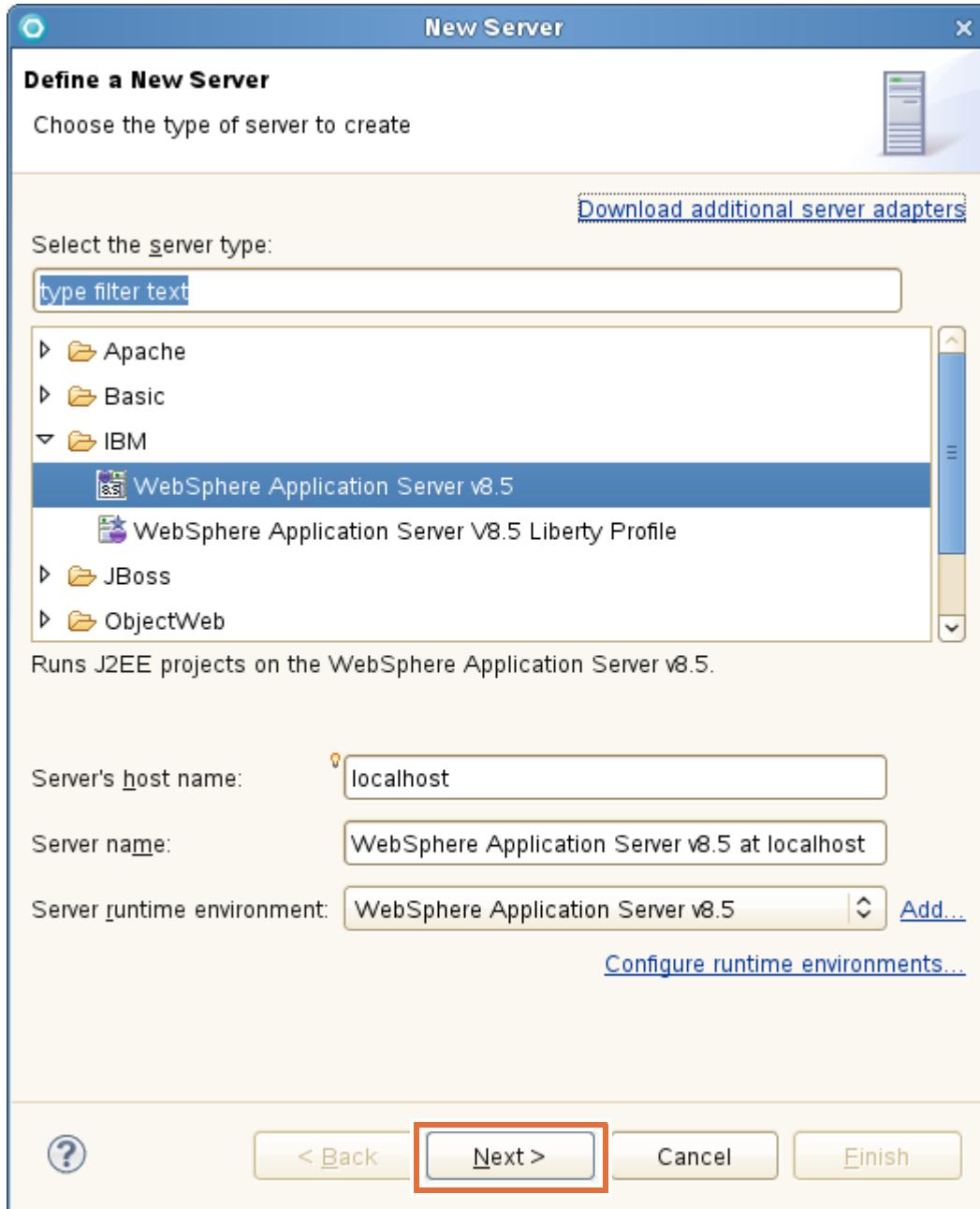
Adding a test server within the IBM Assembly and Deploy Tools allows administrators, assemblers, or developers to test their enterprise applications directly from their development environment. The alternative would be to export the EAR and deploy it to a WebSphere Application Server runtime environment.

Although this test environment is being defined here, it is not used in this exercise.

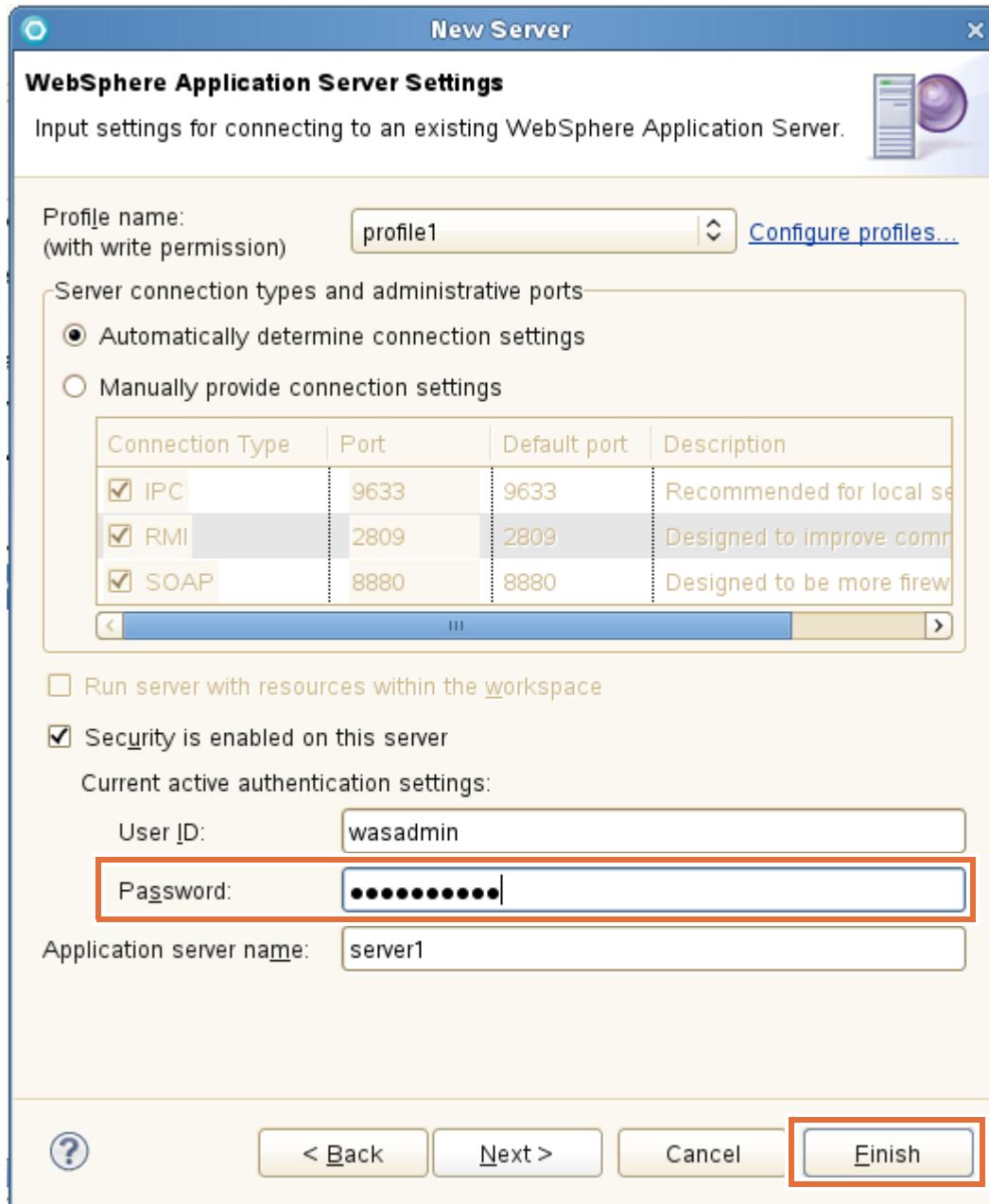
1. Select the **Servers** tab. Right-click in the empty Servers view; then select **New > Server** from the menu.



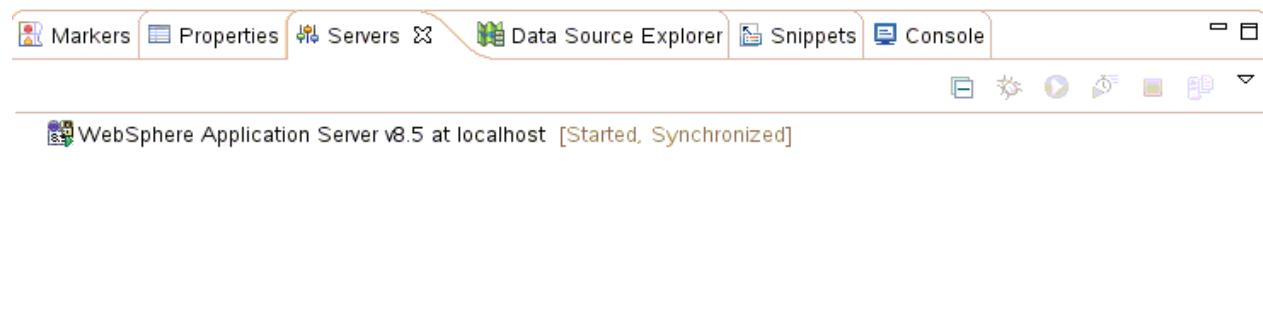
2. In the **Define a New Server** dialog, accept the default options. Click **Next**.



3. Type the password `websphere` in the WebSphere Server Settings dialog. Click **Finish**.



4. WebSphere Application Server v8.5 is shown in the Servers view.





### Information

Creating the WebSphere Application Server V8.5 in the Servers view allows you to test your application on an existing instance of the WebSphere runtime environment directly from the IBM Assembly and Deploy Tools. You can easily install enterprise applications from the menu by right-clicking **WebSphere Application Server v8.5** and selecting **Add and Remove Projects** from the menu.

You are not going to run through a test at this stage. Some additional configuration of the environment must be completed. You are going to test the Plants application in a later exercise.

## Section 7: Configure WebSphere data sources

You can define certain resources that are included with the application in the WebSphere Application Server Deployment editor. Any resources that are defined on this page are defined at the application scope. This approach is valuable in a development or test environment, but is not considered a good approach when releasing applications into a production environment.

When you export this EAR, the resource definitions are included. An EAR file that contains these types of resource definitions is named an enhanced EAR. The resources that are defined within the EAR are scoped at the application scope level. These resources take precedence over any other resource definitions within the existing runtime environment.

Although the steps that follow create application scoped resources, they are not used in the labs that follow. This section is here to demonstrate how to add these resources.

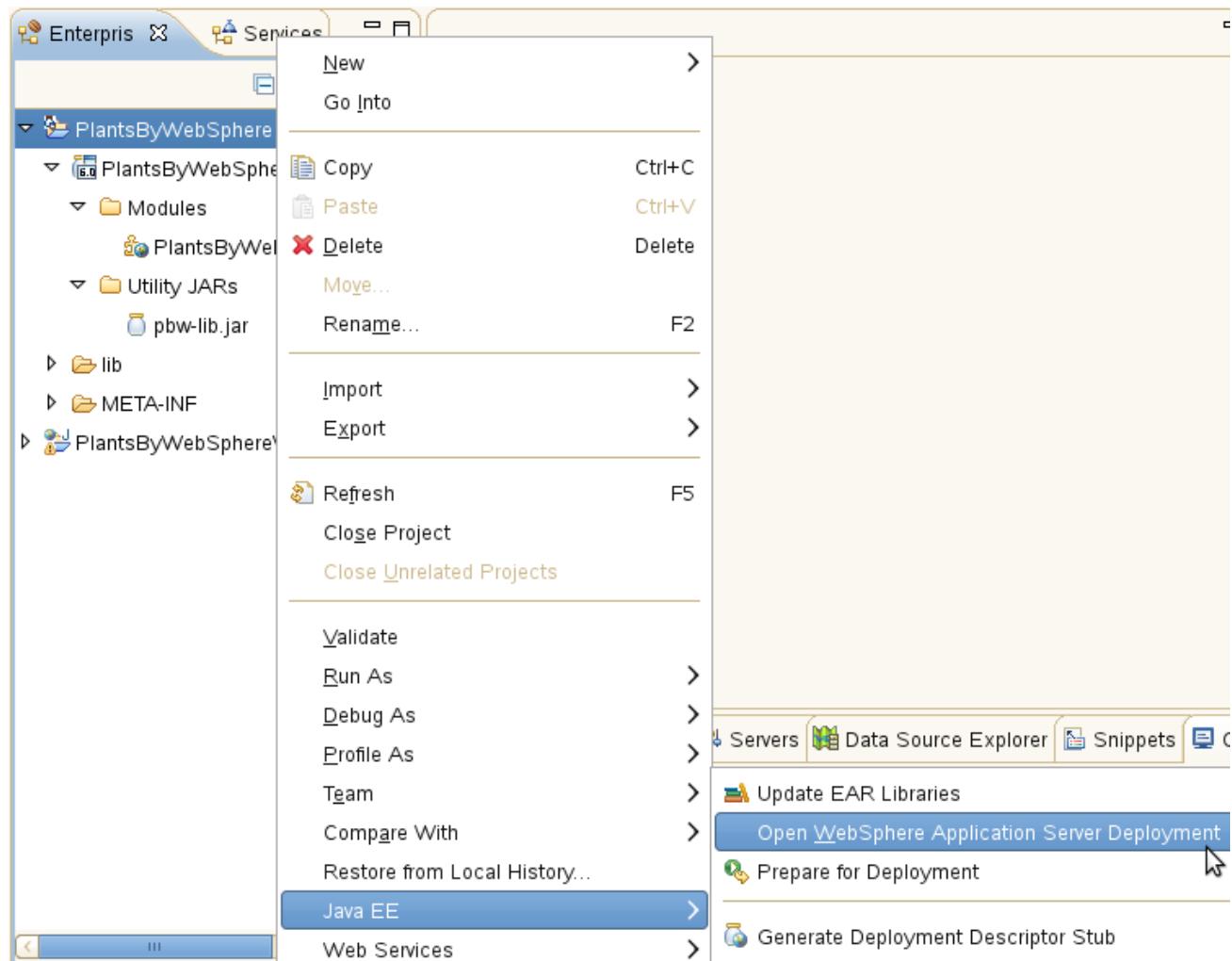


### Information

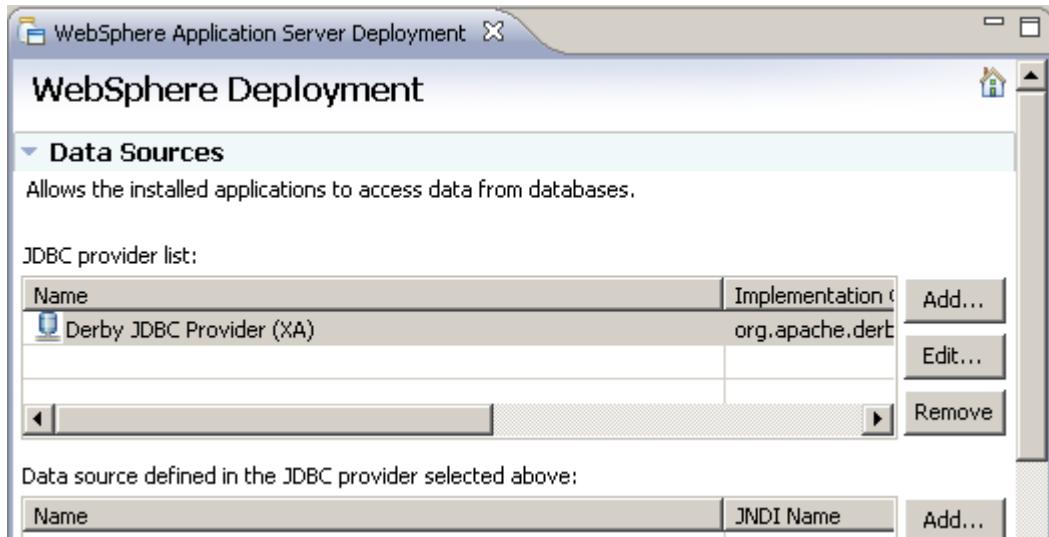
To work with application scoped resources in the WebSphere administrative console, you must select the application and then under **Resources**, select **Application scoped resources**.

In this section, you define a JDBC provider and data source for the PlantsByWebSphere application. Both of these resources are defined at the application scope and visible only to the application.

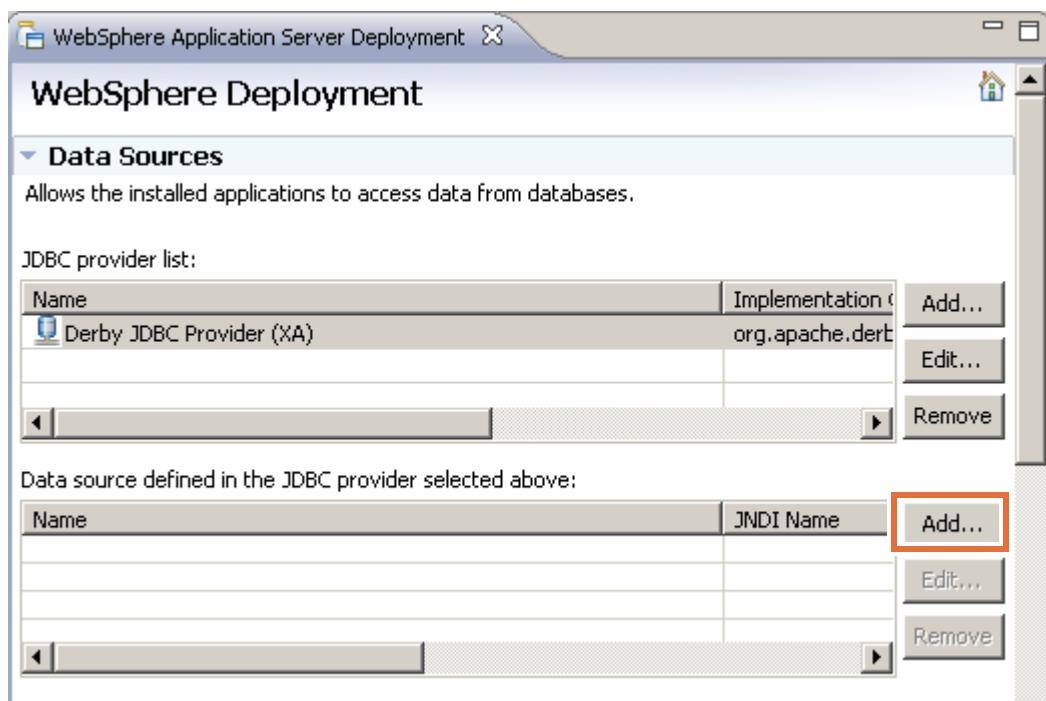
- 1. Open the Server Deployment view.
  - a. Right-click **PlantsByWebSphere**; then select **Java EE > Open WebSphere Application Server Deployment**.



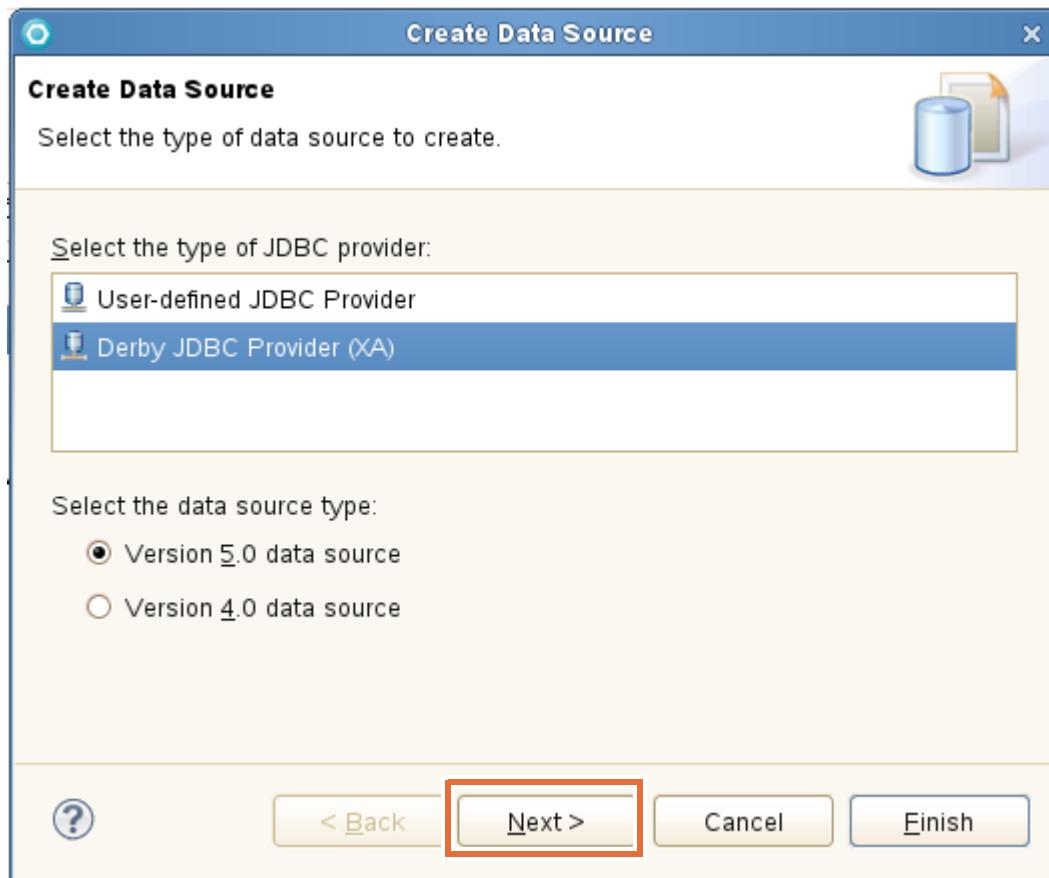
- 2. Review the existing Derby JDBC provider. Derby is the database server that the enhanced EAR file uses.
  - a. At the top of the WebSphere Application Server Deployment view, notice the **Derby JDBC Provider (XA)** in the JDBC provider list. You can look at this configuration by clicking **Edit**.



- 3. Create the data source. The data source is where you can define the specifics about the back-end resource that is used. In this case, it provides the details for accessing the PLANTS database (for example: data source name, JNDI name, J2C authentication alias, database name, database host name).
  - a. Scroll to the top of the editor. Select the **Derby JDBC Provider (XA)** entry. Click **Add** next to the **Data source defined** list.



- \_\_\_ b. In the Create Data Source window, select **Derby JDBC Provider (XA)** as the JDBC provider type.



- \_\_\_ c. Click **Next**.
- \_\_\_ d. On the next page, enter the properties of the data source. Enter the following information in the fields as provided. Leave default values for all other fields.

**Table 1: Data source data**

| Field name                                                   | Value                            |
|--------------------------------------------------------------|----------------------------------|
| Name                                                         | Plants                           |
| JNDI name                                                    | jdbc/PlantsByWebSphereDataSource |
| Description                                                  | Plants data source               |
| Use this data source in container-managed persistence (CMP). | The box is checked.              |

**Create Data Source**

Select the type of data source to create.



Name:

JNDI name:

Description:

Category:

Statement cache size:

Data source helper class name:

Connection timeout:

Maximum connections:

Minimum connections:

Reap time:

Unused timeout:

Aged timeout:

Purge policy:

Component-managed authentication alias:

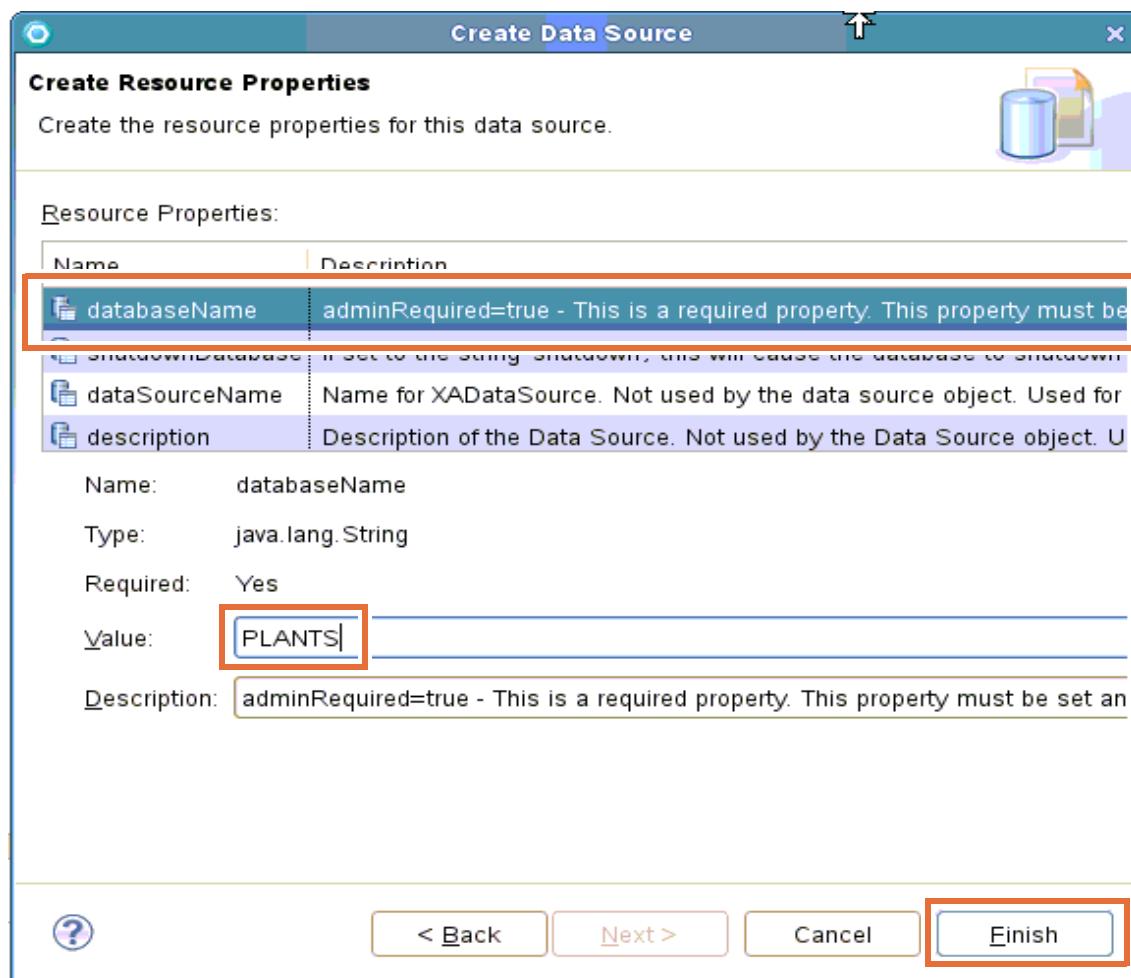
Container-managed authentication alias:

Use this data source in container managed persistence (CMP)

\* Required field.

\_\_ e. Click **Next**.

- \_\_\_ f. For a data source, multiple properties must be defined to access a database. The fields can be set by clicking the field name and then entering the value at the bottom of the wizard screen. For the Derby database, the only field that must be set is the **databaseName**. Set the **databaseName** field to: PLANTS



- \_\_\_ g. Click **Finish**.
- \_\_\_ 4. **Save** your changes. From the main menu, select **File > Save** (or press Ctrl-S). Close the **WebSphere Application Server Deployment** tab.



## Information

These resource settings are saved within the EAR file in files under the META-INF\ibmconfig folder. These files are not part of the Java EE 6 specification; instead, WebSphere Application Server V8 uses these files for attributes that the specification does not provide. This type of enterprise archive is called an enhanced EAR file.

The screenshot shows the IBM Rational Application Developer interface. On the left, the Enterprise Explorer view displays the project structure of 'PlantsByWebSphere'. It includes a 'Modules' folder containing 'PlantsByWebSphereWeb.war', a 'Utility JARs' folder containing 'pbw-lib.jar', a 'lib' folder, and a 'META-INF' folder. Within 'META-INF', there is an 'ibmconfig' folder which contains a 'cells' folder. Inside 'cells' is a 'defaultCell' folder, which further contains 'applications', 'nodes', and 'security.xml'. Under 'applications', there is a 'defaultApp' folder with 'deployments' and 'variables.xml'. Under 'deployments', there is another 'defaultApp' folder with 'deployment.xml' and 'resources.xml'. The 'resources.xml' file is selected in the editor on the right. The editor has two tabs: 'Node' and 'Content'. The 'Content' tab shows the XML structure of the 'resources.xml' file, including nodes like 'xml', 'xmi:XMI', and 'resources.jdbc:JDBCProvider'. The 'Content' tab also lists various properties such as 'version', 'xmi:version', 'xml:version', 'xmi:id', 'name', 'description', 'providerType', 'implementationClassName', 'xa', 'classpath', and 'factories'.

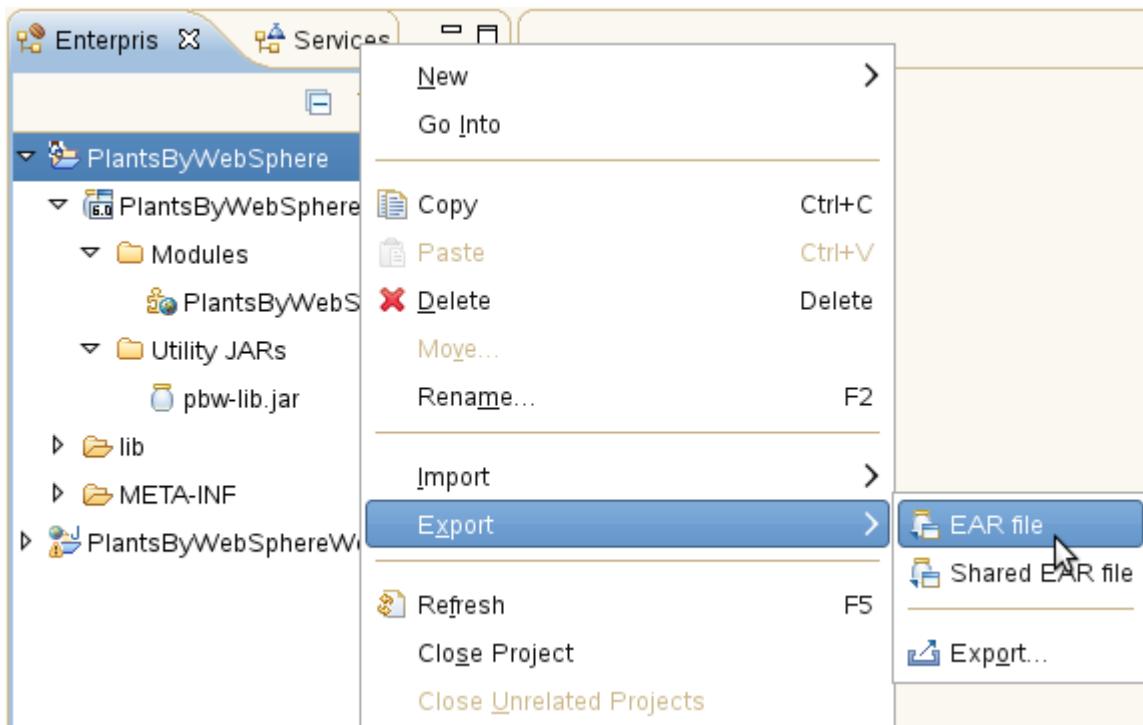
Look at the data that is contained in the `resources.xml` file. Typically this information is defined in the server and is required before the enterprise application is run. In the enhanced EAR file, this configuration data is stored with the application. These enhancements are useful for development and testing, but developers do not use them beyond testing. The enhanced EAR runs contrary to the separation of roles in Java EE.

Close the `resources.xml` file and do not save any changes.

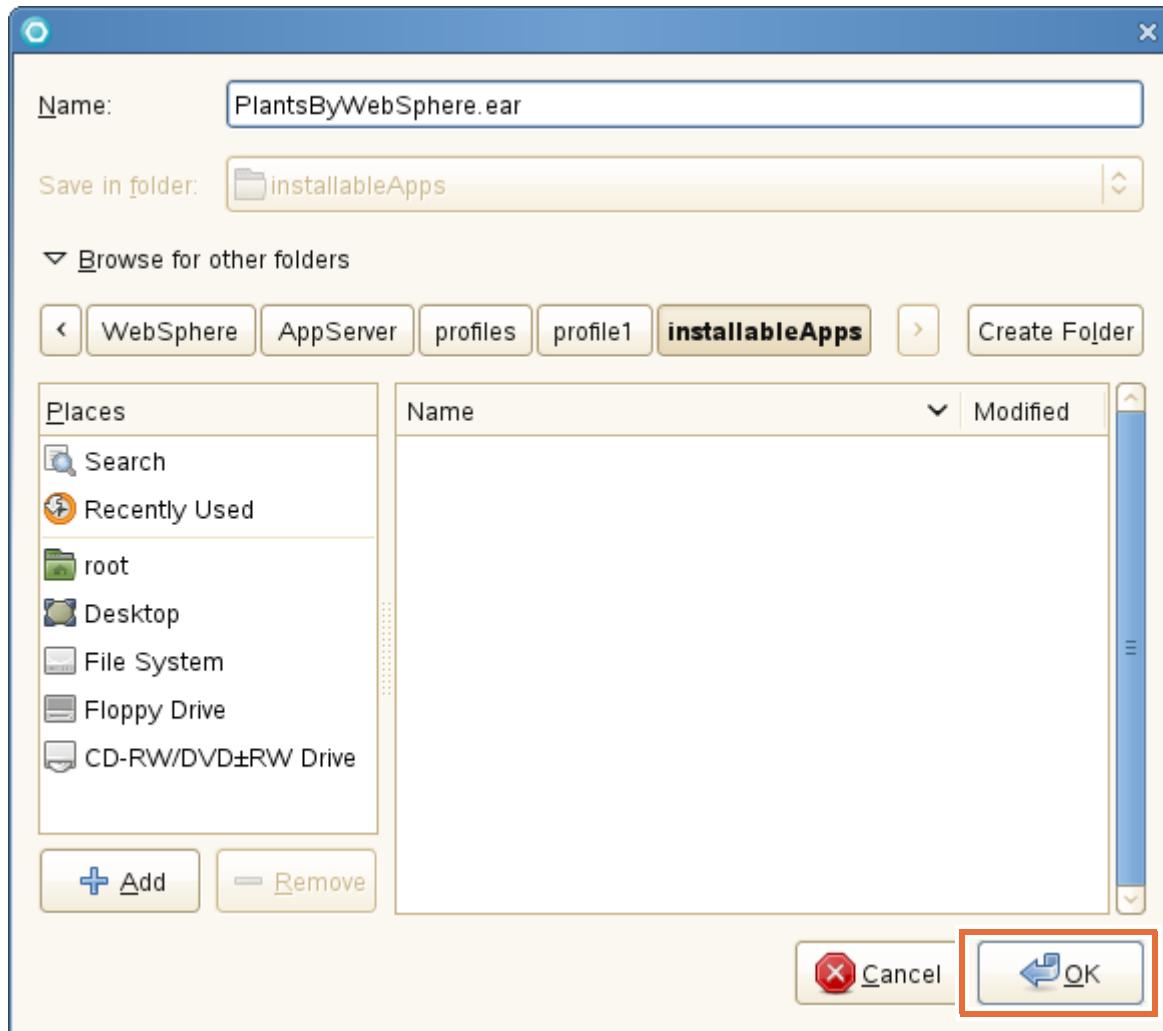
## Section 8: Export the enterprise archive (EAR) file

Save the file in the `<profile_root>/profile1/installableApps` directory.

- 1. Export the PlantsByWebSphere EAR file.
  - a. In the Enterprise Explorer view on the upper left pane, right-click **PlantsByWebSphere > Export > EAR file**.

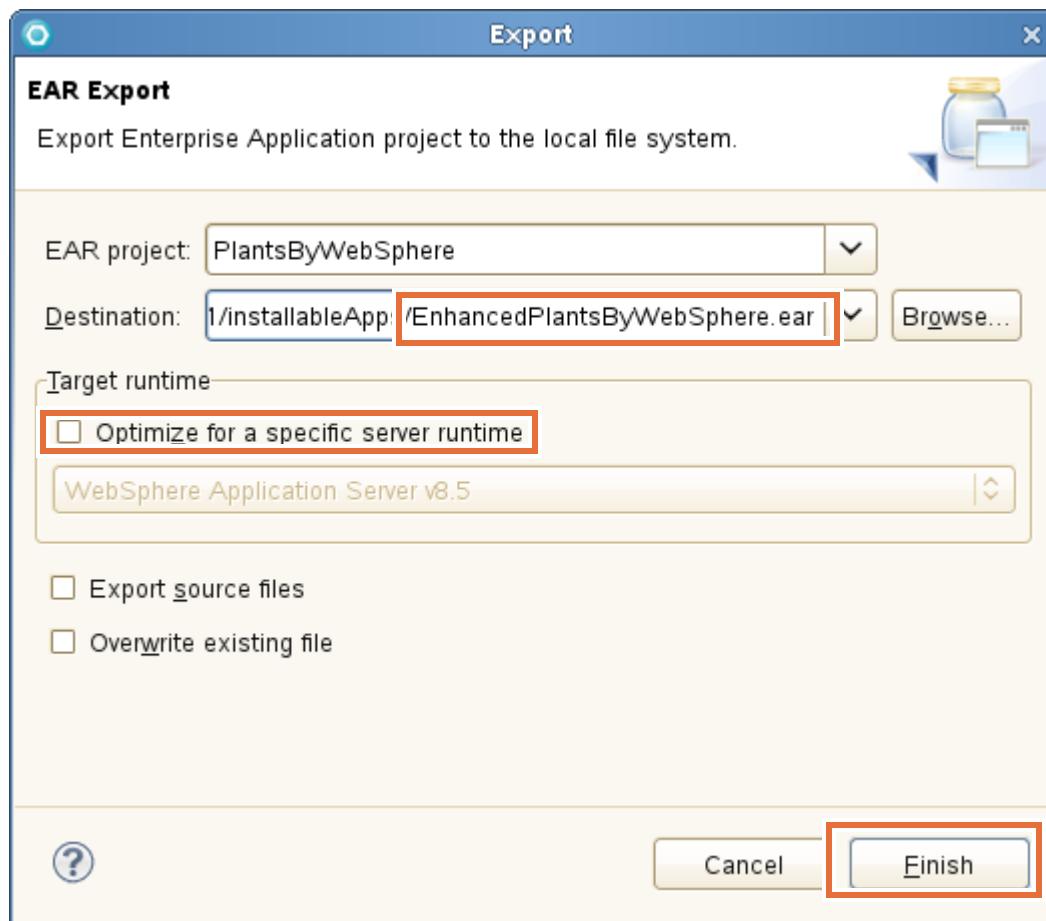


- \_\_ b. In the **Export** dialog, click **Browse** and go to the folder  
`<profile_root>/profile1/installableApps`



- \_\_ c. Click **OK**.

- \_\_ d. In the **Destination** field, change the name of the EAR file to:  
**EnhancedPlantsByWebSphere.ear**  
Clear the check box for **Optimize for a specific server runtime**.



- \_\_ e. Click **Finish** to export the EAR file.



### Information

It is not always obvious when an EAR includes application scoped resources. To make the fact clearer, this exercise includes the word “enhanced” as part of the EAR file name. This convention makes it clear to anyone who might be deploying the EAR in the future that this EAR includes enhancements.

Deploying an enhanced EAR without realizing that application scoped resources are included can cause confusion.

- \_\_ 2. Verify that the EAR file was saved successfully.
- \_\_ a. Use a file system browser to navigate to  
`/opt/IBM/WebSphere/AppServer/profiles/profile1/installableApps`.
  - \_\_ b. Verify that the `EnhancedPlantsByWebSphere.ear` file is present.

- \_\_ 3. Close IBM Assembly and Deploy Tools for WebSphere Administration.
  - \_\_ a. Click **File > Exit**.

**End of exercise**

## Exercise review and wrap-up

In this exercise, IBM Assembly and Deploy Tools for WebSphere Administration was used to assemble the modules for the PlantsByWebSphere application into an enterprise archive. The PlantsByWebSphere application is tested in a later exercise.



# Exercise 6. Installing an application

## What this exercise is about

This exercise covers the tasks that are used to install a WebSphere enterprise application in WebSphere Application Server with the administrative console. The PlantsByWebSphere application, which is packaged as enhanced EAR file, is installed on the server.

This exercise also explores the monitored directory feature, which is also sometimes called the drag-and-drop function. This feature allows a developer or administrator to drag an EAR file into a monitored directory, and WebSphere automatically deploys the application.

## What you should be able to do

At the end of this exercise, you should be able to:

- Use the administrative console to install an application
- Use a web browser to test the application
- Use the drag-and-drop function to deploy an application

## Introduction

In this exercise, you install the PlantsByWebSphere enterprise archive (EAR) file that you assembled with the assembly and deploy tool. The EAR file contains all the application modules, and also contains the definition of other resources that the application requires.

This application is tested by accessing it from a web browser.

Next, you use the drag-and-drop function to install and uninstall the Cache Monitor enterprise archive (EAR) file.

## Requirements

To complete this exercise, WebSphere Application Server must be installed, including a working application server with an administrative console. In addition, you need a web browser and DB2 installed, and the PlantsByWebSphere database must be created and populated.

You also need the `EnhancedPlantsByWebSphere.ear` and `CacheMonitor.ear` files.

# Exercise instructions

## Preface

To do this exercise, you must complete the Installing IBM Installation Manager and Installing WebSphere Application Servers exercises as the exercises set up the environment of the server that is used in this exercise.

### **Section 1: Resetting the WebSphere environment**



#### Note

To reset your WebSphere environment, read **Appendix A** for instructions on how to complete this procedure.

### **Section 2: Start the server and the administrative console**

Use the WebSphere Application Server administrative console to install the PlantsByWebSphere application. Since the administrative console is an application that is running on the server, the server must be running before the administrative console is started.

- \_\_ 1. Start server1 if it is not already running.
  - \_\_ a. The server can be started from the `/opt/IBM/WebSphere/AppServer/bin` directory. When using the `/opt/IBM/WebSphere/AppServer/bin` directory, the profile name must also be specified. In a command-prompt window, navigate to: `/opt/IBM/WebSphere/AppServer/bin`
  - \_\_ b. Enter the following command to start the server:

`./startServer.sh server1 -profileName profile1`

The server is started when a message similar to the following is seen:

`Server server1 open for e-business; process id is 2136`

The process ID is a unique number, which represents the server process that is running on the system.



#### Information

The server can be started without the use of the profile name if you enter the `./startServer.sh server1` command from the `/opt/IBM/WebSphere/AppServer/profiles/profile1/bin` directory.

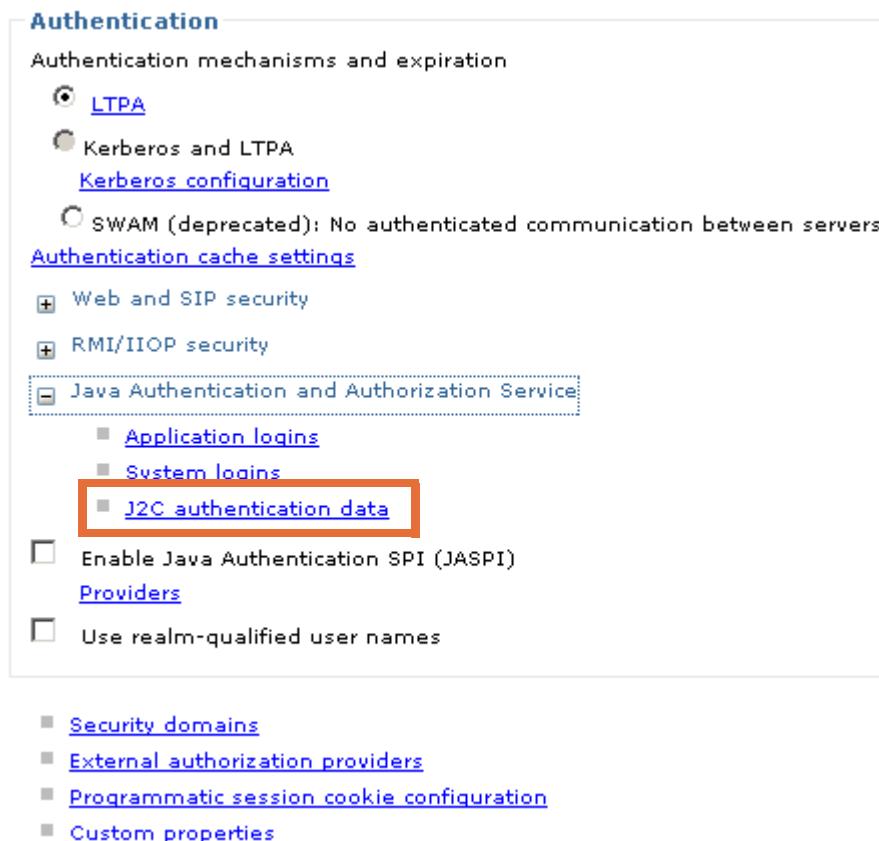
2. Open the administrative console.
- \_\_ a. Open a web browser and enter the following address:  
`http://was85host:9060/ibm/console`  
 If you see a security alert or warning from the web browser, click **OK** or the link to continue to the website.
  - \_\_ b. Enter `wasadmin` for the user ID and `websphere` for the password, and then click **Login**.

### Section 3: Create J2C authentication aliases

Most system resources must be able to authenticate to a registry. Data sources must be able to authenticate to the database server. Here the database is set up to use the local OS user registry.

- \_\_ 1. Create an authentication alias.

  - \_\_ a. From the administrative console, expand **Security** and click **Global security**.
  - \_\_ b. Under **Authentication** on the right, expand **Java Authentication and Authorization Service**.
  - \_\_ c. Click **J2C authentication data**.



The screenshot shows the 'Authentication' configuration page. The left sidebar lists various security mechanisms: LTPA (selected), Kerberos and LTPA, SWAM (disabled), and Java Authentication and Authorization Service (selected). Under J2C authentication data, 'Application logins' and 'System logins' are listed, with 'System logins' expanded to show 'J2C authentication data' (which is highlighted with a red box). Other options like 'Enable Java Authentication SPI (JASPI)' and 'Use realm-qualified user names' are also present.

- \_\_ d. Click **New**.
- \_\_ e. In the **General Properties** area, enter the following values:

**Table 2: J2C details**

| Field       | Value                     |
|-------------|---------------------------|
| Alias       | PlantsApp                 |
| User ID     | db2inst1                  |
| Password    | was1edu                   |
| Description | For PlantsByWebSphere App |

- \_\_ f. Click **OK**.

Notice that the alias was created, but the name is not exactly as you defined. The wizard adds the node name in front of the alias name you entered.

[Global security](#) > JAAS - J2C authentication data

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

**Apply**

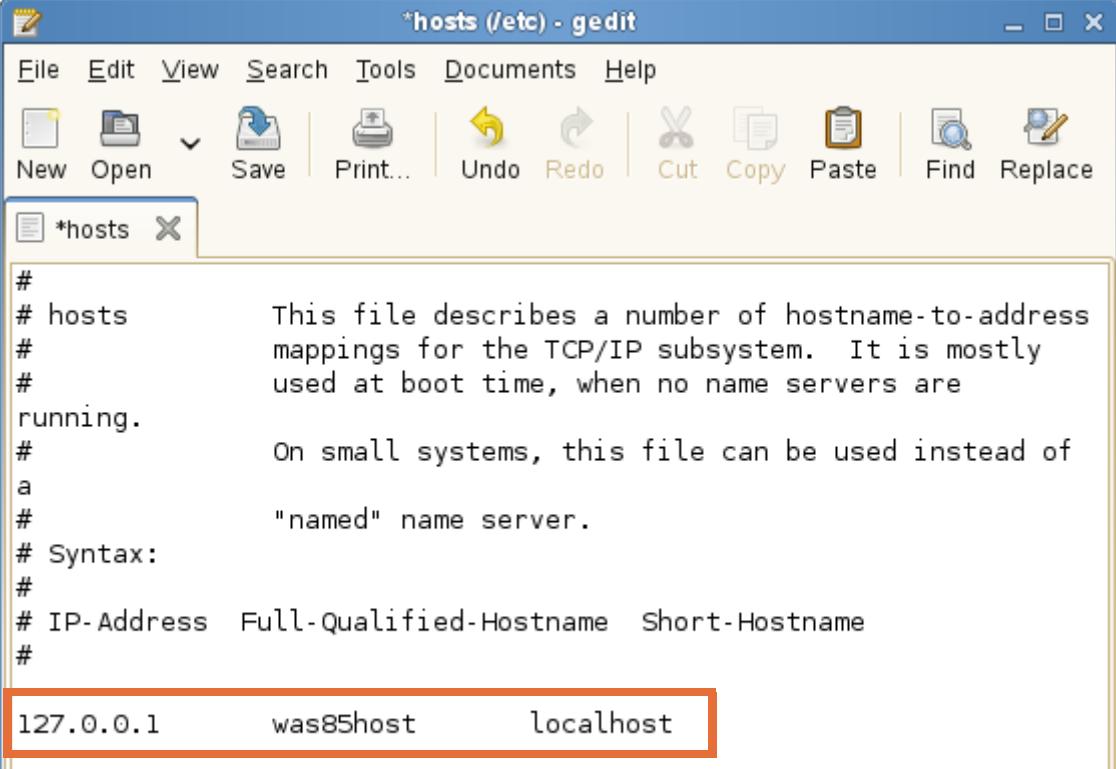
**Preferences**

| Select                   | Alias                     | User ID  | Description               |
|--------------------------|---------------------------|----------|---------------------------|
| <input type="checkbox"/> | was85hostNode01/PlantsApp | db2inst1 | For PlantsByWebSphere App |

Total 1

- \_\_ g. Click the **Save** link to save your changes.
2. Edit the hosts file.
- \_\_ a. Use an editor such as vi to edit the `hosts` file. You can also start the File Browser and navigate to `/etc`. Right-click the `hosts` file and click **Open with "gedit"**.

- \_\_ b. Ensure that the host name was85host is mapped to the localhost IP address.



The screenshot shows the gedit text editor with the file \*hosts (/etc) - gedit open. The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. The toolbar contains icons for New, Open, Save, Print..., Undo, Redo, Cut, Copy, Paste, Find, and Replace. The main window displays the contents of the hosts file:

```

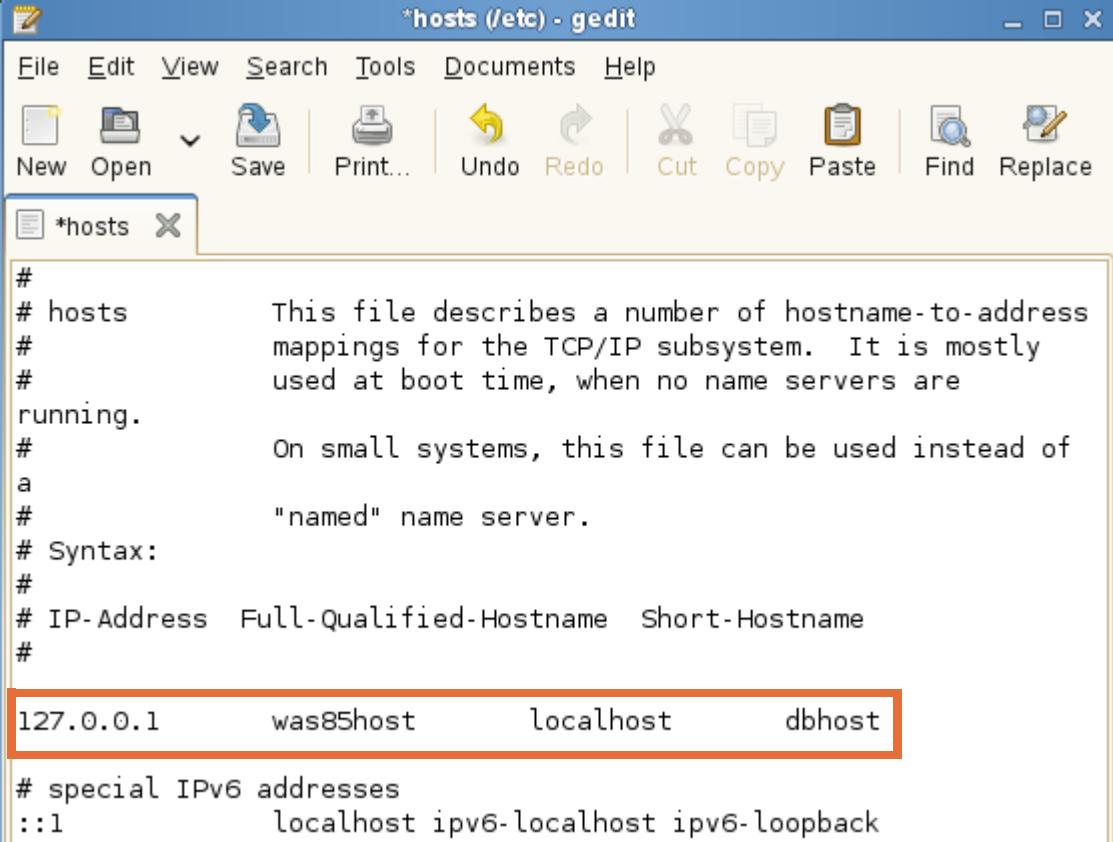
hosts This file describes a number of hostname-to-address
mappings for the TCP/IP subsystem. It is mostly
used at boot time, when no name servers are
running.
On small systems, this file can be used instead of
a
"named" name server.
Syntax:

IP-Address Full-Qualified-Hostname Short-Hostname

127.0.0.1 was85host localhost
```

The line "127.0.0.1 was85host localhost" is highlighted with a red rectangle.

- \_\_\_ c. Edit the hosts file to map dbhost to the IP address where the database server is running. The database is running locally, so the loopback address 127.0.0.1 can be used.



The screenshot shows the gedit text editor with the file \*hosts (/etc) open. The menu bar includes File, Edit, View, Search, Tools, Documents, Help. The toolbar includes New, Open, Save, Print..., Undo, Redo, Cut, Copy, Paste, Find, Replace. The text area contains the hosts file configuration:

```

hosts This file describes a number of hostname-to-address
mappings for the TCP/IP subsystem. It is mostly
used at boot time, when no name servers are
running.
On small systems, this file can be used instead of
a
"named" name server.
Syntax:

IP-Address Full-Qualified-Hostname Short-Hostname

127.0.0.1 was85host localhost dbhost

special IPv6 addresses
::1 localhost ipv6-localhost ipv6-loopback
```



#### Information

In this case, a host name alias of dbhost is being used. This alias allows the database to be moved without changing the configuration. For example, if other users want to test an application on a different host, but with the same database, they define dbhost on their host to point to a different location.

- \_\_\_ d. Save and close the hosts file when completed.

### Section 4: Create a JDBC provider and data sources for the application

If any resources that the application uses are not defined in the EAR file, you must define them. You can use the administrative console to define the resources. In this section, you create the data sources that the PlantsByWebSphere application requires. These data sources define how the application accesses the PLANTS database. You also create the JDBC provider under which the data source exists.



## Information

In general, it is considered an excellent practice to ignore, or remove, application scoped resources from enhanced EAR files when installing applications in a production environment.

- 1. Create the data source.
  - a. From the administrative console, expand **Resources > JDBC > Data sources**.
  - b. Select the **Node=was85hostNode01** scope and click **New**. This step defines the scope at which the data source is visible.

### Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=was85hostNode01 Cell, Node=was85hostNode01

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

| Select  | Name   | JNDI name                        | Scope                | Provider                | Description | Category |
|---------|--------|----------------------------------|----------------------|-------------------------|-------------|----------|
| None    | Plants | jdbc/PlantsByWebSphereDataSource | Node=was85hostNode01 | WebSphere JDBC provider |             |          |
| Total 0 |        |                                  |                      |                         |             |          |

- c. Enter **Plants** for the data source name. This name is just a label and can be anything that you like.
- Enter **jdbc/PlantsByWebSphereDataSource** for the JNDI name. This name is used to look up the data source details. It must be the same name that the

PlantsByWebSphere application uses. Each JNDI name must be unique within the environment.

Create a data source

→ Step 1: Enter basic data source information

Step 2: Select JDBC provider

Step 3: Enter database specific properties for the data source

Step 4: Setup security aliases

Step 5: Summary

**Enter basic data source information**

Set the basic configuration values of a datasource for association with your JDBC provider. A datasource supplies the physical connections between the application server and the database.

Requirement: Use the Datasources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.

Scope  
cells:was85hostNode01 Cell:nodes:was85hostNode01

Data source name  
Plants

JNDI name  
jdbc/PlantsByWebSphereDataSource

**Next** **Cancel**

The screenshot shows the 'Enter basic data source information' step of a 'Create a data source' wizard. The 'Data source name' field is set to 'Plants' and the 'JNDI name' field is set to 'jdbc/PlantsByWebSphereDataSource'. Both of these fields are highlighted with a red rectangular border. The 'Scope' field above them displays the path 'cells:was85hostNode01 Cell:nodes:was85hostNode01'. At the bottom of the screen, there are 'Next' and 'Cancel' buttons.

\_\_ d. Click **Next**.

**Note**

If there are any existing JDBC providers, you see the following screen at Step 2. Select **Create new JDBC provider**, and click **Next**. Otherwise, you go directly to Step 2.1, **Create new JDBC provider**, in the wizard.

**Create a data source**

Step 1: Enter basic data source information

→ **Step 2: Select JDBC provider**

Step 3: Enter database specific properties for the data source

Step 4: Setup security aliases

Step 5: Summary

**Select JDBC provider**

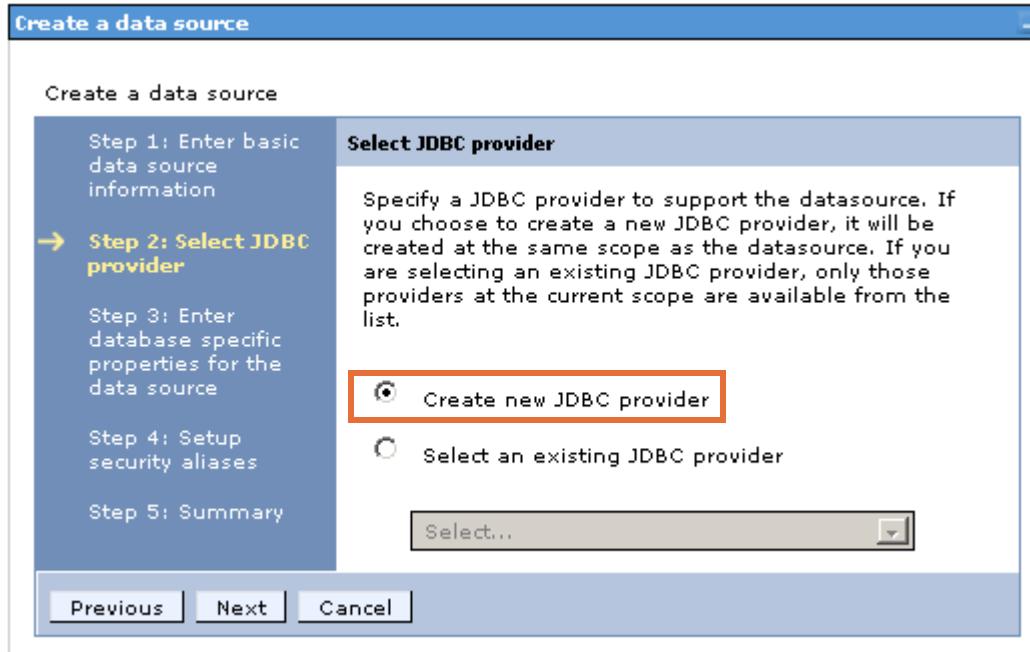
Specify a JDBC provider to support the datasource. If you choose to create a new JDBC provider, it will be created at the same scope as the datasource. If you are selecting an existing JDBC provider, only those providers at the current scope are available from the list.

Create new JDBC provider

Select an existing JDBC provider

Select... ▾

Previous | Next | Cancel



- \_\_ e. Select the values on the table in the **Create new JDBC provider** page. These parameters define the characteristics for the driver that is used to communicate with the database.

Create a data source

**Create new JDBC provider**

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

**Scope**

cells:was85hostNode01 Cell:nodes:was85hostNode01

|                              |                                         |
|------------------------------|-----------------------------------------|
| <b>* Database type</b>       | DB2                                     |
| <b>* Provider type</b>       | DB2 Universal JDBC Driver Provider      |
| <b>* Implementation type</b> | XA data source                          |
| <b>* Name</b>                | DB2 Universal JDBC Driver Provider (XA) |

**Description**

Two-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on the application server for z/OS is not supported for data sources created under this provider.

**Table 3: JDBC details**

| Field               | Value                              |
|---------------------|------------------------------------|
| Database type       | DB2                                |
| Provider type       | DB2 Universal JDBC Driver Provider |
| Implementation type | XA data source                     |

- \_\_ f. Keep all remaining defaults and click **Next**.
- \_\_ g. On the next page, you define where in the file system the JDBC provider finds the JDBC drivers. In this case, the location is defined as the following directory:

/opt/ibm/db2/V9.7/java

Enter this value in both directory location fields.

**Enter database class path information**

Set the class path for the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays a default list of jars and allows you to set the environment variables that define the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).

Entries are separated by using the ENTER key and must not contain path separator characters (such as ';' or ':'). If a value is specified for you, you may click Next to accept the value.

Class path:

```
 ${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar
 ${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar
 ${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar
```

Directory location for "db2jcc.jar, db2jcc\_license\_cisuz.jar" which is saved as WebSphere variable

Native library path

Directory location which is saved as WebSphere variable



### Information

The wizard creates the paths that point to the JDBC drivers and puts values into the WebSphere environment variables named: DB2UNIVERSAL\_JDBC\_DRIVER\_PATH and DB2UNIVERSAL\_JDBC\_DRIVER\_NATIVEPATH

If these variables are set before this step, their values prefill the entry fields. Anything that is entered here overwrites the environment variables.

- \_\_ h. Click **Next**.
- \_\_ i. Now you are back in the data source definition part of the wizard and ready to define the database properties. Enter the following parameters on the page:

**Table 4: Data source details**

| Field         | Value  |
|---------------|--------|
| Driver type   | 4      |
| Database name | PLANTS |
| Server name   | dbhost |

**Table 4: Data source details**

| Field                                                  | Value                   |
|--------------------------------------------------------|-------------------------|
| Port number                                            | 50001                   |
| Use this data source in container managed persistence. | This option is checked. |

- \_\_ j. Click **Next**.
- \_\_ k. On the next page, select **was85hostNode01/PlantsApp** for the **Component-managed authentication alias**, and click **Next**.

Create a data source

Step 1: Enter basic data source information

Step 2: Select JDBC provider

Step 2.1: Create new JDBC provider

Step 2.2: Enter database class path information

Step 3: Enter database specific properties for the data source

→ Step 4: Setup security aliases

Step 5: Summary

**Setup security aliases**

Select the authentication values for this resource.

Authentication alias for XA recovery  
↓ (none)

Component-managed authentication alias  
↓ was85hostNode01/PlantsApp (highlighted with a red box)

Mapping-configuration alias  
↓ (none)

Container-managed authentication alias  
↓ (none)

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost.

[Global J2C authentication alias](#)  
[Security domains](#)

Previous Next Cancel



### Information

By specifying the PlantsApp authentication alias, when the system accesses the database, the data source uses the user name and password that were previously specified.

Separating the user name and password out from the data source definition is important. This indirection allows for the password to be changed in a single place instead of on every resource that might use the same authentication.

The **Authentication alias for XA recovery** option is used to specify the authentication alias that you must use during XA recovery processing.

The **Component-managed authentication alias** involves creating a mapping from an alias name to the user name and password. This alias name is then specified administratively on the connection factory or data source. As the alias can be resolved in the application server only, the alias restricts authenticated access to applications that are running in the application server.

The **Container-managed authentication alias** works in much the same way as the component-managed alias, but the connection factory or data source must be looked up by using a resource-reference that specifies a resource-auth of container. As a consequence, for an application to retrieve the authenticated resource, the administrator must explicitly bind the resource-reference to the resource on deployment of the application.

- \_\_\_ I. On the next page, verify all the values that are entered. Click **Finish** to create the data source and JDBC provider.
- \_\_\_ m. **Save** your changes.
- \_\_\_ 2. Test the data source connections.
  - \_\_\_ a. On the Data Sources page, select the check box for **Plants**, and click **Test connection**.

| Select                              | Name                   | JNDI name                        | Scope                | Provider                                | Description                     | Category |
|-------------------------------------|------------------------|----------------------------------|----------------------|-----------------------------------------|---------------------------------|----------|
| <input checked="" type="checkbox"/> | <a href="#">Plants</a> | jdbc/PlantsByWebSphereDataSource | Node=was85hostNode01 | DB2 Universal JDBC Driver Provider (XA) | DB2 Universal Driver Datasource |          |

Total 1

- \_\_\_ b. Make sure that the connection was successful. Look for the “successful” messages at the top of the work area.

Messages

The test connection operation for data source Plants on server server1 at node was85hostNode01 was successful.

**Information**

This test verifies that the application server is able to connect to the data source that was defined. A success means that the application server is able to connect to host dbhost on port 50001. It can also connect to the database PLANTS with the user name and password that are supplied in the J2C authentication alias. If any of those pieces are incorrectly defined, the test fails.

## **Section 5: Install the PlantsByWebSphere enterprise application**

**Information**

The enterprise archive file (EAR) is enhanced. The enhancement is that this EAR file contains the definition of “how to access the database.” This information is important and useful for test purposes. This EAR enhancement breaks down the “separation of concerns” barrier that is implicit within the Java EE roles. The administration role defines the resources that are required for the application. In the steps that are listed here, you ignore the enhanced part of the EAR file, and define resources from the administrative console.

- 1. Install the EnhancedPlantsByWebSphere.ear file.
  - a. From the administrative console, expand **Applications** and click **New Application**.
  - b. Click **New Enterprise Application**.

**New Application**

**New Application**

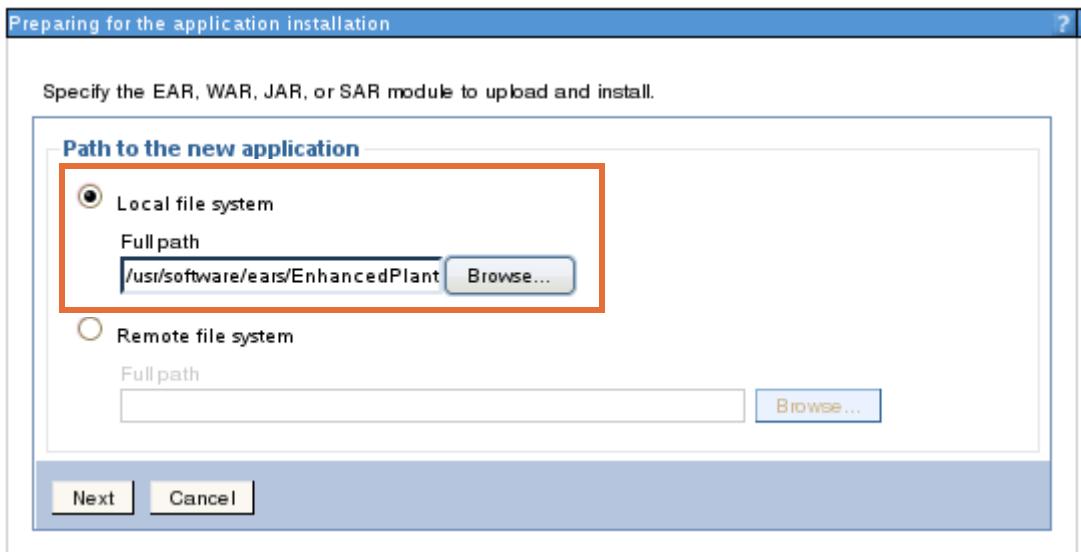
This page provides links to create new applications of different types.

**Install a New Application**

- [New Enterprise Application](#)
- [New Business Level Application](#)
- [New Asset](#)

- c. Select **Local file system** and click **Browse**.

- \_\_\_ d. Navigate to /usr/software/ears and select the **EnhancedPlantsByWebSphere.ear** file. Click **Open**. Review the local file system information and click **Next**.

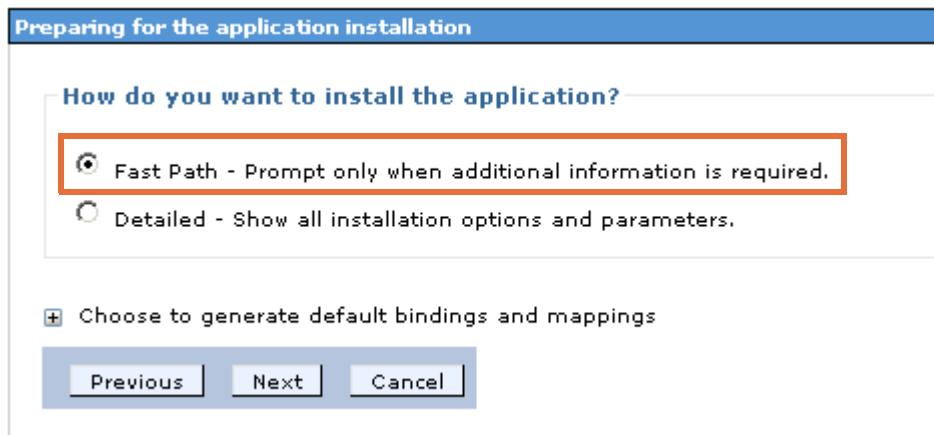


### Information

The EAR file that you are installing can be on either the client computer or the server computer. The client computer runs the browser, and the server computer is the computer to which the client is connected. If you specify an EAR file on the client computer, select **Local file system**. Then, the administrative console uploads the EAR file to the computer on which the console is running and proceeds with application installation.

If you are using a browser on a remote location, select **Remote file system** and browse through the file system where the application server is running.

- \_\_ e. Select **Fast Path - Prompt only when additional information is required**, and click **Next**.



### Information

The **Fast Path** method limits the number of options that are shown, which simplifies the installation process. The **Detailed** method shows all the installation options, including the options with default values assigned.

- \_\_\_ f. On the next page, you can select any additional installation options. Clear the box for **Process embedded configuration**. Make sure that this option is not selected.

**Install New Application**

Specify options for installing enterprise applications and modules.

**Step 1: Select installation options**

- [Step 2 Map modules to servers](#)
- Step 3 Metadata for modules**
- [Step 4 Summary](#)

**Select installation options**

Specify the various options that are available for your application.

Precompile JavaServer Pages files

Directory to install application  
[ ]

Distribute application

Use Binary Configuration

Deploy enterprise beans

Application name  
PlantsByWebSphere

Create MBeans for resources

Override class reloading settings for Web and EJB modules

Reload interval in seconds  
[ ]

Deploy Web services

Validate Input off/warn/fail  
warn [ ]

Process embedded configuration

**File Permission**

|                                                   |     |
|---------------------------------------------------|-----|
| Allow all files to be read but not written to     | [ ] |
| Allow executables to execute                      | [ ] |
| Allow HTML and image files to be read by everyone | [ ] |

\*,\*,dll=755#,\*,so=755#,\*,a=755#,\*,\*,sl=755



### Information

If an EAR file is enhanced, the **Process embedded configuration** check box is selected by default. To ignore the application-scoped resources, the **Process embedded configuration** option must not be selected.

If this enhanced EAR file is installed with the **Process embedded configuration** checked, then various properties are set at the application scope level. Caution must be used when working with application scoped resources because they are not as clearly visible as resources set at higher level scopes.

- Application scoped resources are tied to a specific application. Enhanced EAR files include application resources.

- Settings that are made at the application scope level take precedence over the same settings that are set at a higher level scope, such as the cell or node levels.
- Application scoped resources are not available from scope selection menus.

It is problematic if an administrator is trying to troubleshoot a problem and is not aware that an application is enhanced. A setting at the application scope can cause a problem with the application. The administrator might review all the settings at the various scopes and never look at the application scope settings.

- g. Click **Next**.
- h. Click the **Step 4: Summary** link.
- i. Review the options and click **Finish** to complete the installation.
- j. Look for the message that the application installed successfully.

ADMA5005I: The application PlantsByWebSphere is configured in the WebSphere Application Server repository.

ADMA5113I: Activation plan created successfully.

ADMA5011I: The cleanup of the temp directory for application PlantsByWebSphere is complete.

ADMA5013I: Application PlantsByWebSphere installed successfully.

Application PlantsByWebSphere installed successfully.

To start the application, first save changes to the master configuration.

Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

- k. **Save** the changes.
- 2. Start the PlantsByWebSphere application.
- a. Expand **Applications > Application Types > WebSphere enterprise applications**.
- b. Verify that the **PlantsByWebSphere** application is listed.

- \_\_\_ c. Select the check box next to **PlantsByWebSphere** and click **Start**.

#### Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

##### Preferences

| Start                                       | Stop                               | Install            | Uninstall | Update | Rollout Update | Remove File | Export |
|---------------------------------------------|------------------------------------|--------------------|-----------|--------|----------------|-------------|--------|
|                                             |                                    |                    |           |        |                |             |        |
| Select                                      | Name                               | Application Status |           |        |                |             |        |
| You can administer the following resources: |                                    |                    |           |        |                |             |        |
| <input type="checkbox"/>                    | <a href="#">DefaultApplication</a> |                    |           |        |                |             |        |
| <input checked="" type="checkbox"/>         | <a href="#">PlantsByWebSphere</a>  |                    |           |        |                |             |        |
| <input type="checkbox"/>                    | <a href="#">itvApp</a>             |                    |           |        |                |             |        |
| <input type="checkbox"/>                    | <a href="#">query</a>              |                    |           |        |                |             |        |
| Total 4                                     |                                    |                    |           |        |                |             |        |

- \_\_\_ d. Wait for the application to start successfully.



#### Note

If the application is not running, look at the `SystemOut.log` file in the `/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1` directory for more information.

- \_\_\_ 3. Log out of the administrative console and minimize the web browser.

## Section 6: Test the enterprise application

Test the application by accessing it with the WebSphere Application Server HTTP transport.

- \_\_ 1. Open the PlantsByWebSphere application and log on.
- \_\_ a. Open a new web browser, and access PlantsByWebSphere by entering the following address:

`http://was85host:9080/PlantsByWebSphere`

### PLANTS BY WEBSPHERE

| Tips                                                                                                                                                                                                                 | Specials                                                                                                                                                                                                                                           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Preserve extra grass seed by keeping it dry. Tape boxes and bags closed, or seal them into plastic bags. Be sure to remove extra air from the bags. Store all seed in a cool, dry area such as a garage or basement. | <br>Bonsai Tree<br>\$30.00 each <br>Red Delicious Straw<br>\$3.50 (50 seed |



[Flowers](#) : [Fruits & Vegetables](#) : [Trees](#) : [Accessories](#) : [Home](#) : [Shopping Cart](#) : [Help](#)

- \_\_ b. Click **LOGIN** on the upper right menu bar. The PlantsByWebSphere login page is shown.

- \_\_\_ c. Click the link that is titled, “register for your own account here.”

## PLANTS BY WEBSPHERE



### Login or Register

If you are a returning customer and previously set up an account, please enter your e-mail address and password below.

E-mail address:

Password:

If you are a New customer you can [register for your own account here.](#)

- \_\_ d. Create your own user information and click **Register**.

## Registration

Enter the information below to set up your account. This information will not be shared without your permission. With your permission we will only share your name and email address with our trusted business partners.

Required fields are denoted with a red asterisk (\*).

### Login Information

|                   |                                              |
|-------------------|----------------------------------------------|
| E-mail address *  | <input type="text" value="user@plants.com"/> |
| Password *        | <input type="password" value="*****"/>       |
| Verify Password * | <input type="password" value="*****"/>       |

### Contact Information

|                   |                                           |
|-------------------|-------------------------------------------|
| First Name *      | <input type="text" value="Ima"/>          |
| Last Name *       | <input type="text" value="Gardner"/>      |
| Address Line 1 *  | <input type="text" value="123 Main St."/> |
| Address Line 2    | <input type="text"/>                      |
| City *            | <input type="text" value="Gotham"/>       |
| State *           | <input type="text" value="PA"/>           |
| ZIP Code *        | <input type="text" value="15222"/>        |
| Phone (daytime) * | <input type="text" value="412-555-1234"/> |

- \_\_\_ e. Click **Help** on the top or bottom menu bar. The PlantsByWebSphere help page is seen.

**PLANTS BY WEBSPHERE**

Flowers    Fruits & Vegetables    Trees    Accessories

Home >

## Help

Plants By WebSphere provides limited help support. See the sample docs d for documentation on the design, building, and installation of the sample. Debug mode has been tied to the JSF project stage declaration. Debug mes be displayed when the web app's javax.faces.PROJECT\_STAGE context para to either Development or UnitTest. A value of SystemTest or Production wil debug output. The current state of debugging is indicated in the check box

**Debug messages enabled**

If the database becomes corrupted for some reason, the button below can l delete all data currently in the database and populate it with a fresh set of this does not work, stop the server and repeat the prerequisite steps found docs directory to unzip the Derby database.

[Reset database](#)

[View Server Info](#)

[Admin Home](#)

**Powered by**  
**IBM WebSphere®**  
e-business software ►

Flowers : Fruits & Vegetables : Trees : Accessories : Home ::

- \_\_\_ f. Click the link **View Server Info**. This page provides information about the environment of the PlantsByWebSphere application. This information is useful when running an application in a clustered mode.

## PLANTS BY WEBSPHERE

HOME : ADM

### Runtime server information

| Cell                | Node            | Process | Session Data |
|---------------------|-----------------|---------|--------------|
| was85hostNode01Cell | was85hostNode01 | server1 | null         |

Session Data  Update Refresh

Show cookies

Tue Aug 14 14:49:30 EDT 2012



Home : Admin Home : Help



### Note

Be sure that you understand the information that is shown on the **Server Info** page and that you understand how to update the session data. This page is useful when testing multiple servers in a clustered configuration.

The **Server Info** page is not part of the PlantsByWebSphere application that comes with the WebSphere Application Server product. The **Server Info** page is created specifically for use in this course material.

- \_\_\_ g. Enter Testing into the Session Data field and click **Update**.

The columns for **Session Data** and **Session Created** are updated.

**PLANTS BY WEBSPHERE**

HOME : ADM

| <b>Runtime server information</b> |                 |                |                     |                        |
|-----------------------------------|-----------------|----------------|---------------------|------------------------|
| <b>Cell</b>                       | <b>Node</b>     | <b>Process</b> | <b>Session Data</b> | <b>Session Created</b> |
| was85hostNode01Cell               | was85hostNode01 | server1        | Testing             | Tue Aug 14             |

Session Data    Testing    Update    Refresh

Show cookies

Tue Aug 14 14:50:47 EDT 2012

Powered by  
 Home : Admin Home : Help

Clicking **Refresh** (or refreshing the page with the browser) reloads the page, and the date and time information in the lower left is updated. These steps demonstrate that the page is reloaded, but the session data remains the same.

- \_\_\_ h. Click **Home** and explore other parts of the PlantsByWebSphere application. For example, click **Flowers** and explore the flowers that are listed.



### Information

If the PlantsByWebSphere application works correctly, you are able to retrieve the flower inventory. This success means that the PlantsByWebSphere EAR was installed correctly. The browser was able to communicate with the right port (9080) on the application server. The application server was able to find and use the right data source and JDBC driver to communicate with the DB2 database. The application server was able to communicate with DB2 on the correct host (dbhost), with the correct user name and password as defined in the J2C authentication alias (PlantsApp).

If any of these items are configured incorrectly, the application does not work.

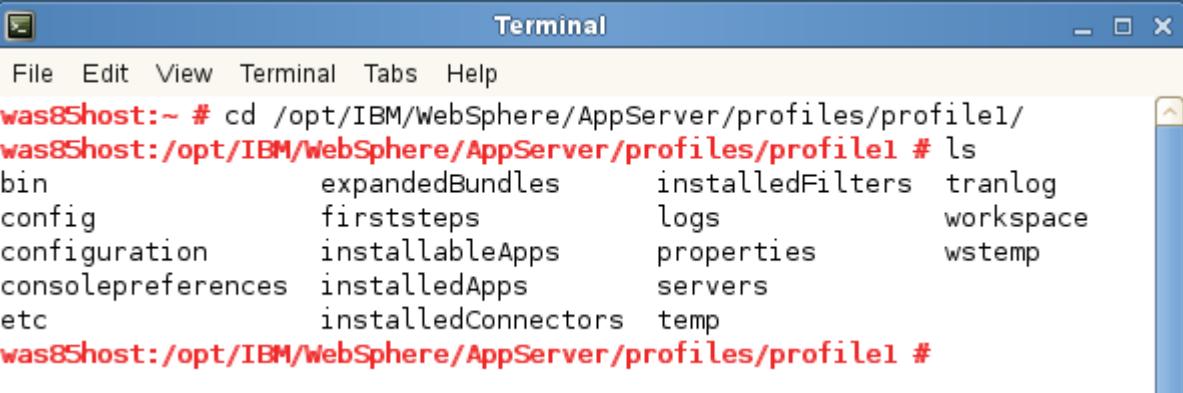
- \_\_\_ 2. It is not necessary to log out of the application, but do close the web browser window.

## Section 7: Use a monitored directory to deploy an enterprise application

In this section, the new monitored directory feature is used to deploy an EAR file. This feature allows the deployment of an application by dragging, or copying, an EAR file into a monitored directory. The application is automatically installed and started.

The monitored directory feature is not enabled by default. The first step is to use the administrative console to enable the feature. This step creates the directory structure that the monitored directory feature uses.

- \_\_\_ 1. Verify that the monitored directory does not yet exist.
  - \_\_\_ a. Open a terminal window.
  - \_\_\_ b. Navigate to the `/opt/IBM/WebSphere/AppServer/profiles/profile1` directory and use the following command to get a directory listing:  
`ls`



```
File Edit View Terminal Tabs Help
was85host:~ # cd /opt/IBM/WebSphere/AppServer/profiles/profile1
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1 # ls
bin expandedBundles installedFilters tranlog
config firststeps logs workspace
configuration installableApps properties wstemp
consolepreferences installedApps servers
etc installedConnectors temp
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1 #
```

- \_\_\_ c. Notice that there is no directory named `monitoredDeployableApps`.
- \_\_\_ 2. Use the administrative console to enable the monitored directory.
  - \_\_\_ a. Maximize the administrative console browser window.
  - \_\_\_ b. Log in using `wasadmin` as the user name and `websphere` as the password.
  - \_\_\_ c. Click **Applications > Global deployment settings**.

- \_\_\_ d. Select the box for **Monitor directory to automatically deploy applications** and accept the defaults for the other fields.

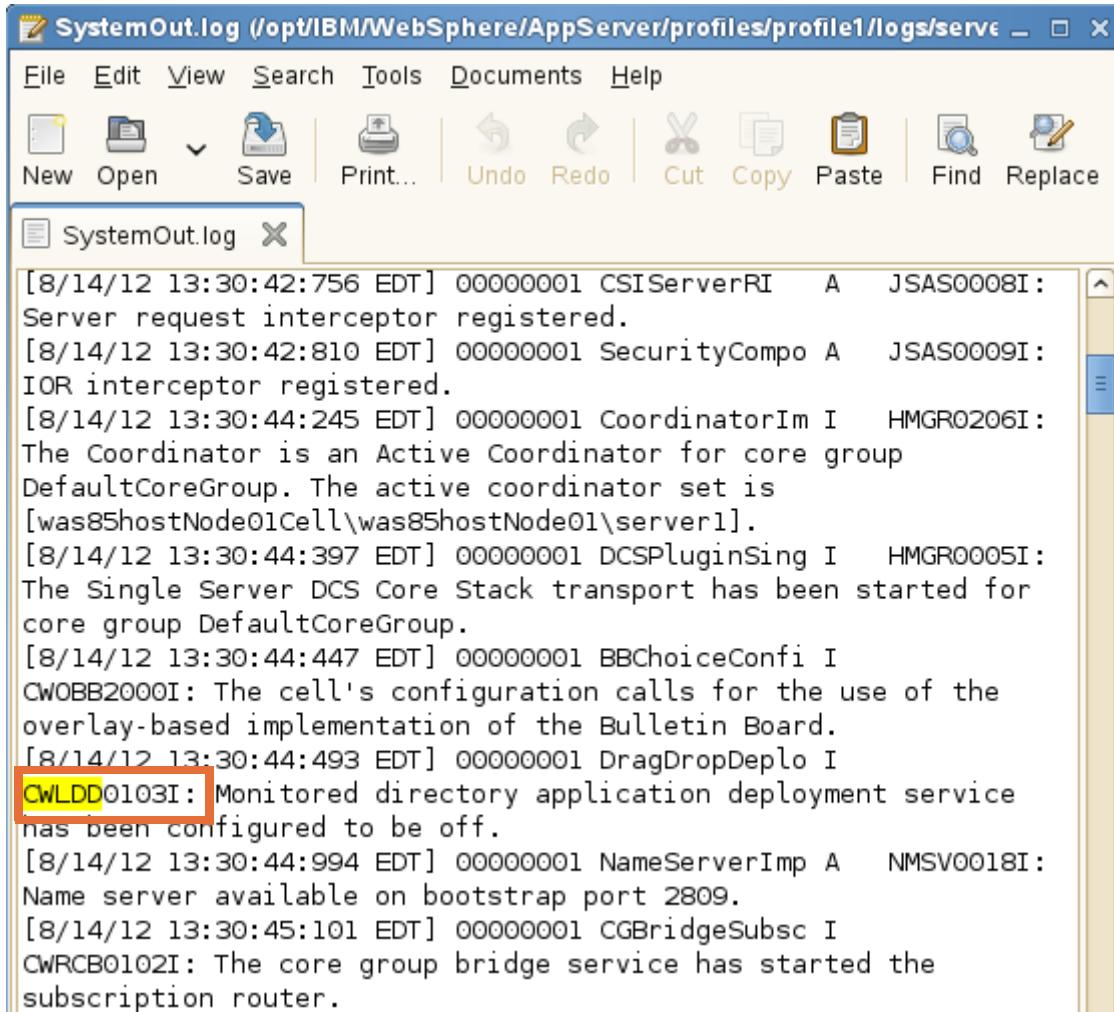
The screenshot shows the 'Global deployment settings' page. At the top, there is a note: 'Use this page to manage settings that apply to all applications. NOTE: some settings may vary by application type.' Below this, there are two tabs: 'Configuration' (selected) and 'General Properties'. Under 'General Properties', there is a section titled 'Monitored Directory Deployment'. It contains the following fields:

- A checked checkbox labeled 'Monitor directory to automatically deploy applications'.
- A field labeled 'Monitored directory' containing the value '\${USER\_INSTALL\_ROOT}/monitoredDeployableApps'.
- A field labeled 'Polling interval' containing the value '5' followed by 'seconds'.

At the bottom of this section are 'Apply' and 'Reset' buttons.

- \_\_\_ e. Click **Apply**.
- \_\_\_ f. **Save** the changes.
- \_\_\_ 3. Examine the log file.
- \_\_\_ a. Use the terminal window and navigate to the log directory for profile1, which is:  
/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1

- \_\_ b. Use a text editor (such as gedit) to open the SystemOut.log file and search for the string: cwldd



```
[8/14/12 13:30:42:756 EDT] 00000001 CSIServerRI A JSAS0008I:
Server request interceptor registered.
[8/14/12 13:30:42:810 EDT] 00000001 SecurityCompo A JSAS0009I:
IOR interceptor registered.
[8/14/12 13:30:44:245 EDT] 00000001 CoordinatorIm I HMGR0206I:
The Coordinator is an Active Coordinator for core group
DefaultCoreGroup. The active coordinator set is
[was85hostNode01Cell\was85hostNode01\server1].
[8/14/12 13:30:44:397 EDT] 00000001 DCSPluginSing I HMGR0005I:
The Single Server DCS Core Stack transport has been started for
core group DefaultCoreGroup.
[8/14/12 13:30:44:447 EDT] 00000001 BBChoiceConfi I
CW0BB2000I: The cell's configuration calls for the use of the
overlay-based implementation of the Bulletin Board.
[8/14/12 13:30:44:493 EDT] 00000001 DragDropDeplo I
CWLDD0103I: Monitored directory application deployment service
has been configured to be off.
[8/14/12 13:30:44:994 EDT] 00000001 NameServerImp A NMSV0018I:
Name server available on bootstrap port 2809.
[8/14/12 13:30:45:101 EDT] 00000001 CGBridgeSubsc I
CWRCB0102I: The core group bridge service has started the
subscription router.
```



### Note

The monitored directory works not only with stand-alone application servers, but also in a network deployment environment. Some of the log file screen captures actually show the messages from a network deployment environment. The messages in your stand-alone environment provide the same information.

- \_\_ 4. Restart server1.

- \_\_ a. Using a terminal window, navigate to the /opt/IBM/WebSphere/AppServer/profiles/profile1/bin directory and enter the following commands:

```
./stopServer.sh server1 -username wasadmin -password web1sphere
./startServer.sh server1
```

- \_\_\_ b. Again, examine the server1 SystemOut.log file. Search for the string: cwldd  
 Notice that entries indicate that the service is started. Also, notice that the description string immediately preceding the CWLDD is DragDropDeplo.

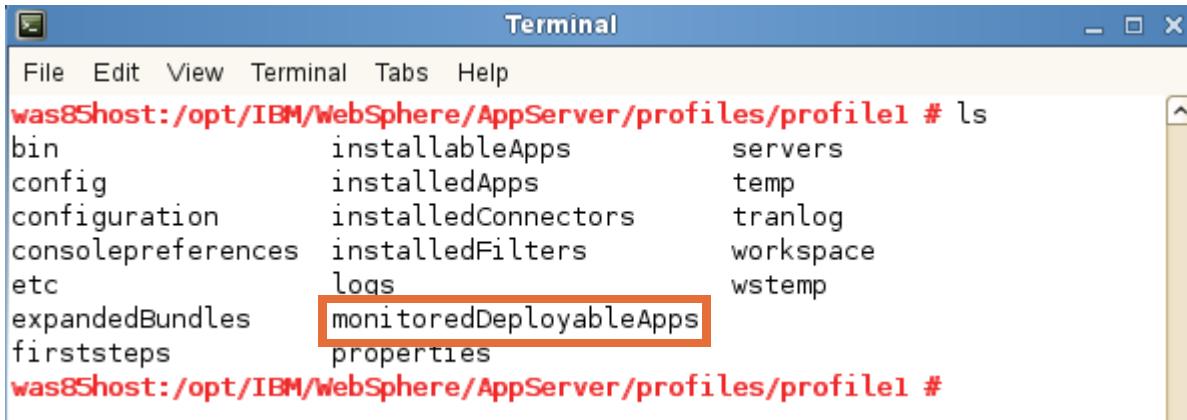
```

The Coordinator is an Active Coordinator for core group
DefaultCoreGroup. The active coordinator set is
[was85hostNode01Cell\was85hostNode01\server1].
[8/15/12 11:48:49:989 EDT] 00000001 DCSPluginSing I HMGR0005I:
The Single Server DCS Core Stack transport has been started for
core group DefaultCoreGroup.
[8/15/12 11:48:50:031 EDT] 00000001 BBChoiceConfig I
CW0BB2000I: The cell's configuration calls for the use of the
overlay-based implementation of the Bulletin Board.
[8/15/12 11:48:50:232 EDT] 00000001 DragDropDeplo I
CWLDD0001I: Starting monitored directory application deployment
service....
[8/15/12 11:48:50:503 EDT] 00000001 DragDropDeplo I
CWLDD0002I: Monitored directory application deployment service
is started and monitoring file changes in directory: /opt/IBM/
WebSphere/AppServer/profiles/profile1/monitoredDeployableApps.
[8/15/12 11:48:50:834 EDT] 00000001 NameServerImp A NMSV0018I:
Name server available on bootstrap port 2809.
[8/15/12 11:48:50:868 EDT] 00000001 CGBridgeSubscr I
CWRCB0102I: The core group bridge service has started the
subscription router.
[8/15/12 11:48:54:299 EDT] 00000001 UserRegistryI A SECJ0136I:

```

- \_\_\_ c. **Close** the SystemOut.log file when completed.

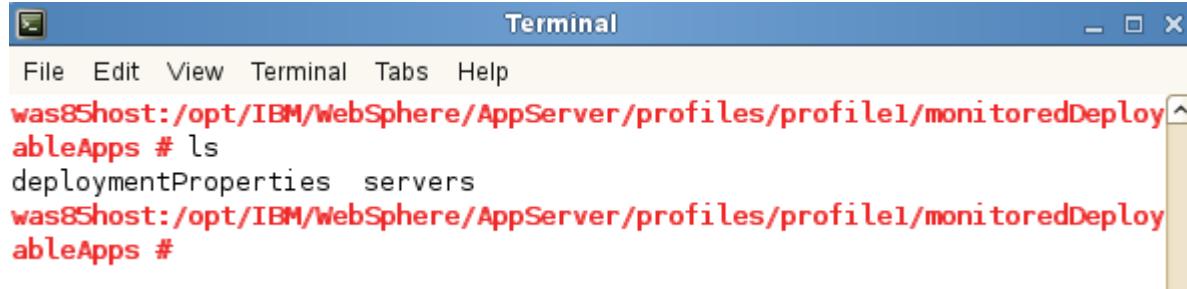
- \_\_\_ 5. Verify that the `monitoredDeployableApps` directory is created.
- \_\_\_ a. Using the terminal window, navigate to the `/opt/IBM/WebSphere/AppServer/profiles/profile1` directory and use the following command to get another directory listing:
- ```
ls
```



The screenshot shows a terminal window titled "Terminal". The command `ls` is run, displaying the contents of the `profile1` directory. The `monitoredDeployableApps` directory is highlighted with a red box.

```
File Edit View Terminal Tabs Help  
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1 # ls  
bin installableApps servers  
config installedApps temp  
configuration installedConnectors tranlog  
consolepreferences installedFilters workspace  
etc logs wstemp  
expandedBundles monitoredDeployableApps  
firststeps properties  
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1 #
```

- ___ b. Notice that the `monitoredDeployableApps` directory is now there.
- ___ c. Navigate to the `monitoredDeployableApps` directory and use the following command to get a directory listing:
- ```
ls
```



The screenshot shows a terminal window titled "Terminal". The command `ls` is run within the `monitoredDeployableApps` directory. It shows two sub-directories: `deploymentProperties` and `servers`.

```
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/monitoredDeployableApps # ls
deploymentProperties servers
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/monitoredDeployableApps #
```

- \_\_\_ d. Notice that by default, there are directories for `servers` and `deploymentProperties`. For this section, only the `servers` directory is used.

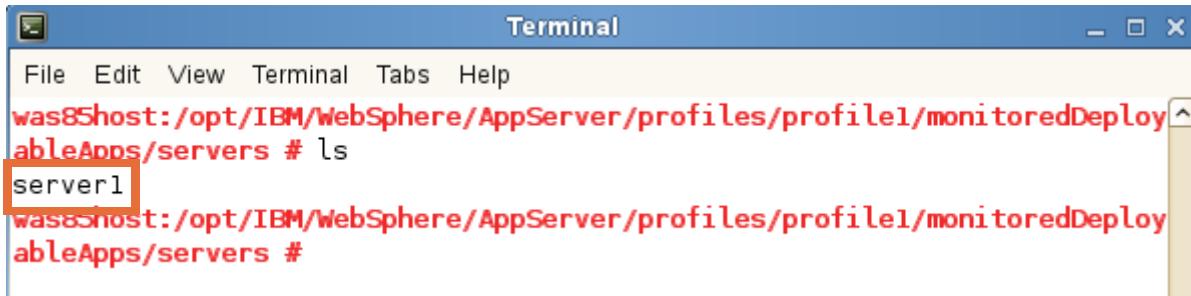


### Information

In a federated environment, the directory contents are slightly different since clusters are also supported. You also manually create each server directory by using the exact name of the server.

- \_\_\_ e. Change to the servers directory and use the following command to get a directory listing:  
`ls`

Notice that a subdirectory exists and is called server1.



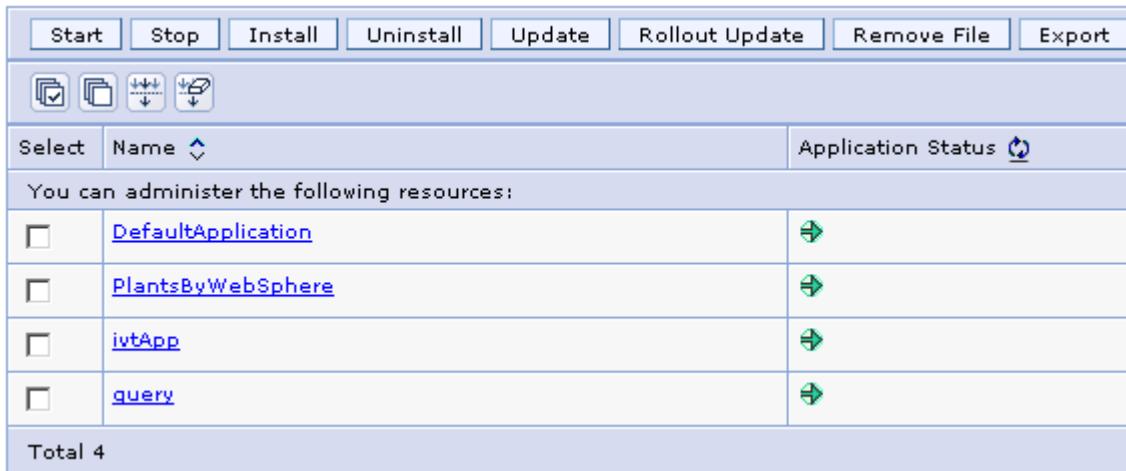
```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/monitoredDeployableApps/servers # ls
server1
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/monitoredDeployableApps/servers #
```



### Information

The server1 directory is used to deploy applications to server1. If there were multiple server1 instances within the cell, it would also be possible to create a directory structure of the format nodes/<node-name>/server1. This directory allows you to specify to which server1 an application is deployed.

- \_\_\_ 6. Use the drag-and-drop feature to install the Cache Monitor application.
- \_\_\_ a. Using the administrative console, click **Applications > Application Types > WebSphere enterprise applications**. Notice which applications are currently deployed.

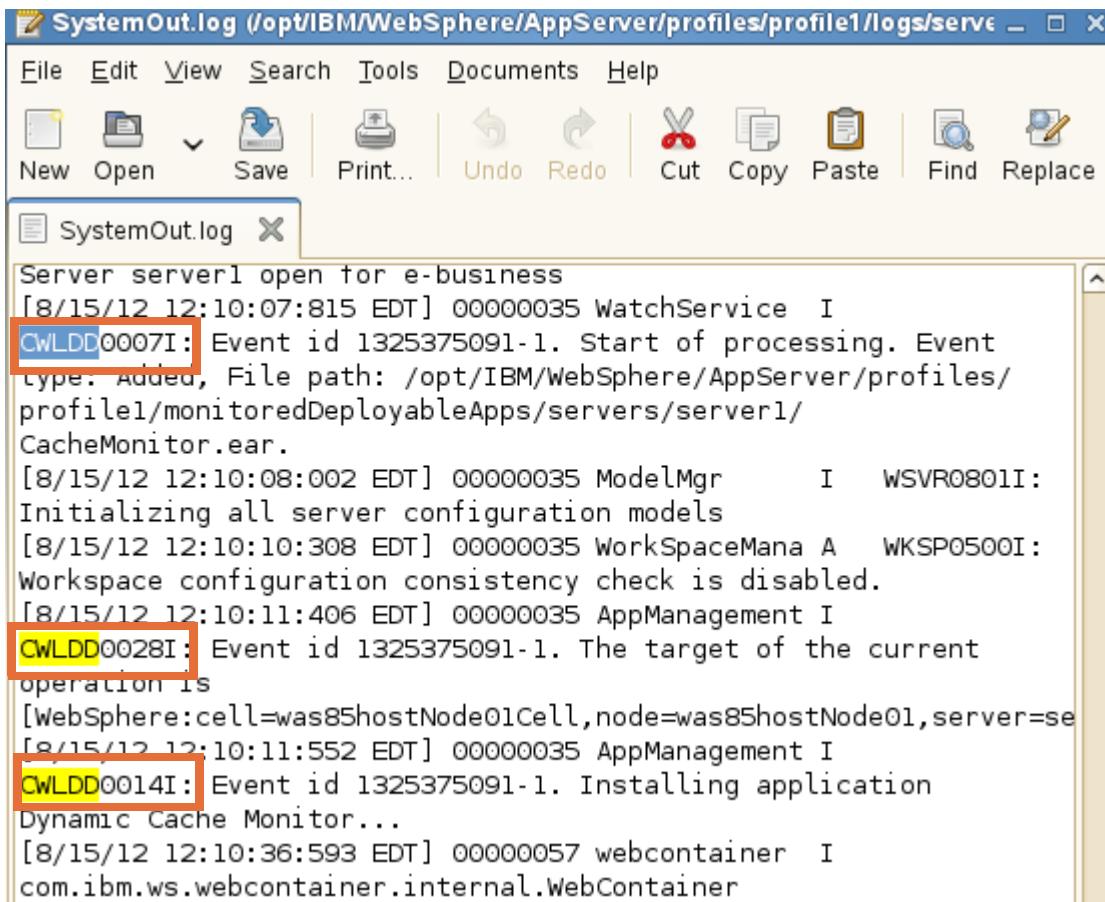


| You can administer the following resources: |                                    |                    |
|---------------------------------------------|------------------------------------|--------------------|
| <input type="checkbox"/>                    | Name                               | Application Status |
| <input type="checkbox"/>                    | <a href="#">DefaultApplication</a> |                    |
| <input type="checkbox"/>                    | <a href="#">PlantsByWebSphere</a>  |                    |
| <input type="checkbox"/>                    | <a href="#">ivtApp</a>             |                    |
| <input type="checkbox"/>                    | <a href="#">query</a>              |                    |

Total 4

- \_\_\_ b. Open two File Browser windows and copy the CacheMonitor.ear file into the monitoredDeployableApps directory. The details on where to copy from and to are as follows:
- From:  
`/opt/IBM/WebSphere/AppServer/installableApps/CacheMonitor.ear`

- To: /opt/IBM/WebSphere/AppServer/profiles/profile1/monitoredDeployableApps/servers/server1/
- \_\_\_ c. Open the SystemOut.log file for server1. Search for the string: cwldd  
Notice that there are new entries in the log file.



The screenshot shows a Windows Notepad window titled "SystemOut.log /opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1". The window contains several lines of log output. Four specific lines are highlighted with colored boxes: a red box surrounds the first line, a yellow box surrounds the second line, another red box surrounds the fourth line, and another yellow box surrounds the fifth line. The log entries are:

```
Server server1 open for e-business
[8/15/12 12:10:07:815 EDT] 00000035 WatchService I
CWLDD007I: Event id 1325375091-1. Start of processing. Event
type: Added, File path: /opt/IBM/WebSphere/AppServer/profiles/
profile1/monitoredDeployableApps/servers/server1/
CacheMonitor.ear.
[8/15/12 12:10:08:002 EDT] 00000035 ModelMgr I WSR0801I:
Initializing all server configuration models
[8/15/12 12:10:10:308 EDT] 00000035 WorkSpaceMana A WKSP0500I:
Workspace configuration consistency check is disabled.
[8/15/12 12:10:11:406 EDT] 00000035 AppManagement I
CWLDD0028I: Event id 1325375091-1. The target of the current
operation is
[WebSphere:cell=was85hostNode01Cell,node=was85hostNode01,server=se
[8/15/12 12:10:11:552 EDT] 00000035 AppManagement I
CWLDD0014I: Event id 1325375091-1. Installing application
Dynamic Cache Monitor...
[8/15/12 12:10:36:593 EDT] 00000057 webcontainer I
com.ibm.ws.webcontainer.internal.WebContainer
```



### Note

The messages that are shown in the screen capture are only some of the entries.

- \_\_\_ d. **Close** the SystemOut.log file when completed.

7. Use the administrative console to explore the Cache Monitor application settings.
- a. Using the administrative console, return to the list of deployed applications and examine the updated list of enterprise applications. If the application is not listed, refresh the administrative console.

**Enterprise Applications**

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

**Preferences**

| Start                                       | Stop                                  | Install            | Uninstall | Update | Rollout Update | Remove File | Export |
|---------------------------------------------|---------------------------------------|--------------------|-----------|--------|----------------|-------------|--------|
|                                             |                                       |                    |           |        |                |             |        |
| Select                                      | Name                                  | Application Status |           |        |                |             |        |
| You can administer the following resources: |                                       |                    |           |        |                |             |        |
| <input type="checkbox"/>                    | <a href="#">DefaultApplication</a>    |                    |           |        |                |             |        |
| <input type="checkbox"/>                    | <a href="#">Dynamic Cache Monitor</a> |                    |           |        |                |             |        |
| <input type="checkbox"/>                    | <a href="#">PlantsByWebSphere</a>     |                    |           |        |                |             |        |
| <input type="checkbox"/>                    | <a href="#">iVtApp</a>                |                    |           |        |                |             |        |
| <input type="checkbox"/>                    | <a href="#">query</a>                 |                    |           |        |                |             |        |
| Total 5                                     |                                       |                    |           |        |                |             |        |

- b. Click **Dynamic Cache Monitor**. Feel free to examine the settings for the application.



### Note

The application is now installed and in the started state. It can be configured, modified, and everything else that can be done with any other application. The only real difference is that if the EAR file is removed from the `monitoredDeployableApps` directory, the application would be uninstalled.

8. Uninstall the Cache Monitor application.
- a. Using a terminal window, navigate to the `/opt/IBM/WebSphere/AppServer/profiles/profile1/monitoredDeployableApps/servers/server1` directory.
- b. Delete the `CacheMonitor.ear` file with the following command:
- ```
rm CacheMonitor.ear
```

- __ c. Open the SystemOut.log file for **profile1**. Search for the string: cwldd
Notice the new log entries.

```

SystemOut.log (/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1/SystemOut.log)

File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace

SystemOut.log X

addPropCategorizedLayout.jsp]: Initialization successful.
[8/15/12 12:23:17:973 EDT] 00000035 WatchService I
CWLDD0007I: Event id 1325375091-2. Start of processing. Event type: Deleted, File path: /opt/IBM/WebSphere/AppServer/profiles/profile1/monitoredDeployableApps/servers/server1/CacheMonitor.ear.
[8/15/12 12:23:18:156 EDT] 00000035 AppManagement I
CWLDD0028I: Event id 1325375091-2. The target of the current operation is [WebSphere:cell=was85hostNode01Cell,node=was85hostNode01,server=se
[8/15/12 12:23:18:166 EDT] 00000035 AppManagement I
CWLDD0023I: Event id 1325375091-2. Stopping application Dynamic Cache Monitor...
[8/15/12 12:23:18:176 EDT] 00000035 AdminHelper A ADMN1010I: An attempt is made to stop the Dynamic Cache Monitor application. (User ID = defaultWIMFileBasedRealm/server:was85hostNode01Cell_was85hostNode01_server1)

```

- __ d. **Close** the SystemOut.log file when completed.
__ e. Return to the administrative console and verify that the application no longer shows up in the list of applications.

	Name	Application Status
<input type="checkbox"/>	DefaultApplication	
<input type="checkbox"/>	PlantsByWebSphere	
<input type="checkbox"/>	ivtAPP	
<input type="checkbox"/>	query	

Total 4



Information

The drag-and-drop feature works well, but only in some cases. If there is a requirement for anything but the default settings for an application deployment, this approach does not do everything that is needed. Using properties file-based configurations to do the installations helps address these additional requirements. This approach allows altering the configurations during the drag-and-drop deployments.

It is important to note that when using properties file based configurations, the drag-and-drop feature is limited to application deployments. Any additional properties that are defined in the configuration files, aside from application deployments, are ignored.

End of exercise

Exercise review and wrap-up

During the first part of the exercise, you installed the PlantsByWebSphere application from the administrative console. After installation of the PlantsByWebSphere application, it was tested. Finally, the exercise explored the process of using the monitored directory function to deploy an application.

Exercise 7. Problem determination

What this exercise is about

This exercise illustrates how to configure and view log and trace files. Both the basic mode of logging and tracing and the new HPEL mode are used. In addition to log and trace data, you learn how to gather JVM-related dump files.

What you should be able to do

At the end of this exercise, you should be able to:

- Use the administrative console to configure and view log data
- Enable a server to use HPEL
- Enable tracing on application server components
- Use the HPEL Log Viewer to examine log and trace data
- Enable verbose garbage collection for an application server
- Enable memory leak detection for an application server
- Describe how IBM Support Assistant tools can be used to analyze JVM memory dumps

Introduction

The first step in problem determination is to collect diagnostic data. This exercise focuses on how to gather runtime and application data with the tools that WebSphere provides such as logging, tracing, and JVM memory dumps. Analyzing the diagnostic data can best be done by using specific tools. Many such tools are available from the IBM Support Assistant. Though the IBM Support Assistant tools are not used in the exercise, there is a brief overview in the last section of some important tools for analyzing JVM memory dump data. This exercise also uses an example application that is called BadApp, which can demonstrate common problems that an administrator troubleshoots.

Requirements

To complete this exercise, you must have a working WebSphere Application Server **server1**, administrative console, and a running PlantsByWebSphere application that is installed on **profile1**.

-
-
-
-

Exercise instructions

Preface

This exercise focuses on how to gather diagnostic data for problem determination with tools that are part of the WebSphere Application Server V8 product. The last part of the exercise provides an overview of specific tools that can be used to help analyze the diagnostic data. All of the tools that are presented here are available in the IBM Support Assistant workbench.

Section 1: Resetting the WebSphere environment



Note

To reset your WebSphere environment, read **Appendix A** for instructions on how to complete this procedure.

Section 2: Working with log files of the application server

In this section, you examine the configuration options for logging in Basic mode.

- ___ 1. Verify that server1 is running.
 - ___ a. Using a terminal window, navigate to: <profile_root>/profile1/bin
 - ___ b. At the prompt, enter the following command:
./serverStatus.sh server1 -username wasadmin -password web1sphere
 - ___ c. If the server is not running, enter the following command:
./startServer.sh server1



Information

If the server is already started, you are challenged to provide a user ID and password when running `serverStatus` commands. Enter `wasadmin` for the user ID and `web1sphere` for the password.

- ___ 2. You can change the location, name, and other settings of log files from the administrative console.
 - ___ a. Use a web browser to start the administrative console.
 - ___ b. Click **YES** to proceed if you get a security alert.
 - ___ c. Log in. Enter `wasadmin` for the user ID and `web1sphere` for the password.

- __ d. In the navigation tree, select **Troubleshooting > Logs and trace**.
- __ e. In the pane on the right, click **server1**.



Information

You can also reach the configuration area for Logging and Tracing by selecting **Servers > Server Types > WebSphere application servers > server1**. Click **Logging and Tracing** under the Troubleshooting section.

- __ 3. Change the number of historical files and set the maximum size of the log file for System.out. The number of historical files grows from zero to the value of the **Maximum Number of Historical Log Files** field. The next rollover deletes the oldest historical file.
- __ a. Select **JVM Logs**.

The screenshot shows a web-based administration interface for a WebSphere application server. The title bar reads "Logging and tracing". Below it, a breadcrumb trail says "Logging and tracing > server1". A message box states: "It is recommended that you switch to High Performance Extensible Logging (HPEL) if you have no existing procedures that prevent you from taking advantage of it." It includes a "Switch to HPEL Mode" button and a note "(Advised for most installations)". Below this, a text box says: "Use this page to select a system log to configure, or to specify a log detail level for components and groups of components. Use log levels to control which events are processed by Java logging." A "General Properties" section has a list of links: Diagnostic Trace, **JVM Logs** (which is highlighted with a red box), Process Logs, IBM Service Logs, Change log detail levels, and NCSA access and HTTP error logging.

- ___ b. You can view and modify settings from the Logging and tracing panel for System.out and System.err logs.

General Properties

System.out

* File Name: \${SERVER_LOG_ROOT}/SystemOut.log

File Formatting: Basic (Compatible)

Log File Rotation

File Size Time

Maximum Size: 1 MB Start Time: 24

Repeat Time: 24 hours

Maximum Number of Historical Log Files. Number in range 1 through 200.
1



Information

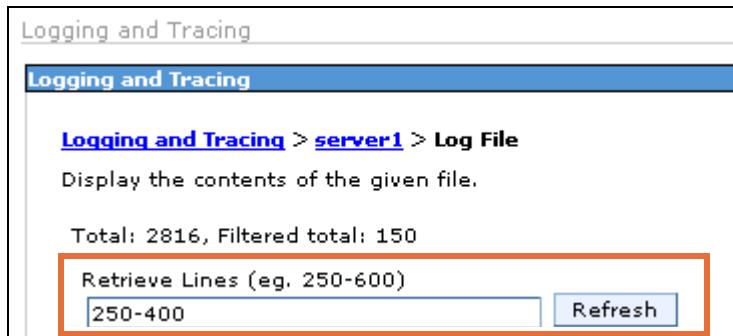
Use this page to view and modify the settings for the Java virtual machine (JVM) System.out and System.err logs for a managed process. The JVM logs are created by redirecting the System.out and System.err streams of the JVM to independent log files. The System.out log is used to monitor the health of the running application server. The System.err log contains exception stack trace information that is used to do problem analysis. One set of JVM logs exists for each application server and all of its applications. JVM logs are also created for the deployment manager and each node manager. Changes on the Configuration panel apply when the server is restarted. Changes on the Runtime panel apply immediately.

- __ c. Under General Properties for System.out, set the **Maximum Size** to 3 MB and the **Maximum Number of Historical Log Files** to 2.

The screenshot shows the 'General Properties' configuration for the 'System.out' output. The 'File Name' is set to \${SERVER_LOG_ROOT}/SystemOut.log. The 'File Formatting' is set to 'Basic (Compatible)'. In the 'Log File Rotation' section, 'File Size' is checked and 'Maximum Size' is set to 3 MB. The 'Time' rotation option is selected with a start time of 24 and a repeat time of 24 hours. The 'Maximum Number of Historical Log Files' is set to 2. Under 'Installed Application Output', 'Show application print statements' and 'Format print statements' are checked. A red box highlights the 'Maximum Number of Historical Log Files' input field.

- __ d. Click **OK**.
- __ e. **Save** the changes to the master configuration.
- __ 4. View the SystemOut.log and SystemErr.log files for server1 from the administrative console.
- __ a. Select **Troubleshooting > Logs and trace > server1 > JVM Logs** and select the **Runtime** tab.
- __ b. Click **View** to the right of the **File Name** field for **System.out**.

- ___ c. The default is to retrieve 250 lines in one step. You can specify the range of lines that are retrieved at the top of the Logging and Tracing window. Retrieve lines 250–400 by typing 250–400 and clicking Refresh.



- ___ d. Notice that several lines from the log file are shown. Scroll down to view the log records.
___ e. Click OK after you are done viewing the log records.

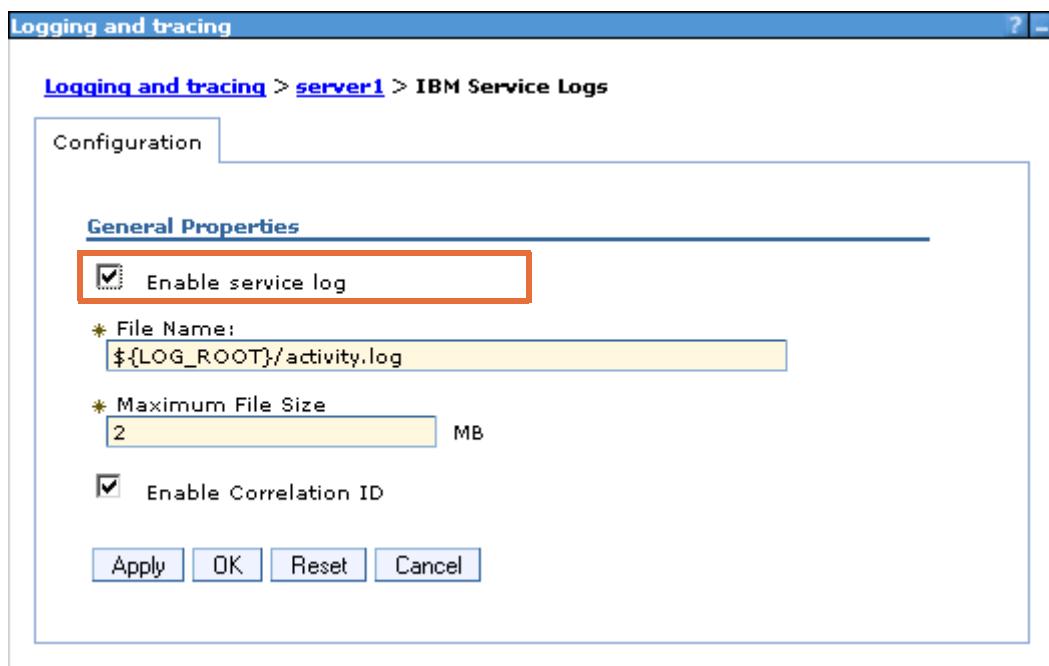


Information

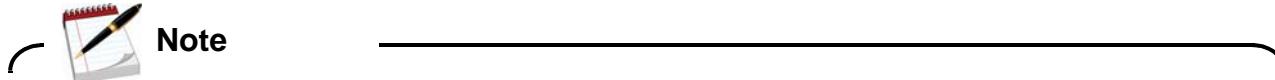
You can also go to the `<profile_root>/profile1/logs/server1` folder to view the logs with a text editor. Using an editor is preferable since you can use the search features of your text editor.

- ___ 5. Configure the IBM service logs from the administrative console. Unlike the JVM logs, the IBM service logs cannot be viewed within the administrative console. You must use a tool such as the Log Analyzer, which is available in the IBM Support Assistant.

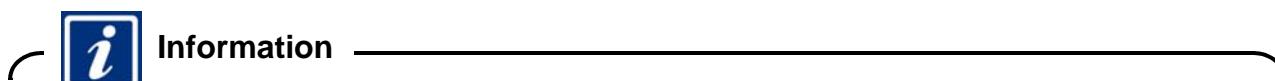
- __ a. Select **Troubleshooting > Logs and trace > server1 > IBM Service Logs.**



- __ b. Check the box for **Enable service log**.



In WebSphere Application Server Version 7 and earlier, the service log is enabled by default. In WebSphere Application Server Version 8 and later, however, the service log is disabled by default. If you do not intend to enable HPEL for version 8, then you can enable the IBM service log to be consistent with earlier versions.



The IBM service log contains both the application server messages that are written to the `System.out` stream and special messages that contain extended service information that you can use to analyze problems. One service log exists for all Java virtual machines (JVMs) on a node, including all application servers and their node agent, if present. A separate activity log is created for a deployment manager in its own logs directory. The IBM Service log is maintained in a binary format. Use the Log Analyzer or Showlog tool to view the IBM service log.

To find the value for `$(LOG_ROOT)`, you can look to **Environment > WebSphere Variables**. The name of the service log is `activity.log`, but the name can be changed along with its location in the file system. Maximum file size can be set, and you can enable or disable a correlation ID. You can use the correlation ID to correlate activity to a particular

client request. You can also use it to correlate activities on multiple application servers, if applicable.

- ___ c. Click **OK**.
- ___ d. **Save** changes.

Section 3: Set up and configure HPEL

High Performance Extensible Logging (HPEL) is a new mode of logging and tracing. To take advantage of this new log and trace framework, HPEL mode must be enabled. After HPEL mode is enabled, the JVM logs (typically `SystemOut.log` and `SystemErr.log`), the trace log (typically `trace.log`), and the service log (typically `activity.log`) are no longer written to. Instead, log and trace content is written to a log data or trace data repository in a proprietary binary format and, if configured, to a text log file. By disabling the text log file, you gain the largest possible performance benefit of HPEL. A log viewing tool, Log Viewer, is provided to allow for viewing, filtering, monitoring, and formatting the log and trace data in the repositories.

In this section, you enable HPEL mode for server1. Then, you explore and modify the log and trace configurations.

- ___ 1. Enable HPEL for server1. In the administrative console, go to the Logging and tracing panel for server1 by clicking **Troubleshooting > Logs and trace > server1**.
- ___ 2. Enable HPEL by clicking **Switch to HPEL Mode**.

- ___ 3. Click **Save** to save the configuration.

- ___ 4. Log out of the administrative console.
- ___ 5. Restart **server1**. It is necessary to restart a server for the HPEL mode to become effective.
 - ___ a. Using a terminal window, navigate to `<profile_root>/profile1/bin` and enter the following command:
`./stopServer.sh server1 -username wasadmin -password web1sphere`
 - ___ b. After server1 stops, enter the following command:
`./startServer.sh server1`



Note

Restarting server1:

You are asked to restart server1 several times throughout this exercise. To save time, keep this terminal window open, and use the keyboard up or down arrow to recall these commands.

-
- ___ 6. Configure HPEL for server1.
 - ___ a. Use a web browser to start the administrative console and log on when you are prompted.

- __ b. Go to the **Logging and tracing** panel for server1 by clicking **Troubleshooting > Logs and trace > server1**.

Logging and tracing > server1

Use this page to select a system log to configure, or to specify log detail levels for components and groups of components.

General Properties

Configure HPEL logging	
Directory	/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1
For cleanup, delete records older than	Disabled
For cleanup, maximum size of logs	50 Megabytes
Configure HPEL trace	
Directory	/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1
For cleanup, delete records older than	Disabled
For cleanup, maximum size of trace	50 Megabytes
Configure HPEL text log	
Current status:	Enabled
Directory	/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1
For cleanup, delete records older than	Disabled
For cleanup, maximum size of text log	50 Megabytes

- __ c. In the general properties section, you can see the current (default) configuration for the three HPEL repositories. Each has a directory location, and cleanup options for age and size of log files. Notice that the HPEL text log has a status of Enabled. In the next step, you disable the text log to improve server performance.
- __ 7. Disable the HPEL text log.
- __ a. In the **General Properties** section, click the link **Configure HPEL text log**.

- __ b. Clear the box for **Enable text log**.

The screenshot shows the 'Logging and tracing' configuration for 'server1'. Under 'General Properties', the 'Enable text log' checkbox is unchecked. Other settings include a directory path of \${SERVER_LOG_ROOT}, log buffering enabled, and a daily log rotation at 12 AM. In the 'Log record purging policies' section, the option to begin cleanup when the log size approaches maximum is selected.

- __ c. Click **OK**.
- __ d. Click **Save** to save the configuration.
- __ e. It is necessary to restart the server for this change to take effect. However, wait until you make some other configuration changes before you restart server1.

8. Explore the configuration for HPEL logging.
- a. Go to the General properties tab and click the link **Configure HPEL logging**.

Logging and tracing > server1 > HPEL Log Configuration

Use this page to configure High Performance Extensible Logging (HPEL) log options command (in the profile bin directory), or using the View HPEL Logs and Trace link.

General Properties

* Directory path: \${SERVER_LOG_ROOT}

Enable log record buffering

Start new log file daily at: 12 AM

Log record purging policies

Begin cleanup of oldest records when log size approaches maximum

Log record age limit: 48 Hours old

Maximum log size: 50 Megabytes

* Out of space action: Stop logging

Apply OK Reset Cancel

- b. Notice the different options for configuring HPEL logging. Also, notice that log buffering is enabled. Since buffering improves performance, it is a good practice to keep it enabled.
9. Modify the HPEL log configuration.
- a. Change **Maximum Log Size** from **50** Megabytes to **20** Megabytes.
- b. Click **OK**.
- c. Click **Save** to save the configuration.

- 10. Explore the configuration for HPEL tracing.
- a. Back in the General Properties tab, click the link **Configure HPEL trace**.

Logging and tracing

Logging and tracing > server1 > HPEL Trace Configuration

Use this page to configure High Performance Extensible Logging (HPEL) trace options. command (in the profile bin directory), or using the View HPEL Logs and Trace link.

Configuration Runtime

General Properties

Trace to a directory

Enable log record buffering

Start new log file daily at: 12 AM

Log record purging policies

Begin cleanup of oldest records
when log size approaches maximum

Log record age limit
48 Hours old

Maximum log size
50 Megabytes

* Out of space action
Purge old records

Trace to a memory buffer

* Memory Buffer Size
8 MB

* Directory to use for tracing and dumping memory buffer
\${SERVER_LOG_ROOT}

Apply OK Reset Cancel

- b. Notice that you have options to trace to a log directory or memory buffer. The default is to trace to a log directory.

___ 11. Modify the HPEL Trace configuration.

- ___ a. Under **Log record purging policies**, configure when to begin cleanup of the oldest records by using the menu to select **when oldest records reach age limit**.
- ___ b. Next, set **Log record age limit** to **12** hours.



- ___ c. Click **OK**.
- ___ d. Click **Save** to save the configuration.
- ___ e. Log out of the administrative console.
- ___ f. Restart **server1** so that the new configuration is in effect.

___ 12. Verify new HPEL configurations.

- ___ a. Log on to the administrative console.
- ___ b. Click **Troubleshooting > Logs and trace > server1**.

The screenshot shows the "Logs and tracing" configuration page for "server1". It has three main sections: "General Properties", "Configure HPEL logging", and "Configure HPEL trace".

- General Properties:** Shows a "Configure HPEL logging" section with a "For cleanup, maximum size of logs" field set to "20 Megabytes".
- Configure HPEL logging:** Shows a "Directory" field set to "/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1". It also shows "For cleanup, delete records older than" set to "Disabled" and "For cleanup, maximum size of logs" set to "20 Megabytes".
- Configure HPEL trace:** Shows a "Directory" field set to "/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1". It also shows "For cleanup, delete records older than" set to "12 Hours" and "For cleanup, maximum size of trace" set to "Disabled".
- Configure HPEL text log:** Shows a "Current status:" field set to "Disabled".

- ___ c. Verify that all the configuration changes that you made are now in effect.

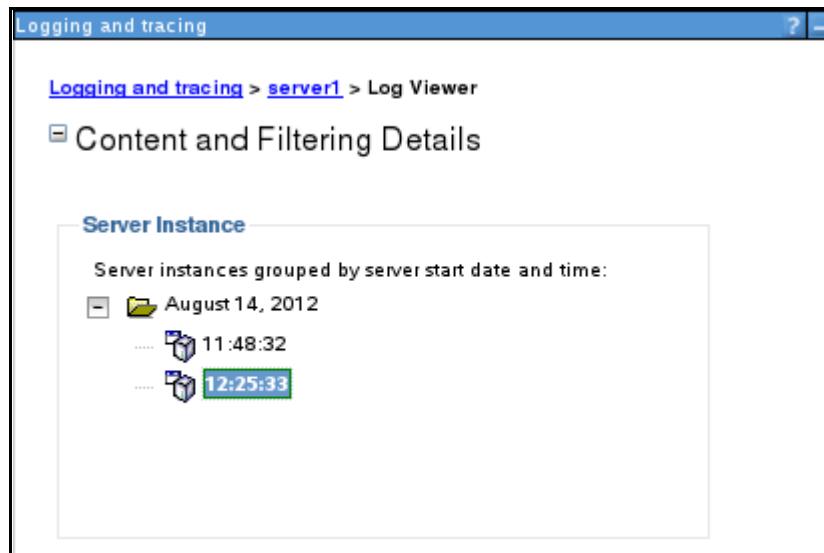
**Note****HPEL configuration**

The configuration changes you made are reasonable for the lab environment in this course. In your own testing and production environments, you must determine what configuration provides the best performance and meets your diagnostic data collection requirements.

Section 4: Use the Log Viewer in the administrative console to examine log data and trace data

In this section, you use the Log Viewer in the administrative console to examine the log messages for an application server. You use various filtering functions to customize what log records are shown.

- 1. Go to the Log Viewer for the server.
 - a. Click **Troubleshooting > Logs and trace > server1 > View HPEL Logs and Trace**.
- 2. Select the latest instance of the application server.
 - a. Expand **Content and Filtering Details**.
 - b. Expand the **Server Instance** tree, and make sure that the latest instance of the server is selected (highlighted).



**Note**

Since you recently enabled HPEL for server1, you do not see many instances of the server log repository. An instance is created for each new start of the server and designated with a time stamp. In this example, which uses the default configuration, the instances for each day are stored in a folder that is designated with the date. Any instance can be viewed by selecting it, and the log records are shown in the Log Viewer.

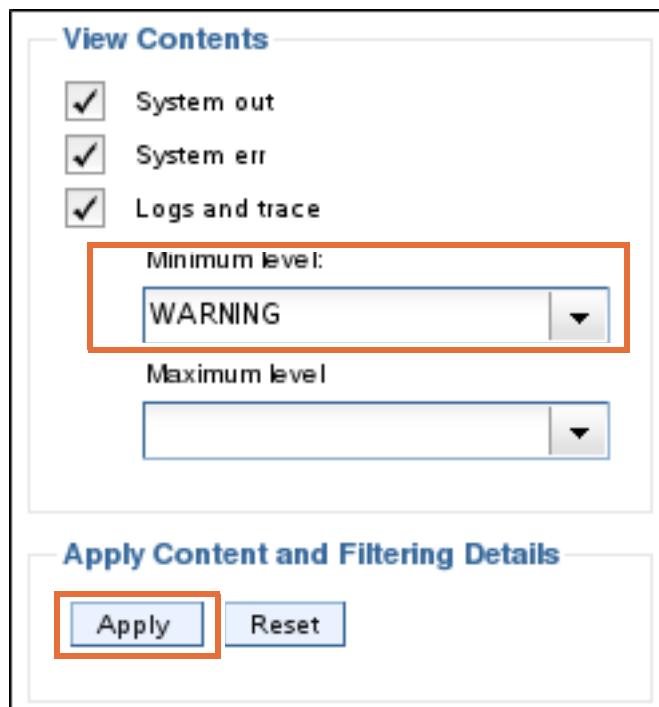
- ___ c. Collapse the **Content and Filtering Details** tree.
- ___ 3. Use the Log Viewer to explore the log records.
 - ___ a. On the Log Viewer, click **Next Page** and **Last Page** to browse though the log records. Then, click **First Page**.

The screenshot shows the 'Logging and tracing' interface with the 'Log Viewer' selected. The title bar says 'Logging and tracing > server1 > Log Viewer'. Below it, there's a section titled 'Content and Filtering Details' with a checkbox. The main area displays log records from 'September 7, 2011 10:42:10'. A table lists entries with columns: TimeStamp, Thread ID, Logger, Level, and Message. The 'Message' column contains several links (e.g., TRAS00171, WSVR08001) which are underlined and colored blue. At the bottom of the log viewer window, there are buttons for 'Refresh View', 'Show Only Selected Threads', 'Show All Threads', 'Select Columns ...', 'Export ...', 'Copy to Clipboard', and 'Server Instance Information'. Below these buttons is a dropdown for 'Number of records to show' set to 20. To the right of the log table are buttons for 'First Page', 'Previous Page', 'Next Page', and 'Last Page', all enclosed in a red rectangular box.

TimeStamp	Thread ID	Logger	Level	Message
9/7/11 10:42:10.168	00000000	zagerAdmin	INFO	TRAS00171 : The startup trace state is *=info.
9/7/11 10:42:10.184	00000000	zagerAdmin	INFO	TRAS01111 : The message IDs that are in use are deprecated
9/7/11 10:42:10.402	00000000	g.ModelMgr	INFO	WSVR08001 : Initializing core configuration models
9/7/11 10:42:11.246	00000000	etaDataMgr	INFO	WSVR01791 : The runtime provisioning feature is disabled. All components will be started.
9/7/11 10:42:11.434	00000000	iderTracker	INFO	com.ibm.ffdc.osgi.ProviderTracker AddingService FFDC10071 ; FFDC Provider Installed: com.ibm.ffdc.util.provider.FfdcOnDirProvider@20418c7
9/7/11 10:42:11.449	00000000	iderTracker	INFO	com.ibm.ffdc.osgi.ProviderTracker AddingService FFDC10071 ; FFDC Provider Installed: com.ibm.ws.ffdc.impl.FfdcProvider
9/7/11 10:42:11.949	00000000	inInitializer	AUDIT	ADMN00151 : The administration service is initialized.
9/7/11 10:42:14.230	00000000	ServiceImpl	INFO	PLGC00571 : The plug-in configuration service started successfully.

- ___ b. Try clicking one or more Message ID links to see more details about a message.

- ___ 4. Use message levels to filter records that are shown.
- ___ a. Expand **Content and Filtering Details** and select **WARNING** as the **Minimum level** in the View Contents section.



- ___ b. Click **Apply**.
- ___ c. View the records in the Log Viewer. You notice that the records that are shown have a minimum level of **WARNING**. Browse through the messages to see whether there are any at a higher level such as **SEVERE**. Alternatively, select **SEVERE** as the Minimum level in the View Contents section, and click **Apply**.
- ___ d. Clear the **Minimum** and **Maximum** level windows. Click **Apply** to see all the records again.
- ___ 5. Show all records that are associated with a specific thread.
- ___ a. Browse through the records and look for any message of interest at level **WARNING**, **AUDIT**, or **SEVERE**. Record the Thread ID for that record:
- _____

- ___ b. Highlight any record with the same Thread ID and click **Show Only Selected Threads**.

Viewing log records from server instance September 7, 2011 10:42:10				
Number of records to show: 10				
TimeStamp	Thread ID	Logger	Level	Message
9/7/11 10:42:27.230	000000000	DirectorImpl	INFO	CWRLS0012I : All persistent services (was8host01Node01Cell\was8host01
9/7/11 10:42:27.574	000000000	RceMgrImpl	INFO	WSVR0049I : Binding PGC_CF as eis/j
9/7/11 10:42:27.590	000000006	RecoveryManager	AUDIT	WTRN0132I : Transaction recovery for 277623c31c546d78e1b481511e8f5
9/7/11 10:42:27.590	000000006	RecoveryManager	AUDIT	WTRN0135I : Transaction service rec
9/7/11 10:42:27.590	000000006	RecoveryManager	AUDIT	WTRN0134I : Recovering XA resour

- ___ c. Browse through the resulting records, and you can see that only the messages from the selected Thread ID are shown. Also, notice that those records are shown in quasi-chronological order (the order in which the server emitted them).
- ___ d. After viewing the records for the selected thread, click **Show All Threads**.

Section 5: Enable tracing for an application server and view trace data from the Log Viewer

In this section, you configure tracing on the session management components of server1. Use the PlantsByWebSphere application to generate trace data, and view the trace data in the Log Viewer.

- ___ 1. Configure the diagnostic trace for the session management components of server1.
- ___ a. In the administrative console, click **Troubleshooting > Logs and trace > server1**.
- ___ b. Click **Change log detail levels**.

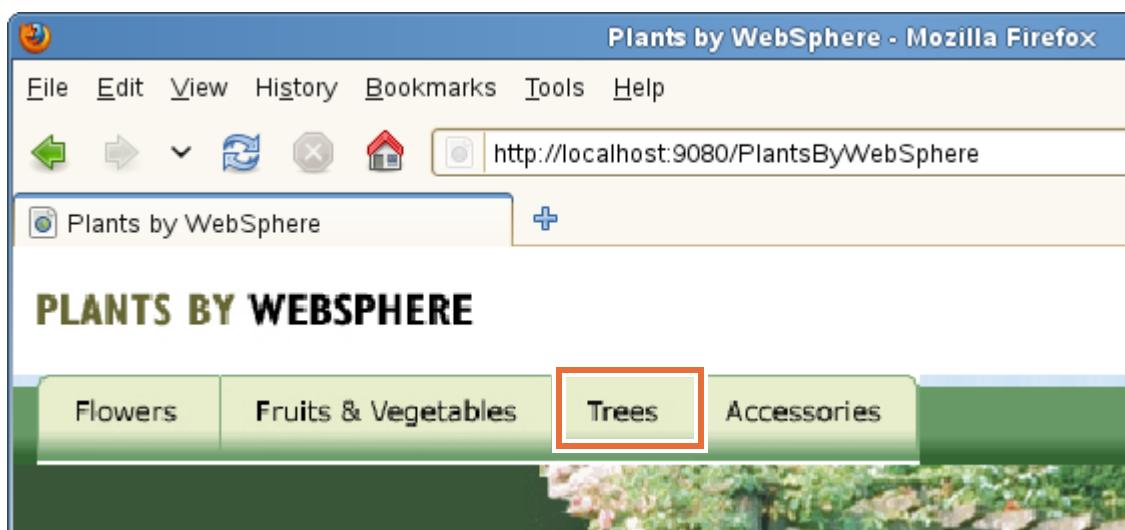


- ___ c. Select the **Runtime** tab, and enter the following trace strings. These strings are for the components and trace levels that the IBM Support MustGather documentation suggests for session management problems.

```
*=info: com.ibm.ws.session.*=all:  
com.ibm.ws.webcontainer.srt.*=all: WAS.j2c=all:  
RRA=all: WAS.database=all
```

The screenshot shows the 'Logging and tracing' configuration page for 'server1'. The 'Change log detail levels' section is expanded, revealing a log entry: *=info: *=info: com.ibm.ws.session.*=all: com.ibm.ws.webcontainer.srt.*=all: WAS.j2c=all: RRA=all: WAS.database=all. This entry is highlighted with a red rectangular box.

- ___ d. Remember to add a colon (:) after the existing trace string *=info:
 - ___ e. Click **Apply**.
- ___ 2. Access the PlantsByWebSphere application and generate some trace data.
- ___ a. Start a new browser and enter the web address:
`http://localhost:9080/PlantsByWebSphere`
 - ___ b. Click the **Trees** tab on the Welcome page.



- __ c. Select any tree that you like, and click **Add to cart**.

Item Selection:			
ITEM#	DESCRIPTION	PRICE	QUANTITY
T0004	10 gallon seedling	\$57.00	<input type="text" value="1"/>
<input type="button" value="Add to cart"/>			

- __ d. This activity is enough to generate some interesting trace data. Close the browser.
- __ 3. Use the Log Viewer in the administrative console to examine the trace data.
- __ a. Click **Troubleshooting > Logs and trace > server1 > View HPEL logs and trace**.
- __ b. Expand the **Contents and Filtering Details** tree.



Information

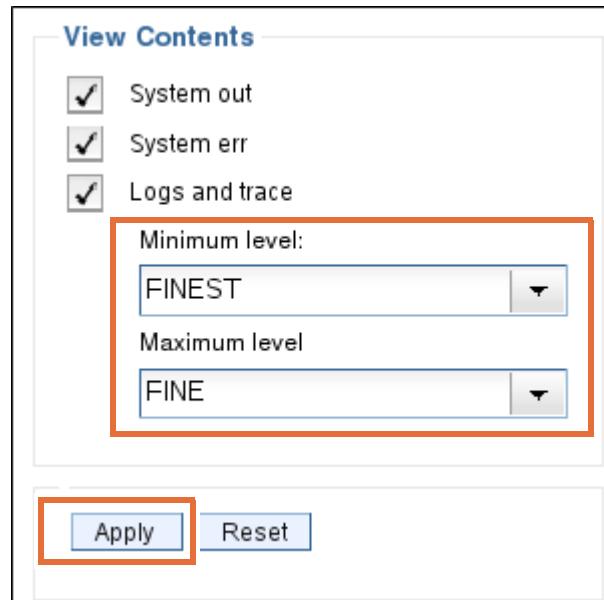
In the View Contents section, you can select **System out**, **System err**, or **Logs and trace**.

- Selecting **System out** shows records that are sent to the System Out stream with an API like `System.out.println(...)`
- Selecting **System err** shows records that are sent to the System Error stream from an API like `System.err.println(...)`
- Selecting **Logs and trace** specifies that log and trace records are included in the log view. Log and trace entries can be further specified to include a minimum or maximum level.

Examples of log and trace filters:

- Selecting **Logs and trace** and clearing minimum level and maximum level fields causes the log view to show records with any log or trace level (default).
- Selecting **Logs and trace** and setting minimum level to WARNING results in log records with levels WARNING, FATAL, or SEVERE in the log view.
- Selecting **Logs and trace** and setting maximum level to FINE results in trace records with levels FINE, FINER, or FINEST in the log view.
- Selecting **Logs and trace** and setting minimum level to DETAIL and maximum level to AUDIT results in log records with levels DETAIL, CONFIG, INFO, or AUDIT in the log view.

- ___ c. In the View Contents section, select **FINEST** as the Minimum level and **FINE** as the Maximum level.

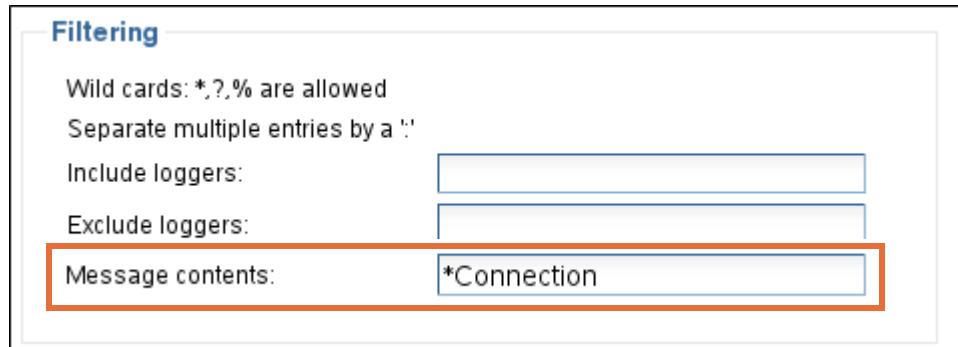


- ___ d. Click **Apply**.
___ e. Click **Refresh View**.

Viewing log records from server instance September 7, 2011 10:42:10				
Number of records to show: 20				
TimeStamp	Thread ID	Logger	Level	Message
9/7/11 14:53:47.967	0000001A	WASSessionCore	FINER	MemorySession getAttribute ENTRY WscIscLaunchedTask
9/7/11 14:53:47.967	0000001A	WASSessionCore	FINER	MemorySession getAttribute RETURN
9/7/11 14:53:47.967	0000001A	WASSessionCore	FINER	SessionContext isValid ENTRY
9/7/11 14:53:47.967	0000001A	WASSessionCore	FINER	SessionContext isValid RETURN true
9/7/11 14:53:47.967	0000001A	WASSessionCore	FINER	MemorySession getAttribute ENTRY portletCompatible
9/7/11 14:53:47.967	0000001A	WASSessionCore	FINER	MemorySession getAttribute RETURN
9/7/11 14:53:47.967	0000001A	WASSessionCore	FINER	MemorySession getAttribute ENTRY federationISCToolbarImages
9/7/11 14:53:47.967	0000001A	WASSessionCore	FINER	MemorySession getAttribute RETURN
9/7/11 14:53:47.967	0000001A	WASSessionCore	FINER	MemorySession setAttribute ENTRY {multiplier:false}
9/7/11 14:53:47.967	0000001A	WASSessionCore	FINER	MemorySession setAttribute RETURN

- ___ f. Use **Next Page** and **Previous Page** to go through the trace data. The level of all records is either **FINE**, **FINER**, or **FINEST**.
- ___ 4. Use the features of the Log Viewer to explore the trace data. Here are a few suggestions that you might try.
- ___ a. Set both Minimum level and Maximum level to **FINEST**. This setting shows you only the records at the **FINEST** level, if there are any.

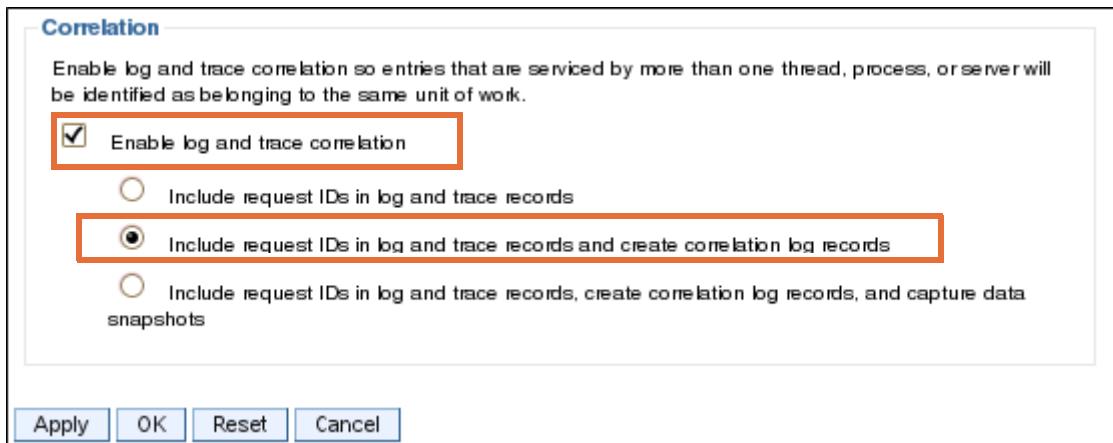
- ___ b. Select (highlight) any Thread ID of interest and click **Show Only Selected Threads**. Observe the number of different loggers that stream messages in that thread.
- ___ c. Try filtering on message content. Look for key words among the message details, such as `getConnection` or `JSESSIONID`. Use wildcards. Remember to click **Apply** and **Refresh View**.



Section 6: Enable cross-component trace (XCT)

In this section, you learn how to enable cross-component trace (XCT) for an application server. You also examine the request IDs and other data that XCT provides in the server logs.

- ___ 1. Enable XCT for server1.
 - ___ a. In the administrative console, click **Troubleshooting > Logs and trace > server1 > Change log detail levels**.
 - ___ b. On the **Runtime** tab, scroll down to the **Correlation** section.
 - ___ c. Check the box for **Enable log and trace correlation** and select **Include request IDs in log and trace records and create correlation log records**.



- ___ d. Click **OK**.



Information

Include request IDs and trace records and create correlation log records

This setting enables XCT to include request IDs in log and trace files when you want to see which log and trace entries, in all threads and application server processes, are related to the same request. Request IDs are recorded only when using HPEL log and trace mode and can be seen or used for filtering when using the `logViewer` command.

In addition, XCT creates correlation log records when you want to log how requests branch between threads and processes, and see extra information about each request.

Warning: Enabling XCT to create correlation log records might have a significant performance cost on your system, so is best suited to test and development environments.

- ___ 2. Change the trace specification to gather data for HTTP requests.
 - ___ a. Click **Troubleshooting > Logs and trace > server1 > Change log detail levels**.
 - ___ b. Select the **Runtime** tab, and enter the following trace strings.

`HTTPChannel=all: GenericBNC=all`

Configuration	Runtime
General Properties <hr/> <input type="checkbox"/> Save runtime changes to configuration as well	
Change log detail levels <hr/> <input type="checkbox"/> Disable logging and tracing of potentially sensitive data (WARNING: This might cause the log detail level setting to be modified when it is applied on the server.)	
Select components and specify a log detail level. Log detail levels specified here will apply to the entire server. Expand Components and Groups and click Components to specify a log detail level for individual components, or click Groups to specify a log detail level for a predefined group of components. Click a component or group name to select a log detail level. Log detail levels are cumulative.	
<input type="text" value="*=info: HTTPChannel=all: GenericBNC=all"/>	
<input type="button" value="Components and Groups"/>	

- ___ c. Click **OK**.
- ___ 3. Use the GUI Log Viewer to generate an HTTP request and examine the XCT data.
 - ___ a. Start a new browser and enter the web address:
`http://localhost:9080/PlantsByWebSphere`
Accessing the Plants By WebSphere Welcome page generates several HTTP requests.
 - ___ b. Log in to the administrative console.

- __ c. Click **Troubleshooting > Logs and trace > server1 > View HPEL Logs and Trace.**
- __ d. Expand the **Contents and Filtering Details** tree.
- __ e. Under the Filtering section, enter `com.ibm.websphere.XCT` in the **Include loggers** field.



- __ f. Click **Apply**. Notice that only the log records from the `com.ibm.websphere.XCT` logger are displayed.
- __ g. Click **Select columns** and select only **Message**. Clear the other columns and click **OK** to see the view that is shown in the screen capture.

Refresh View	Show Only Selected Threads	Show All Threads	Select Columns ...	Export ...	Copy to Clipboard
Viewing log records from server instance August 16, 2012 11:15:10					
Number of records to show: 20			First Page	Previous Page	
Message <pre>BEGIN AAABvgxR9nQ-AAAAAAAAAAA 00000000000-cccccccc2 HTTPCF(InboundRequest /PlantsByWebSphere RemoteAddress(127.0.0.1) RequestContext(276984898)) END AAABvgxR9nQ-AAAAAAAAAAA 00000000000-cccccccc2 HTTPCF(InboundRequest RC=302 RequestContext(276984898)) BEGIN AAABvgxR9nQ-AAAAAAAB 00000000000-cccccccc2 HTTPCF(InboundRequest /PlantsByWebSphere/promo.jsf RemoteAddress(127.0.0.1) RequestContext(276984898)) BEGIN AAABvgxR9nQ-AAAAAAAC 00000000000-cccccccc2 HTTPCF(InboundRequest /PlantsByWebSphere/javax.faces.resource/pbw.jpg.jsf RemoteAddress(127.0.0.1) RequestContext(1970068728)) BEGIN AAABvgxR9nQ-AAAAAAAAD 00000000000-cccccccc2 HTTPCF(InboundRequest /PlantsByWebSphere/javax.faces.resource/1x1_trans.gif.jsf RemoteAddress(127.0.0.1) RequestContext(296284331)) BEGIN AAABvgxR9nQ-AAAAAAAAAE 00000000000-cccccccc2 HTTPCF(InboundRequest /PlantsByWebSphere/javax.faces.resource/oamSubmit.js.jsf RemoteAddress(127.0.0.1) RequestContext(1918170885)) END AAABvgxR9nQ-AAAAAAAB 00000000000-cccccccc2 HTTPCF(InboundRequest RC=200 RequestContext(276984898)) BEGIN AAABvgxR9nQ-AAAAAAAAF 00000000000-cccccccc2 HTTPCF(InboundRequest /PlantsByWebSphere/javax.faces.resource/tab_flowers_u.gif RemoteAddress(127.0.0.1) RequestContext(1153174742)) END AAABvgxR9nQ-AAAAAAAC 00000000000-cccccccc2 HTTPCF(InboundRequest RC=304 RequestContext(1970068728)) END AAABvgxR9nQ-AAAAAAAD 00000000000-cccccccc2 HTTPCF(InboundRequest RC=304 RequestContext(296284331))</pre>					



Information

When an HTTP request arrives, the server does an XCT BEGIN, which indicates that the request is starting to process. The entry in the logs shows the following information:

- Parent XCT ID
- Current XCT ID

- Type of request (InboundRequest or OutboundRequest)
- URI of request RequestContext object
- ID from HTTPChannel
- RemoteAddress from the connection that the request originated from

When the request completes processing, the server does an XCT END; this action marks the request as finished. An XCT END for an HTTP request shows the following attributes:

- Parent XCT ID
- Current XCT ID
- The type of request, which includes InboundRequest and OutboundRequest
- Return code (RC) of the response
- HTTP Channel RequestContext object ID

4. Use the command line Log Viewer to examine the XCT data
- a. From a terminal window, navigate to:
 `/opt/IBM/WebSphere/AppServer/profiles/profile1/bin`
- b. Find the instance ID of the running server by entering the following command:
 `./logViewer.sh -listInstances`
- c. Enter the last instance ID here: _____
- d. Enter the following command:
 `./logViewer.sh -includeLoggers "com.ibm.websphere.XCT" -format advanced -instance <your_Instance_ID> > /usr/software/Troubleshooting/xct_data.txt`

- ___ e. When this command completes, the log records are written to the text file by using the advanced format. Open the `xct_data.txt` file with an editor such as gedit to explore its contents.

```
[time_stamp] 00000059 | UOW= source=com.ibm.websphere.XCT org=null prod=null component=null
thread=[WebContainer : 0] requestID=[AAABvgxR9nQ-AAAAAAAAAAAA]
    BEGIN AAABvgxR9nQ-AAAAAAAAAAA 000000000000-cccccccccc2 HTTPCF(InboundRequest
/PlantsByWebSphere RemoteAddress(127.0.0.1) RequestContext(276984898))

[time_stamp] 00000059 | UOW= source=com.ibm.websphere.XCT org=null prod=null component=null
thread=[WebContainer : 0] requestID=[AAABvgxR9nQ-AAAAAAAAAAA]
    END AAABvgxR9nQ-AAAAAAAAAAA 000000000000-cccccccccc2 HTTPCF(InboundRequest
RC=302 RequestContext(276984898))

...
    BEGIN AAABvgxR9nQ-AAAAAAAAAAE 000000000000-cccccccccc2 HTTPCF(InboundRequest
/PlantsByWebSphere/javax.faces.resource/oamSubmit.js.jsf RemoteAddress(127.0.0.1)
RequestContext(1918170885))

    END AAABvgxR9nQ-AAAAAAAAAAE 000000000000-cccccccccc2 HTTPCF(InboundRequest
RC=200 RequestContext(1918170885))
```

- ___ f. Notice that the advanced format shows the unit of work (UOW), in this case the XCT logger and the requestID.
 ___ g. Close the `xct_data.txt` file when you are finished examining it.



Information

IBM Cross Component Trace Log Viewer

Available in the IBM Support Assistant, IBM WebSphere Cross Component Trace Log Viewer provides enhanced log file views for logs that are augmented with Cross Component Trace correlation log records. Logs can be displayed in flat or hierarchical layouts, and multiple logs can be loaded and viewed simultaneously with log entries related to each request conveniently grouped.

- ___ 5. Clear the HTTP trace strings and disable XCT.
- ___ a. Click **Troubleshooting > Logs and trace > server1 > Change log detail levels.**
- ___ b. Remove the trace string: `HTTPChannel=all: GenericBNC=all`
- ___ c. Scroll down to the Correlation section and clear the box for **Enable log and trace correlation.**
- ___ d. Click **OK.**
- ___ e. Click **Save** to save the configuration.

Section 7: Collecting JVM data

There are several common JVM-related problems such as hung threads, memory leaks, and out-of-memory conditions. This section shows you how to collect diagnostic data to help troubleshoot these problems. First, you install an example application that is written to illustrate several JVM-related problems.

- ___ 1. Install the **badapp** application.
 - ___ a. In a terminal window, navigate to: <profile_root>/profile1/bin
 - ___ b. Enter the following command all on one line:


```
./wsadmin.sh -f
/usr/software/Troubleshooting/install_badapp_linux.py -username
wasadmin -password web1spHERE
```
 - ___ c. Wait until you see the following message in the command window:

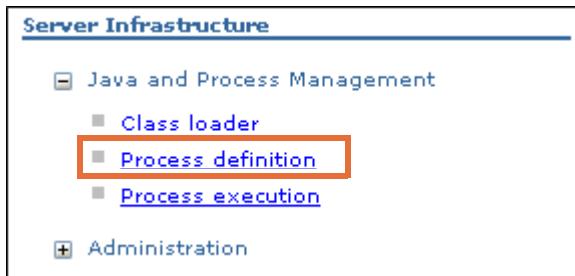

```
ADMA5013I: Application BadAppEARProject installed successfully.
```
- ___ 2. Verify that the application is installed and start it from the administrative console.
 - ___ a. Go to the administrative console and click **Applications > Application Types > WebSphere enterprise applications**.

Select	Name	Application Status
<input checked="" type="checkbox"/>	BadAppEARProject	
<input type="checkbox"/>	DefaultApplication	
<input type="checkbox"/>	PlantsByWebSphere	
<input type="checkbox"/>	iytApp	
<input type="checkbox"/>	query	

Total 5

- ___ b. Check the box for **BadAppEARProject**, and click **Start**.
- ___ c. Wait for the application to start successfully and its status is started (green arrow).
- ___ 3. Prepare server1 to log garbage collection data.
 - ___ a. Click **Servers > Server Types > WebSphere application servers > server1**.

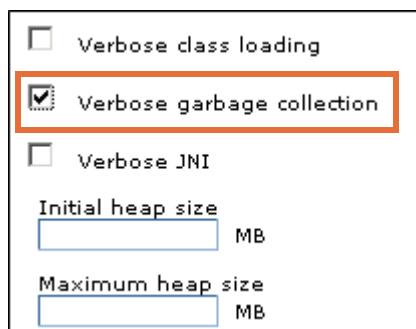
- ___ b. On the configuration tab, scroll down to **Server Infrastructure**, expand **Java and Process Management**, and click **Process definition**.



- ___ c. Click **Java Virtual Machine**.



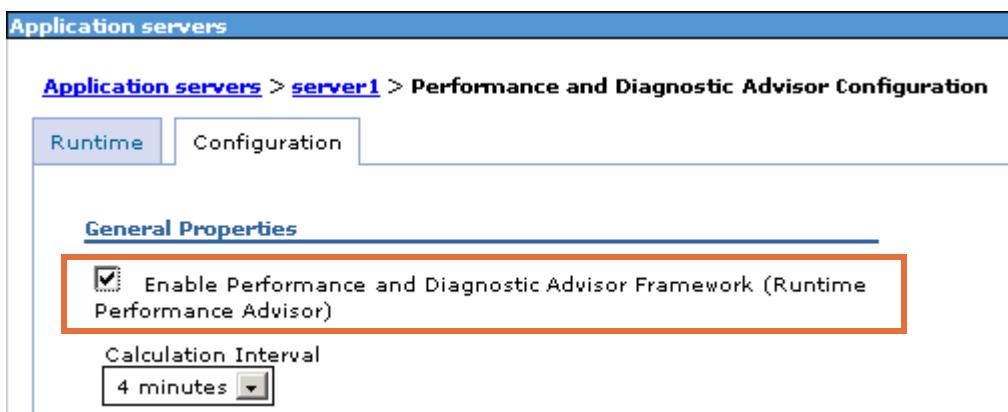
- ___ d. Check the box for **Verbose garbage collection**.



- ___ e. Click **OK**.
 ___ f. Click **Save** to save the configuration.
- ___ 4. Configure memory leak detection on server1.
- ___ a. Click **server1** in the breadcrumb trail.
 ___ b. On the configuration tab, scroll down to the **Performance** section and click **Performance and Diagnostic Advisor Configuration**.



- __ c. Check the box for **Enable Performance and Diagnostic Advisor Framework**.



- __ d. Click **OK**. Click **OK** on the warning message.
- __ e. Under **Additional Properties**, click **Performance and Diagnostic Advice configuration**.



- __ f. Scroll down on the configuration tab to the entry for **Memory Leak Rule**.

<input type="checkbox"/>	Memory Leak Rule	Jvm	Performance	Low	
--------------------------	------------------	-----	-------------	-----	--

- __ g. Verify that the status is started (solid green arrow).
- __ h. Check the box for **Memory Leak Rule**. When this rule is checked, tuning advice is written to the JVM log files for a server when a possible memory leak is detected.

<input checked="" type="checkbox"/>	Memory Leak Rule	Jvm	Performance	Low	
-------------------------------------	------------------	-----	-------------	-----	--

- __ i. Click **Save** to save the configuration.
- __ j. Log out of the administrative console, and restart **server1**.
- __ k. When you start server1 from the terminal window, notice the process ID and record it here: _____
You need the PID to terminate the server process in the next section.
- __ l. Open a new terminal window, and navigate to: `<profile_root>/profile1/bin`
- __ m. Enter the following command:

```
./logViewer.sh -monitor
```

5. Run the application.

a. From a web browser, enter the web address:

`http://was85host:9080/BadAppWebProject`

b. Enter a 5 in the **Bad Behavior Mode** window, and click **Submit**.

index.html - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Bad App Home Page

Bad Behavior Mode

Valid values for "Bad Behavior Mode" are:

- 1 - a particular condition you will be asked to analyze
- 2 - another condition for you to analyze
- 3 - a condition related to 2
- 4 - yet another condition you must analyze
- 5 - and another condition you must analyze

Submit Reset

c. The browser seems to “hang” for several minutes. During this time, examine the Log Viewer messages in the command window. You see messages similar to the following.

```
0000001e SystemOut      0  BadApp: Parameters passed from the user:
0000001e SystemOut      0  BadApp: Parameter badBehaviorMode = 5
0000001e SystemOut      0  BadApp: Request has been running for 1 minutes.

0000001e SystemOut      0  BadApp: Request has been running for 2 minutes.

0000001e SystemOut      0  BadApp: Request has been running for 3 minutes.

0000001e SystemOut      0  BadApp: Request has been running for 4 minutes.

0000001e SystemOut      0  BadApp: Request has been running for 5 minutes.

0000001e SystemOut      0  BadApp: hit an OutOfMemoryError (maybe as expec
0000001e SystemErr       R  java.lang.OutOfMemoryError: null
0000001e SystemErr       R  at com.ibm.issf.atjolin.badapp.BadAppServlet
```

- __ d. After about 5 minutes, you see the following error message in the web browser:

Return page for BadApp

Request status:OutOfMemoryException was thrown (was this error expected?) , see WebSphere Application Server logs



Information

An OutOfMemoryException is thrown. If the application was not purposely written to show the error message, you would see this symptom only if you examined the JVM logs of the server.

- __ 6. Terminate the **server1** process. It is likely that this OutOfMemory condition completely hung the application server so that it is unresponsive to a `stopServer` command. Before the `server1` process crashes, terminate it with the following command.
- __ a. From the terminal window, use the process ID that you recorded earlier, and enter: `kill -9 <PID>`

A screenshot of a terminal window titled "Terminal". The menu bar includes "File", "Edit", "View", "Terminal", "Tabs", and "Help". The main pane shows the following text:
ADMU3000I: Server server1 open for e-business; process id is 16197
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # kill -9 16197
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #

- __ 7. Examine the logs for `server1`.
- __ a. Start **server1**.
- __ b. Log in to the administrative console.
- __ c. Click **Troubleshooting > Logs and trace > server1 > View HPEL logs and trace**.

- ___ d. Expand Content and Filtering Details, and select the previous server instance.



- ___ e. In the **Filtering** section, enter `BadApp*` in the **Message contents** field.



- ___ f. Click **Apply**.

- ___ g. You now see the same messages that are shown in the command window. Select (highlight) any of these messages, and click **Show Only Selected Threads**.

Viewing log records from server instance September 8, 2011 13:45:29				
Number of records to show: 20				
TimeStamp	Thread ID	Logger	Level	Message
9/8/11 13:51:01.325	00000001E	SystemOut	DETAIL	BadApp: Parameters passed from the user:
9/8/11 13:51:01.325	00000001E	SystemOut	DETAIL	BadApp: Parameter 'badBehaviorMode' = 5
9/8/11 13:52:12.263	00000001E	SystemOut	DETAIL	BadApp: Request has been running for 1 minutes.
9/8/11 13:53:12.779	00000001E	SystemOut	DETAIL	BadApp: Request has been running for 2 minutes.
9/8/11 13:54:13.075	00000001E	SystemOut	DETAIL	BadApp: Request has been running for 3 minutes.
9/8/11 13:55:13.357	00000001E	SystemOut	DETAIL	BadApp: Request has been running for 4 minutes.
9/8/11 13:56:13.841	00000001E	SystemOut	DETAIL	BadApp: Request has been running for 5 minutes.
9/8/11 13:56:32.185	00000001E	SystemOut	DETAIL	BadApp: hit an OutOfMemoryError (maybe as expected)

- __ h. You now have several pages of log records from the selected thread. Use **Next Page** on the Log Viewer to page through these records. After you reach the last BadApp message, you will see the stack trace that resulted from the OutOfMemoryError.

SystemErr	DETAIL	java.lang.OutOfMemoryError@null
SystemErr	DETAIL	at com.ibm.issf.atjolin.badapp.BadAppServlet\$F.<init>(BadAppServlet.java:192)@null
SystemErr	DETAIL	at com.ibm.issf.atjolin.badapp.BadAppServlet.docMethod(BadAppServlet.java:406)@null
SystemErr	DETAIL	at com.ibm.issf.atjolin.badapp.BadAppServlet.doPost(BadAppServlet.java:280)@null
SystemErr	DETAIL	at javax.servlet.http.HttpServlet.service(HttpServlet.java:595)@null
SystemErr	DETAIL	at javax.servlet.http.HttpServlet.service(HttpServlet.java:668)@null
SystemErr	DETAIL	at com.ibm.ws.webcontainer.servlet.ServletWrapper.service(ServletWrapper.java:1147)@null

- __ i. In the **View Contents** section, select **WARNING** for both Minimum and Maximum levels.



- __ j. Clear the **Message content** field, and click **Apply**.
- __ k. Page through the warning messages to look for a message from the Performance Advisor about memory leaks. It is possible that you might see a message with the code `TUNE90001W`, depending on how unresponsive your server was after the OutOfMemory exception.

WARNING	<code>TUNE90001W</code> : Heap utilization patterns indicate that you may have a memory leak. Additional explanatory data follows. Data values for free memory between 9/8/11 1:57 PM and 9/8/11 1:59 PM were consistently below minimum required percentage.
---------	---

- __ l. If you do see this message, click the `TUNE90001W` link to see details.

8. Examine other JVM-related diagnostic memory dumps. Typically a server is configured to generate a `javacore` file and a heap memory dump on `OutOfMemory` exceptions. The default location for these files is the profile root directory.
- a. Using a file system explorer, go to `/opt/IBM/WebSphere/AppServer/profiles/profile1`. You can see at least one system core, `javacore`, heap dump, and Snap file; it depends on how many `OutOfMemoryError` exceptions were thrown. Notice that each memory dump includes the server PID (in this example, 6999) in the file name.

Name	Size	Type
► tranlog	1 item	folder
► workspace	10 items	folder
► wstemp	4 items	folder
core.20120816.092453.6999.0001.dmp	447.6 MB	program crash data
headdump.20120816.092453.6999.0002.phd	12.4 MB	unknown
javacore.20120816.092453.6999.0003.txt	2.0 MB	plain text document
Snap.20120816.092453.6999.0004.trc	509.9 KB	trc document



Information

Default behavior for `OutOfMemory` exceptions

WebSphere Application Server Version 8 Fix Pack 2 and later includes IBM Java 6 R26 on supported operating systems. Beginning with this version, the default behavior for `OutOfMemory` (OOM) exceptions is changed.

By default in IBM Java 5 and later, the first four OOM exceptions for the lifetime of a Java process produce:

- A PHD-formatted heap memory dump
- A Java dump file (`javacore`)
- A snap dump file

By default in IBM Java 6 R26 and later, the first OOM for the lifetime of a Java process produces:

- A PHD-formatted heap memory dump
- A Java dump file
- A snap dump file
- An operating system memory dump (core file on Linux, AIX, and IBM i, user-mode minidump with full memory on Windows operating systems, and SYSTDUMP on z/OS)

The second, third, and fourth OOM exceptions produce only a PHD-formatted heap memory dump and a Java dump file. Therefore, the change in default behavior is an extra system memory dump on the first OOM exception.

Javacore file

The javacore file, also known a thread dump file, is a text file and can be viewed with a text editor. An experienced administrator can analyze these files manually, but it is often better to use a tool such as the **IBM Thread and Monitor Dump Analyzer**. Using Thread and Monitor Dump Analyzer, you can import several Java cores that are generated over a period during which the server is hung. Thread and Monitor Dump Analyzer can then do a comparative analysis of the threads over that period.

Heap dump file

The heap dump file is a memory dump of all the Java objects on the JVM heap. It is a binary (.phd) file and must be analyzed from a tool such as **Heap Analyzer** or **Memory Analyzer**.

System core file

A system memory dump is a superset of a PHD heap memory dump. A system memory dump also includes memory contents (strings, primitives, variable names, and other objects), thread and frame local information, some native memory information, and more.

Snap file

Snap trace files are binary files that contain trace point data that is held in the trace buffers at a point in time.

- ___ b. Using a file system explorer, go to:
`/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1`
- ___ c. Use a text editor to open the `native_stderr.log` file. This file is the default location for verbose GC data.
- ___ d. Search the verbose GC data for: `JVMDUMP039I`
This search takes you to the GC data that is logged at the time of the `OutOfMemory` exception. Examine the allocation failure data just before the exception was thrown. You can see how much memory was requested (`bytesRequested`), how much heap was in use, and how much heap was free then (0 percent).
- ___ e. Close the `native_stderr.log` file when you are done examining it.



Information

Verbose garbage collection (GC) data

The verbose GC data is in text format, and an experienced administrator can analyze the file by hand. However, it is a good practice to use a tool such as **Garbage Collection and Memory Visualizer (GCMV)** to analyze GC data.

Section 8: Clean up server1

The last section concludes the active exercise. The next section is read-only. Follow these steps to clean up server1 and uninstall the BadApp application.

- ___ 1. Disable verbose garbage collection.
 - ___ a. Click **Servers > Server Types > WebSphere application servers > server1**.
 - ___ b. On the configuration tab, scroll down to **Server Infrastructure**, expand **Java and Process Management**, and click **Process definition > Java Virtual Machine**.
 - ___ c. **Clear** the box for **Verbose garbage collection**.
 - ___ d. Click **OK**.
 - ___ e. Click **Save** to save the configuration.
- ___ 2. Disable the Runtime Performance Advisor.
 - ___ a. Click **server1** in the breadcrumb trail.
 - ___ b. On the configuration tab, scroll down to the **Performance** section and click **Performance and Diagnostic Advisor Configuration**.
 - ___ c. **Clear** the box for **Enable Performance and Diagnostic Advisor Framework**.
 - ___ d. Click **OK**.
 - ___ e. Click **Save** to save the configuration.
- ___ 3. Uninstall the BadApp application.
 - ___ a. Go to the administrative console.
 - ___ b. Click **Applications > Application Types > WebSphere enterprise applications**.

- __ c. Select **BadAppEARProject** and click **Uninstall**.



- __ d. Click **OK**.
 __ e. Click **Save** to save the configuration.
 __ f. Log out of the administrative console, and restart **server1**.

Section 9: READ ONLY: Using IBM Support Assistant tools to analyze JVM data

In this section, you can see and read about the analysis of JVM diagnostic data with various tools available in the IBM Support Assistant.



Important

This section is for **reference only**. The IBM Support Assistant workbench is **not** installed on the VMware image that is used for this course. You cannot run any of the tools that are described in the following section from the VMware image.



Information

For more details and education resources about IBM Support Assistant Problem Determination Tools, go to the following websites:

- IBM Support Assistant website
<http://www.ibm.com/software/support/isa/>
- The IBM Monitoring and Diagnostic Tools for Java, Memory Analyzer
<http://www.ibm.com/developerworks/java/jdk/tools/memoryanalyzer/>
- The IBM Monitoring and Diagnostic Tools for Java, Garbage Collection, and Memory Visualizer (GCMV)
<http://www.ibm.com/developerworks/java/jdk/tools/gcmv/>

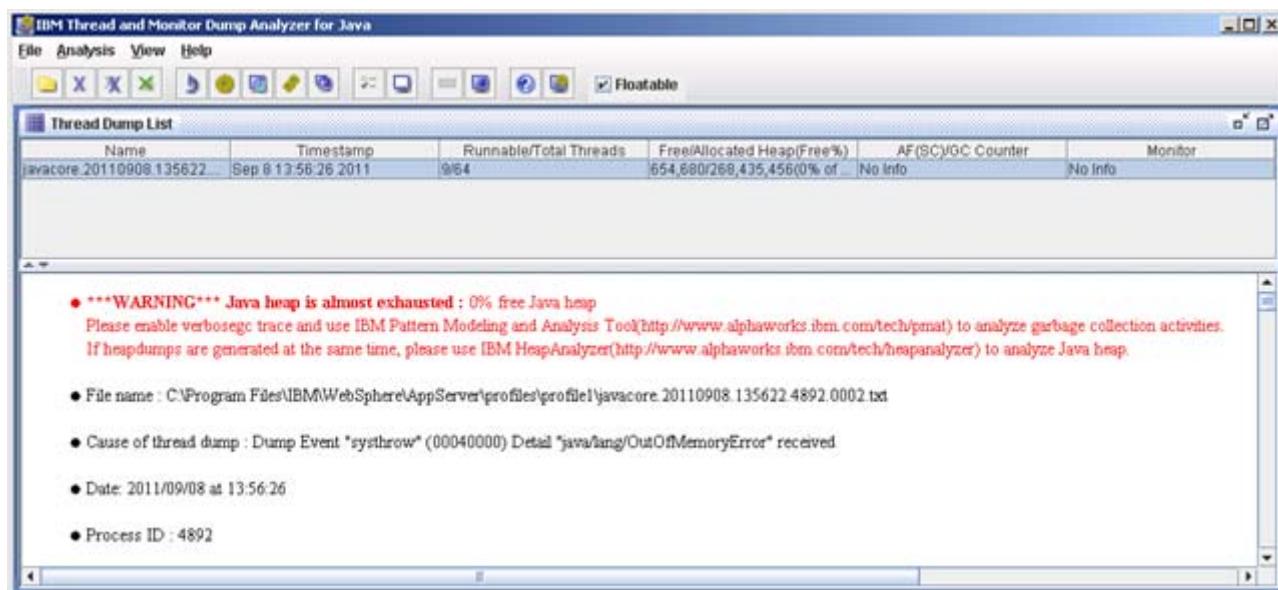
Javacore analysis

The javacore file that is dumped during the OutOfMemory condition, which the BadApp application generated, is imported into the **IBM Thread and Monitor Dump Analyzer** tool and analyzed.

IBM Thread and Monitor Dump Analyzer for Java allows you to find deadlocks, possible hung threads, and resource contention through its heuristic engine and analysis of the javacore.

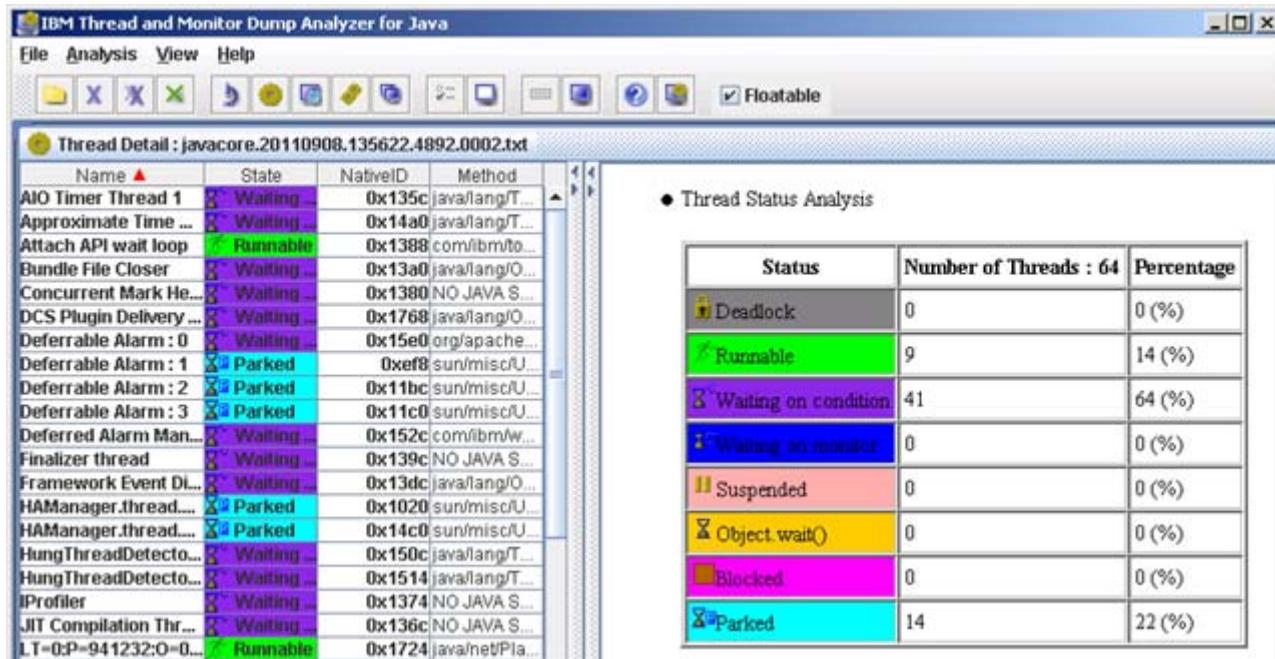
Some of the features include:

- Summary of javacore
- Thread detail view
- Monitor detail view
- List of hang suspects
- Thread compare view
- Thread comparison summary
- Monitor lock compare view
- Garbage collector statistics for IBM JVM
- Comparative analysis of multiple javacores



After importing and analyzing a javacore, the tool provides a brief report, which shows the javacore file name, the cause of the thread memory dump, and the process ID of the server instance. One or more warnings might be seen, indicating that there are deadlocked

threads or that the heap is exhausted. The warning in this screen capture shows that the Java heap is exhausted.



Right-clicking **Analysis** on the toolbar shows an analysis of the threads, monitors, or a comparative analysis of multiple javacores. This screen capture shows the thread analysis. On the left is a sortable list of all threads, their state, and the method where that thread is running. Clicking any thread name gives you a stack trace of the method and identifies what threads it is waiting on (if its state is Waiting), or what threads are possibly waiting on it. A table also summarizes the number of threads in each state.

Heap memory dump analysis

The heap dump file that the OutOfMemory condition generated from the BadApp application is imported into the Memory Analyzer tool and analyzed.

The IBM Monitoring and Diagnostic Tools for Java Memory Analyzer is a fast and feature-rich Java heap analyzer that helps you find memory leaks and reduce memory consumption.

This tool can analyze memory dumps that contain millions of objects, providing the following information:

- The retained sizes of objects
- Processes that are preventing the garbage collector from collecting objects
- A report to automatically extract leak suspects
- Suitable memory dump types include:
 - Java heap dumps
 - IBM Portable Heap Dump (.phd) binary file
 - Oracle Java Virtual Machine (JVM) hprof binary heap dump
 - System dumps that an IBM JVM generates (processed with jextract where required)
 - z/OS SVC dumps that an IBM 1.4.2 JVM generates

heapprof,20110908.135622.4892.0001.phd

Overview default_report org.eclipse.mat.api:suspects

Problem Suspect > **Leaks** > **Problem Suspect 1**

Description

The class "com.ibm.issf.atjolin.badapp.BadAppServlet", loaded by "com.ibm.ws.classloader.CompoundClassLoader @ 0x2628fb0", occupies 205,839,656 (80.48%) bytes. The memory is accumulated in one instance of "java.lang.Object[]" loaded by "com.ibm.oti.vm.BootstrapClassLoader @ 0x969920".

Keywords

com.ibm.oti.vm.BootstrapClassLoader @ 0x969920
 com.ibm.ws.classloader.CompoundClassLoader @ 0x2628fb0
 java.lang.Object[]
 com.ibm.issf.atjolin.badapp.BadAppServlet

One or more leak suspects are identified. The description of the single suspect in this analysis shows the class name and Java object, class loaders, and amount of heap occupied (80.48%).

Class Name	Shallow Heap	Retained Heap
java.lang.Object[256] @ 0xd5c2478	1,040	205,839,512
java.util.ArrayList @ 0x181e7e8	24	205,839,536
class com.ibm.issf.atjolin.badapp.BadAppServlet @ 0x3099520	64	205,839,656
↳ <Java Local> com.ibm.ws.util.ThreadPool\$Worker @ 0x19f08b8 Thread	128	128
↳ <class> com.ibm.issf.atjolin.badapp.BadAppServlet @ 0x19f1d38 »	32	48
↳ com.ibm.ws.classloader.CompoundClassLoader @ 0x2628fb0 »	152	2,440
Σ Total: 3 entries		

The **shallow heap** is the amount of memory that one object requires. A **retained set** is one or more objects plus any objects that are referenced, directly or indirectly, only from those original objects. The retained set is the set of objects that garbage collection would remove when an object, or multiple objects, are garbage collected. The **retained heap** is the total heap size of all the objects in the retained set. This value is the amount of memory that all the objects that are kept alive by the objects at the root of the retained set.

Class Name	Shallow Heap	Retained Heap	Percentage
class com.ibm.issf.atjolin.badapp.BadAppServlet @ 0x3099520	64	205,839,656	80.48%
↳ java.util.ArrayList @ 0x181e7e8	24	205,839,536	80.48%
↳ java.lang.Object[256] @ 0xd5c2478	1,040	205,839,512	80.48%
↳ com.ibm.issf.atjolin.badapp.BadAppServlet @ 0xa31250	24	1,024,072	0.40%
↳ com.ibm.issf.atjolin.badapp.BadAppServlet @ 0xa31268	24	1,024,072	0.40%
↳ com.ibm.issf.atjolin.badapp.BadAppServlet @ 0xa31280	24	1,024,072	0.40%
↳ com.ibm.issf.atjolin.badapp.BadAppServlet @ 0xa31298	24	1,024,072	0.40%
↳ com.ibm.issf.atjolin.badapp.BadAppServlet @ 0xa312b0	24	1,024,072	0.40%

A tree of accumulated objects can be expanded and examined object by object. Each object name is a link that can be clicked to reveal a menu of options for further analysis.

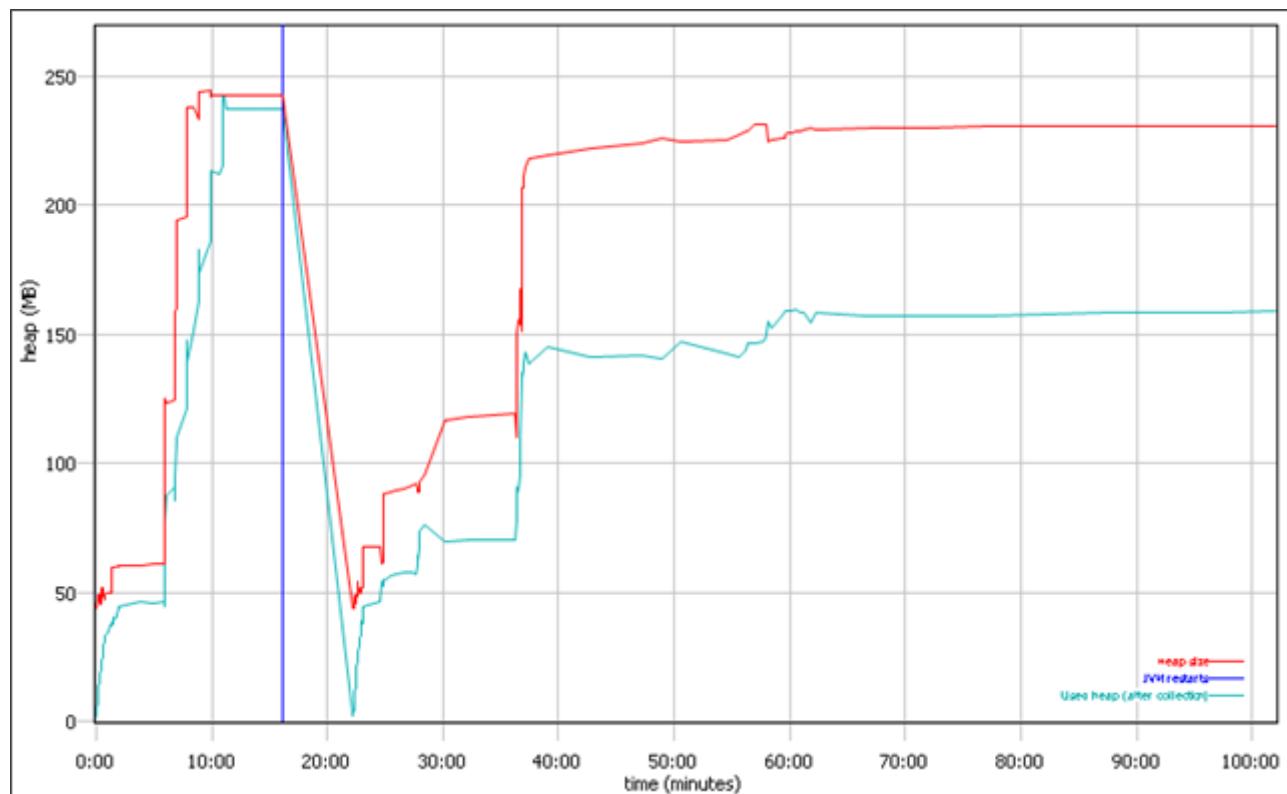
Verbose GC data

The IBM Monitoring and Diagnostic Tools for Java – Garbage Collection and Memory Visualizer is a verbose GC data visualizer. The GC and Memory Visualizer parses and plots various log types, including verbose GC logs, `-Xtgc` output, and native memory logs (output from `ps`, `svmon`, and `perfmon`).

It provides:

- A graphical display of a wide range of verbose GC data values
- Tuning recommendations and detection of problems such as memory leaks
- Report, raw log, tabulated data, and graph views
- Saving of data to HTML reports, `.jpeg` images, or `.csv` files (for export to spreadsheets)
- Viewing and comparing multiple logs

The line plot of the verbose GC data



The top line plots the heap size and the bottom line plots the used heap (after collection). The vertical line at approximately 16 minutes represents a restart of the JVM, so the data that is plotted here is for two instances of the application server. Both plots show a sudden expansion of the heap size to near the maximum heap size of 250 MB, signaling OutOfMemory exceptions.

The report and recommendations

Tuning recommendation

- ✖ The garbage collector seems to be compacting excessively, compaction is an expensive operation and leads to long pause times. On average 58% of each pause was spent compacting the heap. Compaction occurred on 77% of collections. Compaction will occur if the heap is too small or fragmented or if the heap is resized. It will also occur frequently if -Xcompactgc is specified on the command line. If fragmentation is causing compaction you should consider increasing the heap size. If compaction is occurring when the heap is resized consider fixing the heap size by setting -Xmx and -Xms to the same value as this will prevent the heap being resized automatically leading to more predictable pause times. If you have -Xcompactgc enabled and are keen to minimise pause times you may wish to remove that option from your command line. Further information about the command line options can be found in the [Diagnostics Guide](#).
- ✖ The Java Heap has been exhausted, leading to an out of memory error. You should consider increasing the Java Heap size using -Xmx if space allows. You can analyse the usage of the Java Heap for a memory leak by using the ISA Tool Add-on, IBM Monitoring and Diagnostic Tools for Java - Memory Analyzer.
- ⚠ At one point 4303 objects were queued for finalization. Using finalizers is not recommended as it can slow garbage collection and cause wasted space in the heap. Consider reviewing your application for occurrences of the finalize() method. You can use the ISA Tool Add-on, IBM Monitoring and Diagnostic Tools for Java - Memory Analyzer to list objects that are only retained through finalizers.
- ⚠ The application seems to be using some quite large objects. The largest request which triggered an allocation failure (and was recorded in the verbose gc log) was for 5120008 bytes.

The tool provides a detailed report of object utilization and recommendations for tuning garbage collection. It is possible that there are warnings about heap exhaustion, as shown in this example.

Summary of GC data

Concurrent collection count	20
Forced collection count	4
GC Mode	gencon
Global collections - Mean garbage collection pause (ms)	320
Global collections - Mean interval between collections (minutes)	0.52
Global collections - Number of collections	71
Global collections - Total amount tenured (MB)	13583
Largest memory request (bytes)	5120008
Number of collections triggered by allocation failure	297
Nursery collections - Mean garbage collection pause (ms)	22.3
Nursery collections - Mean interval between collections (ms)	25728
Nursery collections - Number of collections	250
Nursery collections - Total amount flipped (MB)	884
Nursery collections - Total amount tenured (MB)	332
Proportion of time spent in garbage collection pauses (%)	0.46
Proportion of time spent unpause (%)	99.54
Rate of garbage collection (MB/minutes)	49.1

The summary of the GC data shows the GC mode, gencon in this example, and detailed statistics about the garbage collection that are helpful for tuning the performance of a JVM. In general, tuning requires minimizing pause times and maximizing time between collections. The largest memory request data might indicate a memory leak or the use of overly large objects in the application.

End of exercise

Exercise review and wrap-up

In this exercise, you learned how to use the administrative console to configure and view log data. You also learned how to configure a server to use HPEL and set up tracing on application server components. You used the HPEL Log Viewer to examine log and trace data. To gather diagnostic JVM-related data, you configured verbose garbage collection and memory leak detection for the application server. Finally, you saw how specific IBM Support Assistant tools can be used to analyze JVM memory dumps.

Exercise 8. Using wsadmin

What this exercise is about

In this lab exercise, you use the `wsadmin` command. You learn to manipulate `wsadmin` objects from the command line and from scripts.

You also learn to create an administrative script and use command assistance in the console.

What you should be able to do

At the end of the lab, you should be able to:

- Use `wsadmin` to run administrative commands interactively and with scripts
- Create a simple administrative script
- Use console command assistance
- Use property file based configuration to modify your settings

Introduction

The WebSphere Application Server `wsadmin` tool can be used to run scripts for making configuration changes in the application server.

You can use the `wsadmin` tool to manage a WebSphere Application Server installation. This tool uses the Bean Scripting Framework (BSF), which supports various scripting languages to configure and control your WebSphere Application Server installation. The `wsadmin` tool supports the Jython and Jacl scripting languages. The Jython syntax for the `wsadmin` tool is the strategic direction for WebSphere Application Server administrative automation.

The `wsadmin` shell makes Java objects available through language-specific interfaces. Scripts use these objects for application management, configuration, operational control, and communication with MBeans running in the WebSphere server processes.

Scripting is a nongraphical alternative that you can use to configure and manage the WebSphere Application Server.

Requirements

To complete this exercise, you must have a working WebSphere Application Server installed in your working environment.

Exercise instructions

Preface

To do this exercise, you must complete the Installing IBM Installation Manager and Installing WebSphere Application Servers exercises as the exercises set up the environment that is used in this exercise.

Section 1: Resetting the WebSphere environment



Section 2: wsadmin command-line arguments

The `wsadmin` utility can be used either interactively or to run scripts. This section goes through some of the ways to start `wsadmin` and some of the command-line arguments that are available.

- 1. Confirm that the WebSphere Application Server is running.
 - a. Enter the following commands to confirm that the application server is running (if prompted to authenticate, use `wasadmin` and `web1sphere`):

```
cd /opt/IBM/WebSphere/AppServer/profiles/profile1/bin
./serverStatus.sh -all
```

```
Terminal
File Edit View Terminal Tabs Help
was85host:~ # cd /opt/IBM/WebSphere/AppServer/profiles/profile1/bin
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./serverStatus.sh
-all
ADMU0116I: Tool information is being logged in file
          /opt/IBM/WebSphere/AppServer/profiles/profile1/logs/serverStatus.log
ADMU0128I: Starting tool with the profile1 profile
ADMU0503I: Retrieving server status for all servers
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: server1
ADMU0509I: The Application Server "server1" cannot be reached. It appears to be
          stopped.
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```

- __ b. If the application server is not running, use the following command to start it:

```
./startServer.sh server1
```

A screenshot of a Windows terminal window titled "Terminal". The window has a blue title bar and a white body. The text in the window is as follows:

```
File Edit View Terminal Tabs Help  
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./startServer.sh  
server1  
ADMU0116I: Tool information is being logged in file  
        /opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1/startServer.log  
ADMU0128I: Starting tool with the profile1 profile  
ADMU3100I: Reading configuration for server: server1  
ADMU3200I: Server launched. Waiting for initialization status.  
ADMU3000I: Server server1 open for e-business; process id is 3186  
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```

- __ 2. Investigate the command-line arguments for wsadmin.

- __ a. From the bin directory, list the wsadmin command-line arguments with the following command:

```
./wsadmin.sh -help
```

A screenshot of a Windows terminal window titled "Terminal". The window has a blue title bar and a white body. The text in the window is as follows:

```
File Edit View Terminal Tabs Help  
was85host:~/profile1/bin # ./wsadmin.sh -help  
WASX7001I: wsadmin is the executable for WebSphere scripting.  
Syntax:  
  
wsadmin  
[ -h(elp) ]  
[ -? ]  
[ -c <command> ]  
[ -p <properties_file_name> ]  
[ -profile <profile_script_name> ]  
[ -f <script_file_name> ]  
[ -javaoption java_option ]  
[ -lang language ]  
[ -wsadmin_classpath class_path ]  
[ -profileName profile ]  
[ -conntype  
    SOAP  
        [ -host host_name ]  
        [ -port port_number ]  
        [ -user userid ]  
        [ -password password ] |  
    RMI  
        [ -host host_name ]  
        [ -port port_number ]  
        [ -user userid ]  
        [ -password password ] |
```

- ___ b. Notice the `-lang` argument. This argument allows you to choose which interpreter to use. Start an interactive wsadmin session with the following command:

```
./wsadmin.sh
```

- ___ c. When prompted to authenticate, use `wasadmin` for the user ID and `websphere` for the password. Depending on your environment, the prompt might open as a dialog box or in the command-line window.



Information

Your choices for the language are the following languages:

- Jacl (a version of Java to a Tcl interpreter)
- Jython (a version of Java to a Python interpreter)

Even though the default language for wsadmin is Jacl, this lab exercise focuses on Jython. The Jacl support over the last few versions of WebSphere is limited, whereas the Jython tool and support continue to be enhanced. Customers are generally advised to use Jython instead of Jacl when possible.

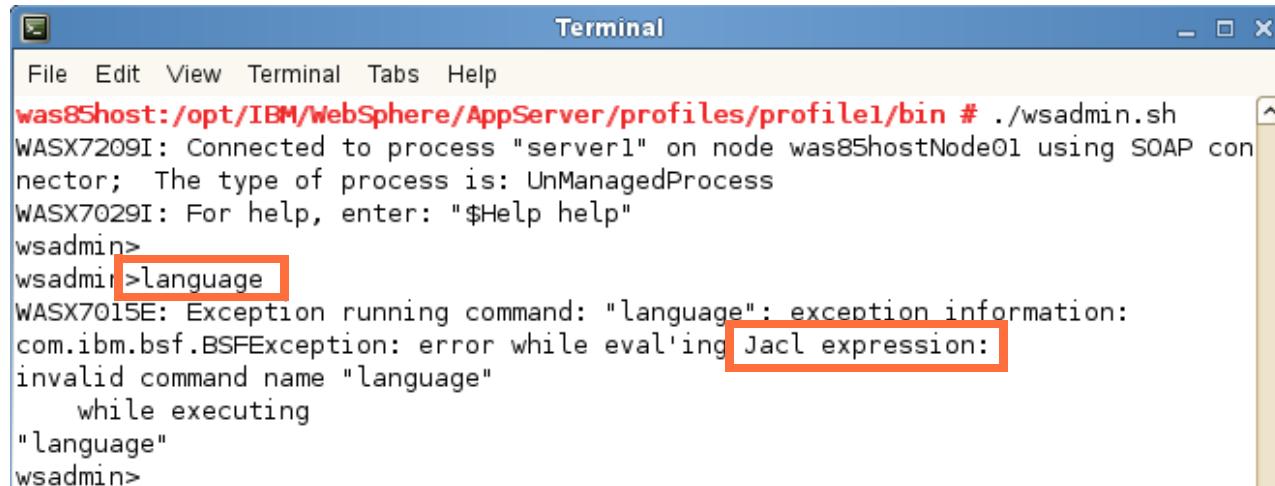
- ___ d. The startup messages do not explicitly indicate which language is used. But you can tell by the syntax of the `help` command that is displayed. The Jacl interpreter starts the `help` command with a `$` symbol.

A screenshot of a terminal window titled "Terminal". The window has a blue header bar with menu options: File, Edit, View, Terminal, Tabs, Help. Below the header is a red status bar displaying the path: "was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./wsadmin.sh". The main area of the terminal shows the following text:

```
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./wsadmin.sh
WASX7209I: Connected to process "server1" on node was85hostNode01 using SOAP connector; The type of process is: UnManagedProcess
WASX7029I: For help, enter: "$Help help"
wsadmin>
```

The line "For help, enter: \"\$Help help\"" is highlighted with a red rectangular box.

- ___ e. To confirm which language is being used, type in an unknown command. For example, try the command `language` (which is not a defined term).



The screenshot shows a terminal window titled "Terminal". The window contains the following text:

```
File Edit View Terminal Tabs Help  
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./wsadmin.sh  
WASX7209I: Connected to process "server1" on node was85hostNode01 using SOAP connector; The type of process is: UnManagedProcess  
WASX7029I: For help, enter: "$Help help"  
wsadmin>  
wsadmin>language  
WASX7015E: Exception running command: "language": exception information:  
com.ibm.bsf.BSFException: error while eval'ing Jacl expression:  
invalid command name "language"  
    while executing  
"language"  
wsadmin>
```

- ___ f. Notice that the error message indicates a problem with a "Jacl expression".

- ___ g. Use the following command to exit the wsadmin interactive mode:

```
quit
```

- ___ h. Start a new wsadmin session; use Jython as the language. Also, provide the user name and password directly on the command line. Use the following command:

```
./wsadmin.sh -lang jython -user wasadmin -password web1sphere
```



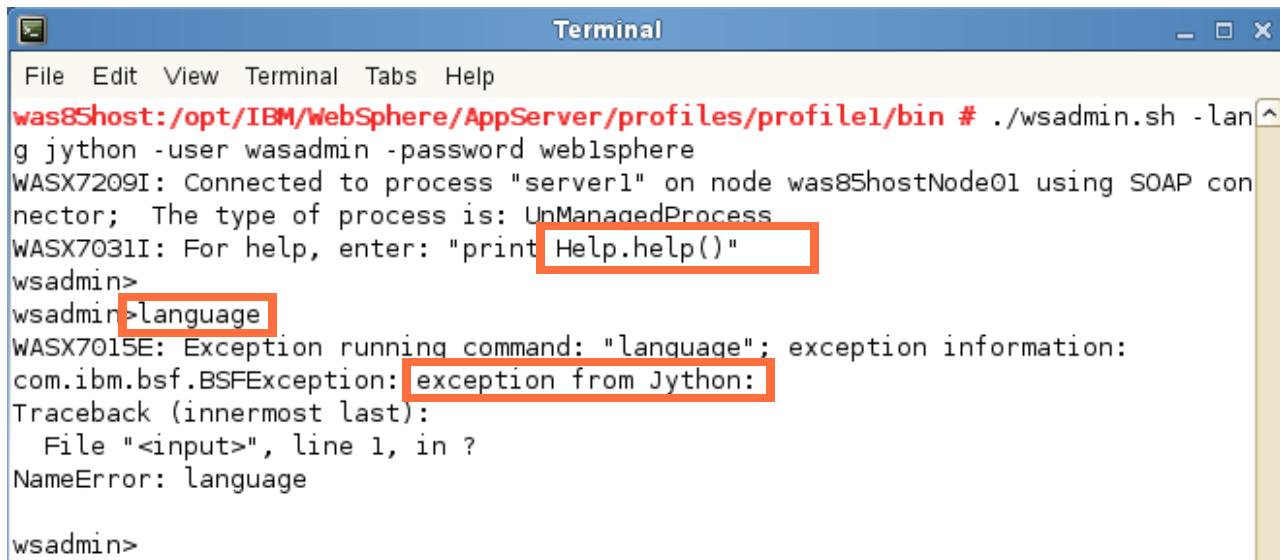
Information

The first time wsadmin is started with the Jython interpreter, you see a number of the following messages:

```
processing new jar
```

If you entered wsadmin commands in earlier exercises, you do not see the messages in this exercise.

- ___ i. Again, use the command language to see which interpreter is active.



```

File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./wsadmin.sh -lang jython -user wasadmin -password weblsphere
WASX7209I: Connected to process "server1" on node was85hostNode01 using SOAP connector; The type of process is: UnManagedProcess
WASX7031I: For help, enter: "print Help.help()"
wsadmin>
wsadmin>language
WASX7015E: Exception running command: "language"; exception information:
com.ibm.bsf.BSFException: exception from Jython:
Traceback (innermost last):
File "<input>", line 1, in ?
NameError: language
wsadmin>
```

- ___ j. Notice that you are not prompted to authenticate since the user name and password are supplied on the command line. Also, notice that the startup help command did not include a \$ symbol (the \$ flag is used by Jacl only). But finally, notice that the language command returned an error message that included "exception from Jython".



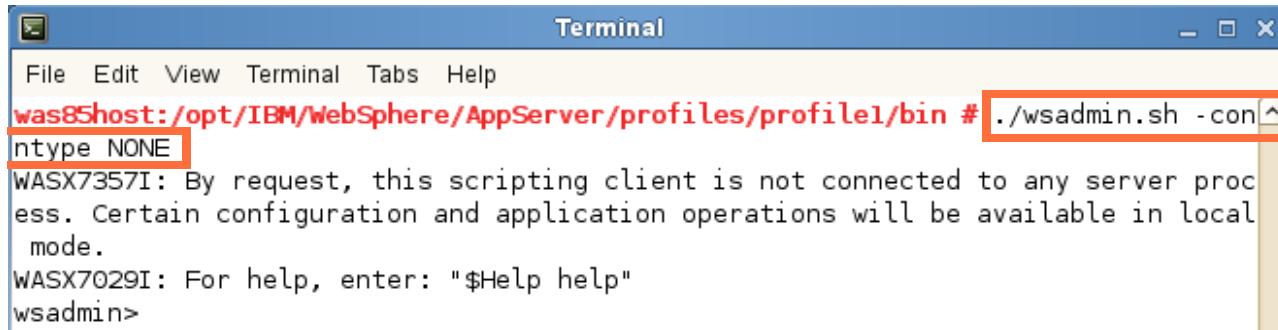
Information

The other command-line argument to take notice of is `-conntype` (connection type). This argument allows you to specify which protocol is used when the wsadmin client communicates with the application server. The default protocol is SOAP. The other choices are RMI and NONE.

It is important to understand that the wsadmin client communicates with a running application server to retrieve information and to make configuration changes. The wsadmin process is just requesting the running application server to make configuration changes on its behalf.

The `-conntype NONE` approach means that a connection to a running application server is not made. Instead, the wsadmin process reads and writes directly to the application server

configuration files, which can be useful if the application server is not starting because of a configuration problem.

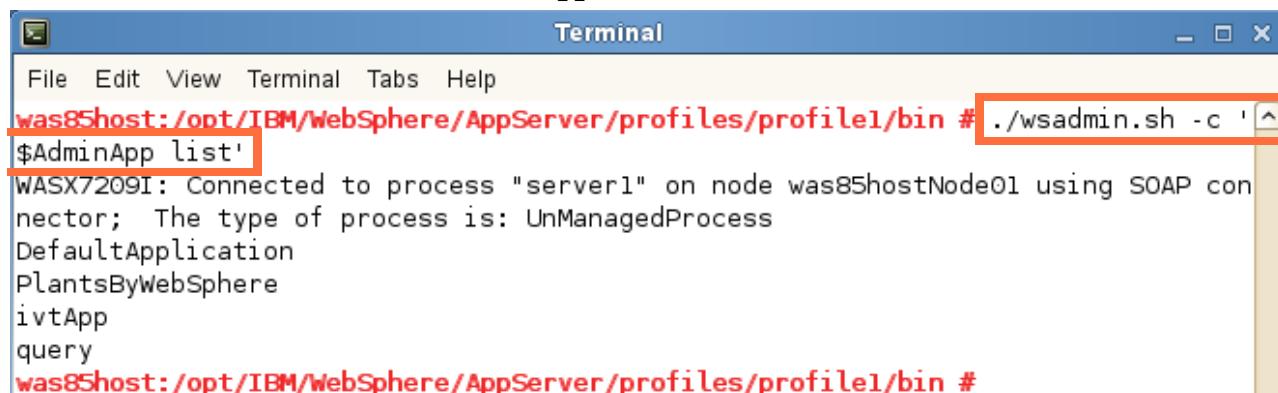


```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./wsadmin.sh -conntype NONE
WASX7357I: By request, this scripting client is not connected to any server process. Certain configuration and application operations will be available in local mode.
WASX7029I: For help, enter: "$Help help"
wsadmin>
```

Notice that when the `-conntype NONE` is used, the wsadmin process starts with a message “this scripting client is not connected to any server process.”

- ___ k. Use the following command to exit the wsadmin interactive mode:
`quit`
- ___ 3. Run wsadmin for a single command. You used wsadmin in an interactive mode, which means that you started wsadmin and are taken to an interactive prompt. This method starts wsadmin, enters a single wsadmin command, and exits out of wsadmin.
 - ___ a. To enter a single command, use `-c <command>` directly on the command line. Enter the following in the terminal window (when prompted, authenticate with wasadmin and websphere):


```
./wsadmin.sh -c '$AdminApp list'
```



```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./wsadmin.sh -c '$AdminApp list'
WASX7209I: Connected to process "server1" on node was85hostNode01 using SOAP connector; The type of process is: UnManagedProcess
DefaultApplication
PlantsByWebSphere
ivtApp
query
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```

- ___ b. Next, use the following command to enter the same command in the Jython interpreter:

```
./wsadmin.sh -lang jython -c 'AdminApp.list()'
```

A screenshot of a terminal window titled "Terminal". The window has a blue header bar with the title. Below it is a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The main area of the terminal shows the following text:

```
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./wsadmin.sh -lang jython -c 'AdminApp.list()'
WASX7209I: Connected to process "server1" on node was85hostNode01 using SOAP connector; The type of process is: UnManagedProcess
'DefaultApplication\nPlantsByWebSphere\nivtApp\nquery'
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```

- ___ c. Notice that the output is a little difficult to read. To make things easier, use the following command:

```
./wsadmin.sh -lang jython -c 'print AdminApp.list()'
```

A screenshot of a terminal window titled "Terminal". The window has a blue header bar with the title. Below it is a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The main area of the terminal shows the following text:

```
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./wsadmin.sh -lang jython -c 'print AdminApp.list()'
WASX7209I: Connected to process "server1" on node was85hostNode01 using SOAP connector; The type of process is: UnManagedProcess
DefaultApplication
PlantsByWebSphere
ivtApp
query
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```

- ___ d. Use the following command to exit the wsadmin interactive mode:

```
quit
```



Information

The wsadmin tool uses the workspace to hold configuration changes. You must save your changes to transfer the updates to the master configuration repository. If a scripting process ends and you did not save your changes, the changes are discarded. If you are using interactive mode with the wsadmin tool, you are prompted to save your changes before they are discarded. If you are using the `-c` option with the wsadmin tool, changes are automatically saved.



Information

Using the `-c <command>` approach is not efficient since it requires starting a new wsadmin instance for each command. Another approach is to use the `-f <script-file-name>` command-line argument. This approach allows you to have multiple wsadmin commands in a single file and run them as one continuous script. This approach starts a single wsadmin instance, runs all the wsadmin commands, and exits wsadmin. The load of the script approach is much less than using a number of `-c <command>` instances.

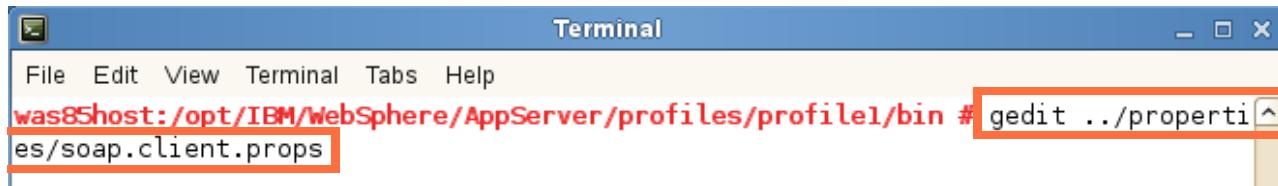
This exercise uses the `-f <script-file-name>` approach in a subsequent section.

Section 3: Configuring the wsadmin environment

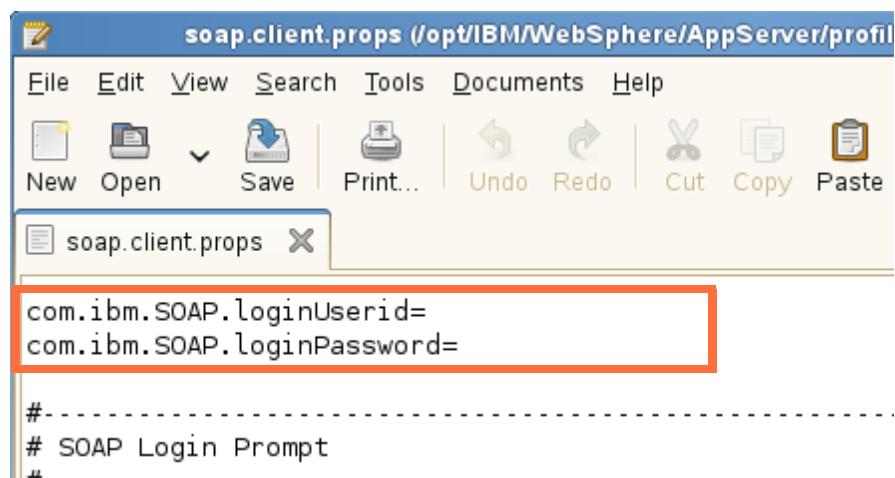
For some administrators, it can be helpful to change some of the default behaviors or the wsadmin environment.

- 1. Configure your wsadmin security. In some cases, it is helpful to avoid typing in the authentication information with every startup of wsadmin.
 - a. Use a text editor (for example, gedit) to open the profile's SOAP configuration file in the properties directory:

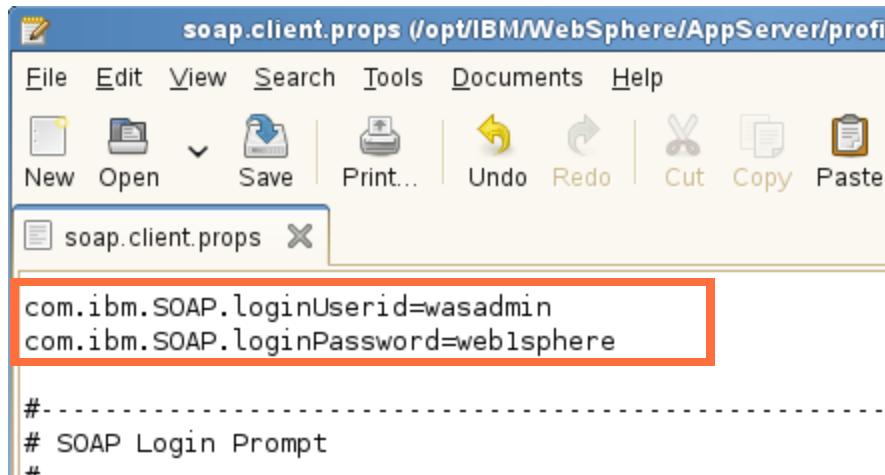
```
gedit ../properties/soap.client.props
```



- b. Scroll down and look for the authentication information for `com.ibm.SOAP.loginUserId` and `com.ibm.SOAP.loginPassword`.



- ___ c. Enter wasadmin for **com.ibm.SOAP.loginUserId** and websphere for **com.ibm.SOAP.loginPassword**. These entries cause wsadmin (and other administrative scripts such as stopServer) to automatically authenticate by using these credentials.



- ___ d. **Save** the file.
___ e. **Close** the editor.



Information

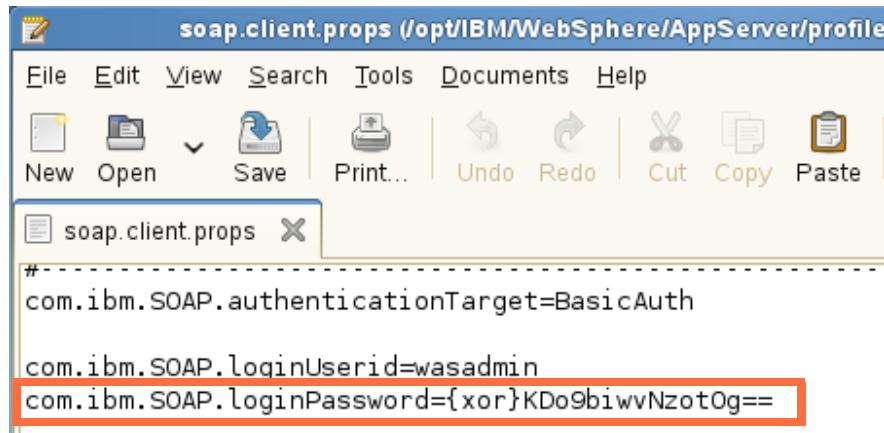
A separate properties file (`sas.client.props`) is used to configure RMI-based connections.

- ___ f. Start a new instance of wsadmin from the command line and notice that you are no longer prompted to authenticate.
___ g. Encode the password field within the properties file. This step helps obscure the password if someone happens to read the file. Enter the following command from the bin directory:

```
./PropFilePasswordEncoder.sh ../properties/soap.client.props
<space> com.ibm.SOAP.loginPassword
```

```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./PropFilePasswordEncoder.sh ../properties/soap.client.props com.ibm.SOAP.loginPassword
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```

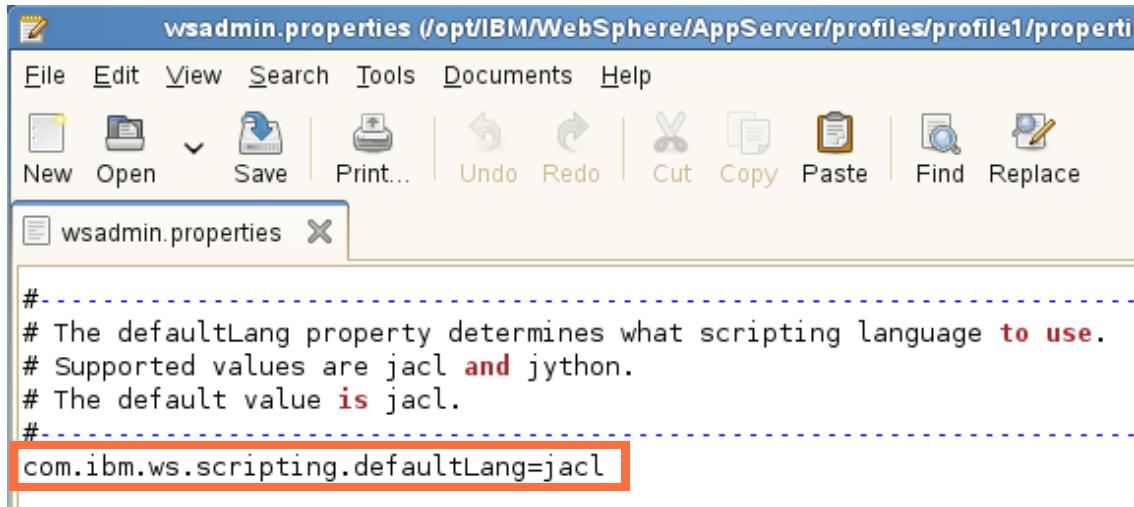
- __ h. Open the properties file again and notice that the password is now encoded.



```
soap.client.props (/opt/IBM/WebSphere/AppServer/profile1/properties)
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste
soap.client.props X
#
com.ibm.SOAP.authenticationTarget=BasicAuth
com.ibm.SOAP.loginUserId=wasadmin
com.ibm.SOAP.loginPassword={xor}KDo9biwvNzotOg==
```

- __ 2. Look at the wsadmin.properties file.

- __ a. Using a text editor, open the wsadmin.properties file in the /opt/IBM/WebSphere/AppServer/profiles/profile1/properties directory. This file defines the default values for many of the wsadmin settings.
- __ b. Scroll down and find the entry for com.ibm.ws.scripting.defaultLang. Notice that you can modify the default language wsadmin uses.



```
wsadmin.properties (/opt/IBM/WebSphere/AppServer/profiles/profile1/properties)
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
wsadmin.properties X
#
# The defaultLang property determines what scripting language to use.
# Supported values are jacl and jython.
# The default value is jacl.
#
com.ibm.ws.scripting.defaultLang=jacl
```

- __ c. Scroll through the properties file and notice that you can also change other settings that include the default host, the default port, default protocol, and other settings.
- __ d. **Close** the property file without saving any changes.

Section 4: Work with wsadmin administrative objects

In this part of the exercise, you work within the wsadmin shell in the interactive mode. When the wsadmin shell starts, you are presented with a command prompt where you can run any valid command.

- ___ 1. Start a new wsadmin session with Jython.
 - ___ a. From the terminal window, in the profile `bin` directory, enter the following command:
`./wsadmin.sh -lang jython`
- ___ 2. There are five wsadmin administrative objects. They are:
 - **AdminControl**: manages runtime objects
 - **AdminConfig**: manages configuration object
 - **AdminApp**: manages the installation of applications
 - **AdminTask**: provides another task-oriented interface to all objects
 - **Help**: provides help

Enter the following command to get general help:

```
print Help.help()
```



Information

The purpose of this section is not to try to teach wsadmin syntax or how to script in Jython. Covering those topics would take much too long. The intention here is to provide a brief introduction to what some of the commands look like.

For more in-depth information, there is a wsadmin scripting course available. Additionally, there is a wealth of information available in the information center.

There is also a book titled *WebSphere Application Server Administration Using Jython* available through IBM Press. It can be found at:

<http://www.ibmpressbooks.com/bookstore/product.asp?isbn=0137009526>

- ___ 3. Investigate the AdminConfig object. The AdminConfig object allows you to access and change the configuration information that is associated with a WebSphere Application Server environment.
 - ___ a. Start by looking at the help specific to the AdminConfig object with the command:
`print AdminConfig.help()`
 - ___ b. Notice that the help lists various commands that can be run. Investigate the AdminConfig methods by typing in the following commands:
`print AdminConfig.list('DataSource')`
`print AdminConfig.attributes('DataSource')`

___ 4. Investigate the AdminControl object. The AdminControl object allows you to access and change the running objects within a WebSphere Application Server environment.

___ a. Start with looking at the help specific to the AdminControl object with the command:

```
print AdminControl.help()
```

___ b. Notice that the help lists various commands that can be run. Enter a couple of the AdminControl methods by typing in following commands:

```
print AdminControl.getHost()  
print AdminConfig.getPort()
```

___ 5. Investigate the AdminApp object. The AdminApp object allows you to access and change the installed applications within a WebSphere Application Server environment.

___ a. Start with looking at the help specific to the AdminApp object with the command:

```
print AdminApp.help()
```

___ b. Notice that the help lists various commands that can be run. Enter a couple of the AdminApp methods by typing in following commands:

```
print AdminApp.list()  
print AdminApp.listModules("PlantsByWebSphere")
```

___ 6. Investigate the AdminTask object. The AdminTask object is different from the previous Admin objects. AdminTask provides many task-oriented methods that make configuration and management of the WebSphere Application Server environment simpler. Instead of using numerous complex AdminConfig, AdminControl, and AdminApp steps, the AdminTask methods can accomplish the same things but in much easier ways.

___ a. Start with looking at the help specific to the AdminTask object with the command:

```
print AdminTask.help()
```

___ b. To get a list of available AdminTask commands, use the following command:

```
print AdminTask.help('-commands')
```



Note

Use the following command to get a count of the number of AdminTask commands (over 1400 are listed in V8.5.0.0):

```
./wsadmin.sh -lang jython -c "print AdminTask.help(' -commands ') | wc -l"
```

- __ c. Investigate AdminTask methods by typing in following commands:

```
print AdminTask.listNodes()
print AdminTask.listServers()
print AdminTask.listKeyStores()
```



Information

Many AdminTask functions support an interactive mode. This mode can be helpful since it prompts the user for each argument. It then creates and displays the final non-interactive command, which can be used when creating a script. For example, type in the following command:

```
AdminTask.listDatasources('[-interactive]')
```

Enter Node=was85hostNode01 for the scope. Choose F to complete the operation.

```
Terminal
File Edit View Terminal Tabs Help
wsadmin: AdminTask.listDatasources('[-interactive]')
List the specified datasources.

List the datasources that are contained in the specified scope.

Scope string (scope): Node=was85hostNode01
List the specified datasources.

F (Finish)
C (Cancel)

Select [F, C]: [F]
WASX7278I: Generated command line: AdminTask.listDatasources('[-scope Node=was85hostNode01]')
'Plants(cells/localhostNode01Cell/nodes/was85hostNode01/resources.xml#DataSource_1345074713612)'
wsadmin>
```

Section 5: Creating a wsadmin script

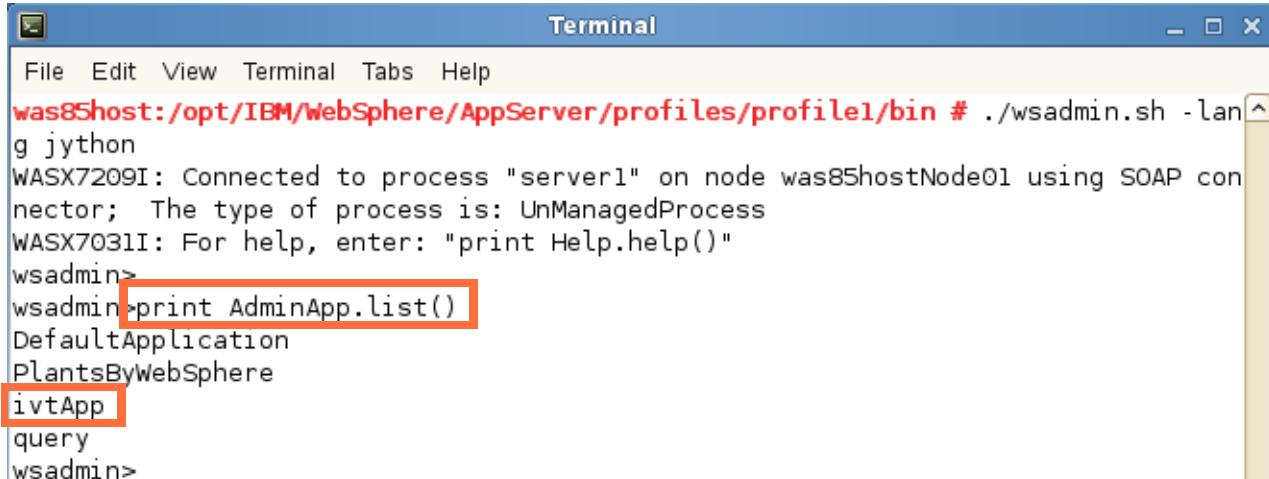
The wsadmin syntax and interface are not intended for interactive use since they are too complex for administrators to remember all of the details. The true power comes with creating reusable scripts to complete your necessary operations.

This section of the exercise goes through the process of creating a relatively simple script to install an application. To help make it simple, the ivtApp is used for this demonstration.

___ 1. Start by uninstalling the ivtApp.

___ a. From wsadmin instance with the Jython interpreter, use the following command to verify that the ivtApp is installed (feel free to copy and paste from /usr/software/wsadmin/exercise-instructions.py):

```
print AdminApp.list()
```



A screenshot of a terminal window titled "Terminal". The window shows the following text:

```
File Edit View Terminal Tabs Help  
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./wsadmin.sh -lang jython  
WASX7209I: Connected to process "server1" on node was85hostNode01 using SOAP connector; The type of process is: UnManagedProcess  
WASX7031I: For help, enter: "print Help.help()"  
wsadmin>  
wsadmin>print AdminApp.list()  
DefaultApplication  
PlantsByWebSphere  
ivtApp  
query  
wsadmin>
```

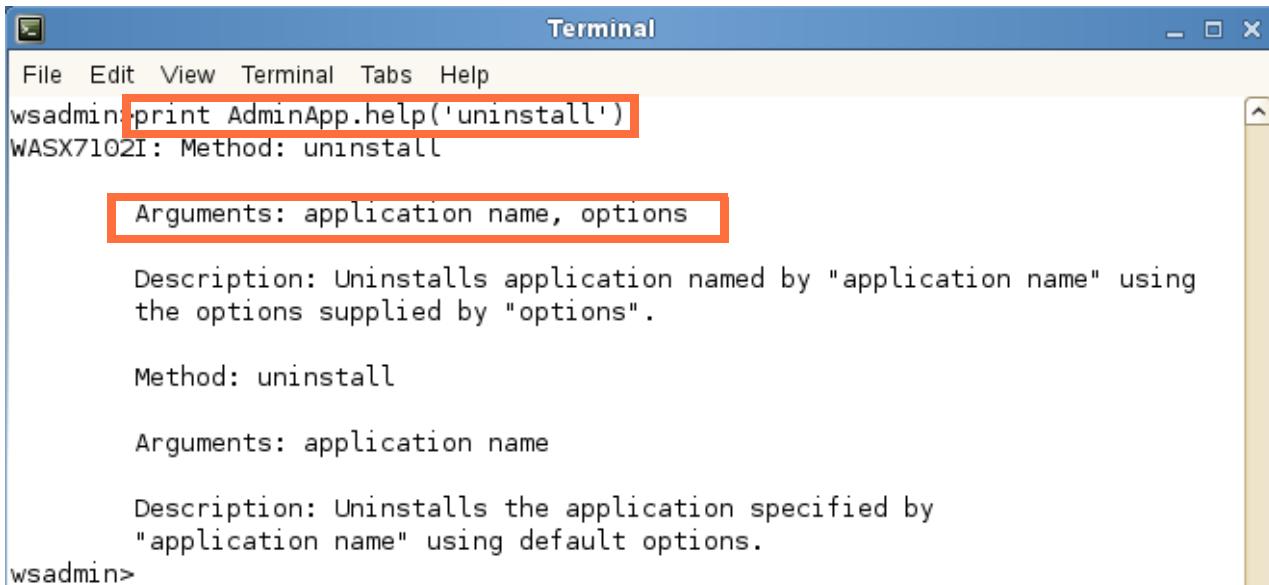
The command `print AdminApp.list()` is highlighted with a red box. The result `ivtApp` is also highlighted with a red box.

___ b. Use the following command to verify that there is an AdminApp function to uninstall applications:

```
print AdminApp.help()
```

___ c. To get more information about the specific syntax, use the following command:

```
print AdminApp.help('uninstall')
```



A screenshot of a terminal window titled "Terminal". The window shows the following text:

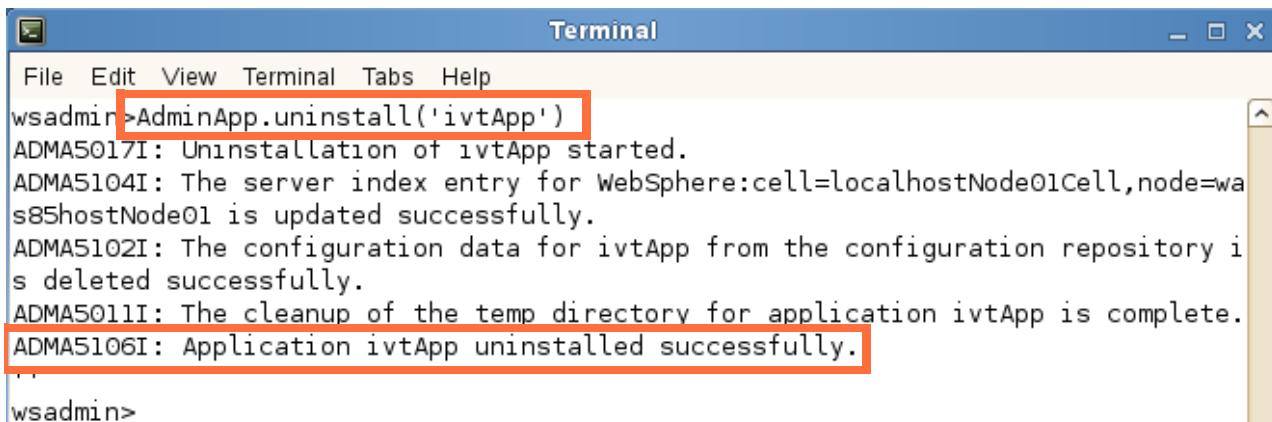
```
File Edit View Terminal Tabs Help  
wsadmin>print AdminApp.help('uninstall')  
WASX7102I: Method: uninstall  
  
Arguments: application name, options  
  
Description: Uninstalls application named by "application name" using the options supplied by "options".  
  
Method: uninstall  
  
Arguments: application name  
  
Description: Uninstalls the application specified by "application name" using default options.  
wsadmin>
```

The command `print AdminApp.help('uninstall')` is highlighted with a red box. The argument `application name, options` is highlighted with a red box.

___ d. Notice that the only required argument is the name of the application.

- ___ e. Use the following command to uninstall the ivtApp:

```
AdminApp.uninstall('ivtApp')
```



```
Terminal
File Edit View Terminal Tabs Help
wsadmin>AdminApp.uninstall('ivtApp')
ADMA5017I: Uninstallation of ivtApp started.
ADMA5104I: The server index entry for WebSphere:cell=localhostNode01Cell,node=wa
s85hostNode01 is updated successfully.
ADMA5102I: The configuration data for ivtApp from the configuration repository i
s deleted successfully.
ADMA5011I: The cleanup of the temp directory for application ivtApp is complete.
ADMA5106I: Application ivtApp uninstalled successfully.

wsadmin>
```

- ___ f. Since the command changed a configuration file, a save is required. Use the following command to save the updates:

```
AdminConfig.save()
```

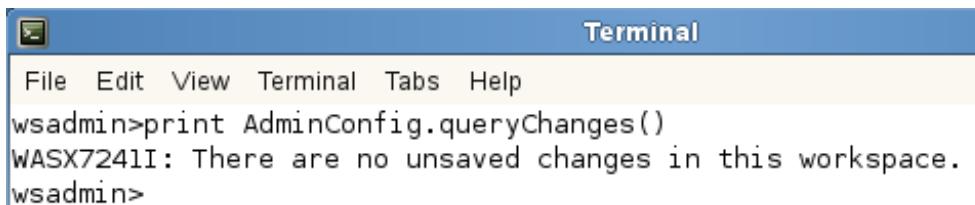


Information

You can use the following command to see whether a save command is needed:

```
print AdminConfig.queryChanges()
```

A list of changed files would be listed if a save is needed. If no save is needed, the message **There are no unsaved changes in this workspace.** shows.



```
Terminal
File Edit View Terminal Tabs Help
wsadmin>print AdminConfig.queryChanges()
WASX7241I: There are no unsaved changes in this workspace.
wsadmin>
```

2. Use the administrative console to verify that the ivtApp is no longer installed.

- a. Click Applications > Application Types > WebSphere enterprise applications.

Enterprise Applications

Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

[+] Preferences

Start Stop Install Uninstall Update Rollout Update Remove File Export

Select Name Application Status

You can administer the following resources:

<input type="checkbox"/>	DefaultApplication	
<input type="checkbox"/>	PlantsByWebSphere	
<input type="checkbox"/>	query	

Total 3

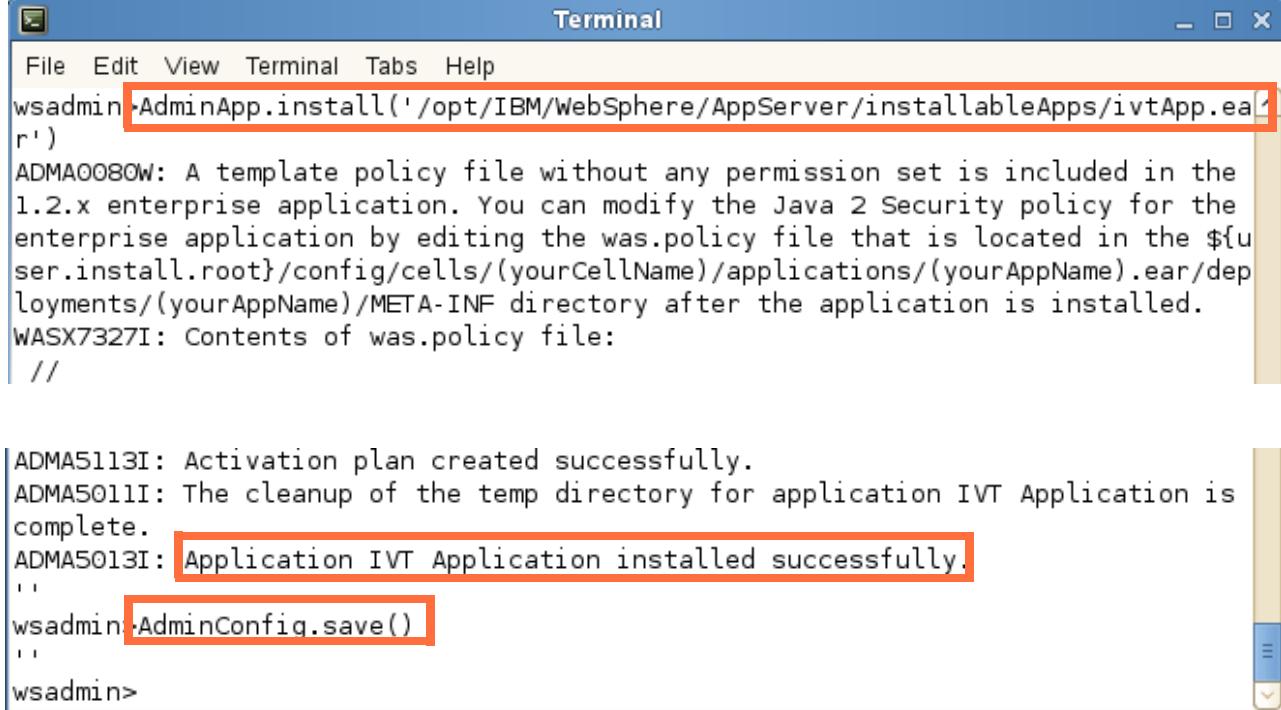
3. Use wsadmin to install the ivtApp manually.

- a. Return to the interactive wsadmin session and use the following commands to verify that the installation command exists and what arguments are needed:

```
print AdminApp.help()  
print AdminApp.help('install')
```

- ___ b. Now that you see only the EAR file is needed, use the following commands to install the ivtApp:

```
AdminApp.install('/opt/IBM/WebSphere/AppServer/installableApps/ivtApp.ear')
AdminConfig.save()
```



A screenshot of a Windows terminal window titled "Terminal". The window contains the following text:

```
File Edit View Terminal Tabs Help
wsadmin>AdminApp.install('/opt/IBM/WebSphere/AppServer/installableApps/ivtApp.ear')
ADMA0080W: A template policy file without any permission set is included in the
1.2.x enterprise application. You can modify the Java 2 Security policy for the
enterprise application by editing the was.policy file that is located in the ${u
ser.install.root}/config/cells/(yourCellName)/applications/(yourAppName).ear/de
ployments/(yourAppName)/META-INF directory after the application is installed.
WASX7327I: Contents of was.policy file:
//
```

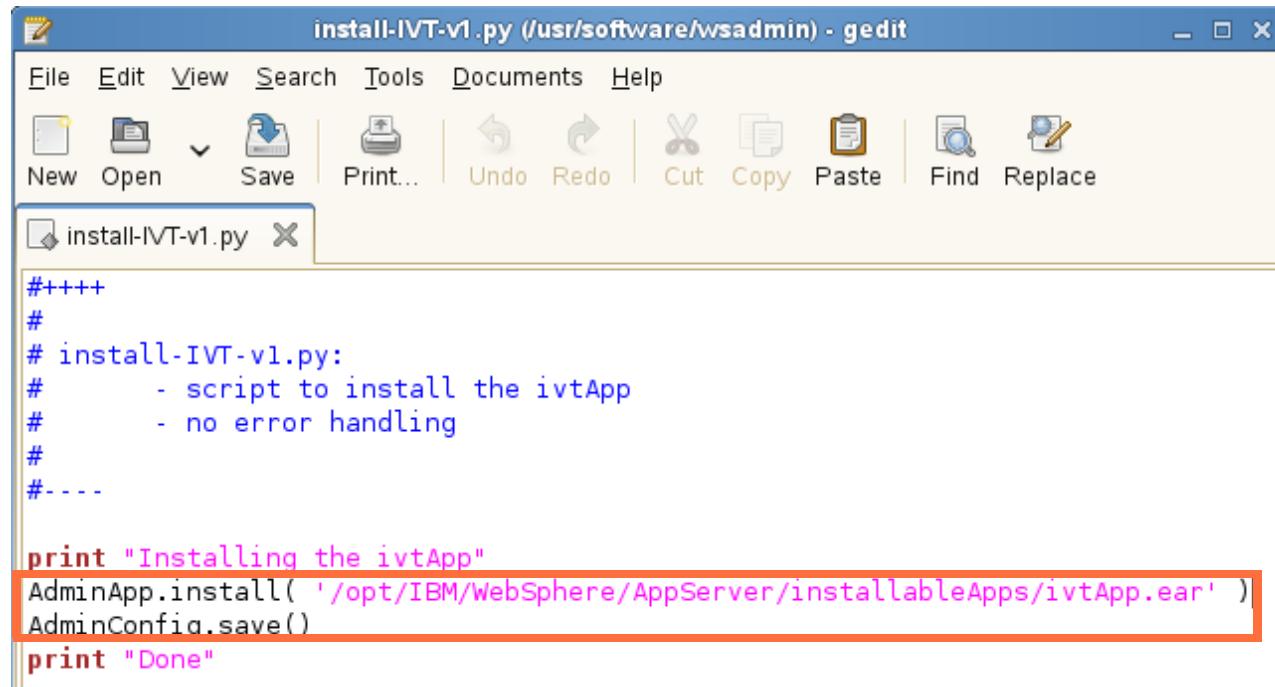


```
ADMA5113I: Activation plan created successfully.
ADMA5011I: The cleanup of the temp directory for application IVT Application is
complete.
ADMA5013I: Application IVT Application installed successfully.
//
wsadmin>AdminConfig.save()
//
wsadmin>
```

- ___ c. Confirm that the ivtApp is installed by refreshing the administrative console or by using the command:

```
print AdminApp.list()
```

- ___ 4. Automate the installation by creating a script to install the ivtApp. To create this script, use the commands that you used in the previous step and add them to a script file.
- ___ a. To save time, the script is already created for you. Using an editor such as gedit, open /usr/software/wsadmin/install-IVT-v1.py.



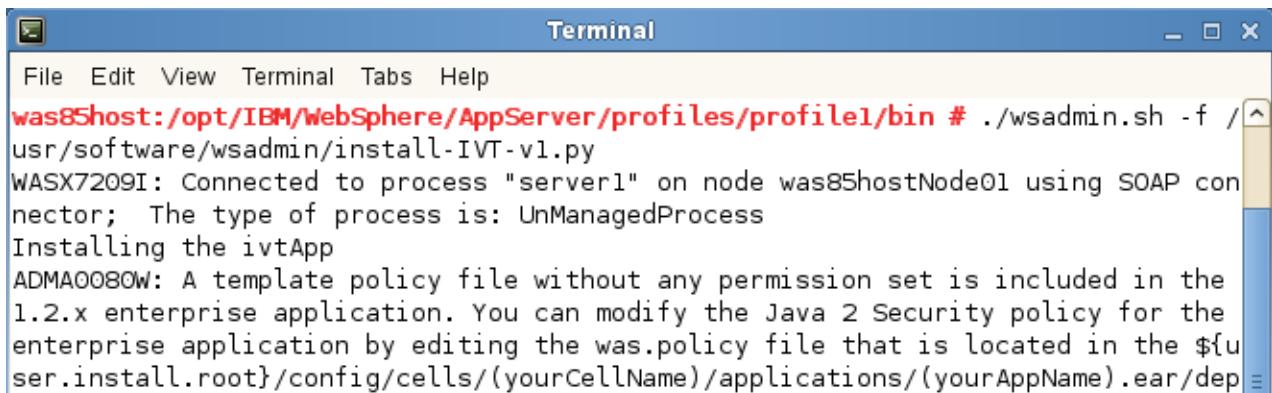
The screenshot shows a Gedit window titled "install-IVT-v1.py (/usr/software/wsadmin) - gedit". The window contains Python code for installing the ivtApp. The code includes comments explaining the purpose of the script and its lack of error handling. It uses the AdminApp.install() method to install the application from a specific path, and AdminConfig.save() to persist the changes. Finally, it prints "Done" to the console.

```
#+++++
#
# install-IVT-v1.py:
#     - script to install the ivtApp
#     - no error handling
#
#-----
#
print "Installing the ivtApp"
AdminApp.install( '/opt/IBM/WebSphere/AppServer/installableApps/ivtApp.ear' )
AdminConfig.save()
print "Done"
```

- ___ b. Notice that the commands that you typed in manually during the previous step are duplicated here in the script.

- ___ c. Open a new terminal window and navigate to the profile bin directory. Enter the following command to run the script:

```
./wsadmin.sh -f /usr/software/wsadmin/install-IVT-v1.py
```



A screenshot of a terminal window titled "Terminal". The window shows the command `./wsadmin.sh -f /usr/software/wsadmin/install-IVT-v1.py` being run. The output includes connection information, a warning about a template policy file, and an error message indicating that an application named "IVT Application" already exists.

```
File Edit View Terminal Tabs Help  
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./wsadmin.sh -f /  
usr/software/wsadmin/install-IVT-v1.py  
WASX7209I: Connected to process "server1" on node was85hostNode01 using SOAP con  
nector; The type of process is: UnManagedProcess  
Installing the ivtApp  
ADMA0080W: A template policy file without any permission set is included in the  
1.2.x enterprise application. You can modify the Java 2 Security policy for the  
enterprise application by editing the was.policy file that is located in the ${u  
ser.install.root}/config/cells/(yourCellName)/applications/(yourAppName).ear/dep
```

```
WASX7017E: Exception received while running file "/usr/software/wsadmin/install-  
IVT-v1.py"; exception information: com.ibm.ws.scripting.ScriptingException: WASX  
7279E: An application with name "IVT Application" already exists. Select a diffe  
rent name.
```

```
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```

- ___ d. Notice that this script failed because the ivtApp is already installed.



Note

Notice that you were not required to use the `-lang` command option because `wsadmin` recognizes the `.py` extension as a Jython script.

Also, notice that you were not required to authenticate because you entered the user name and password in the `soap.client.props` file.

- ___ 5. Upgrade the script to add some error handling. A second script (version 2) is already created, which checks to see whether the application exists.
- ___ a. Open /usr/software/wsadmin/install-IVT-v2.py in an editor.

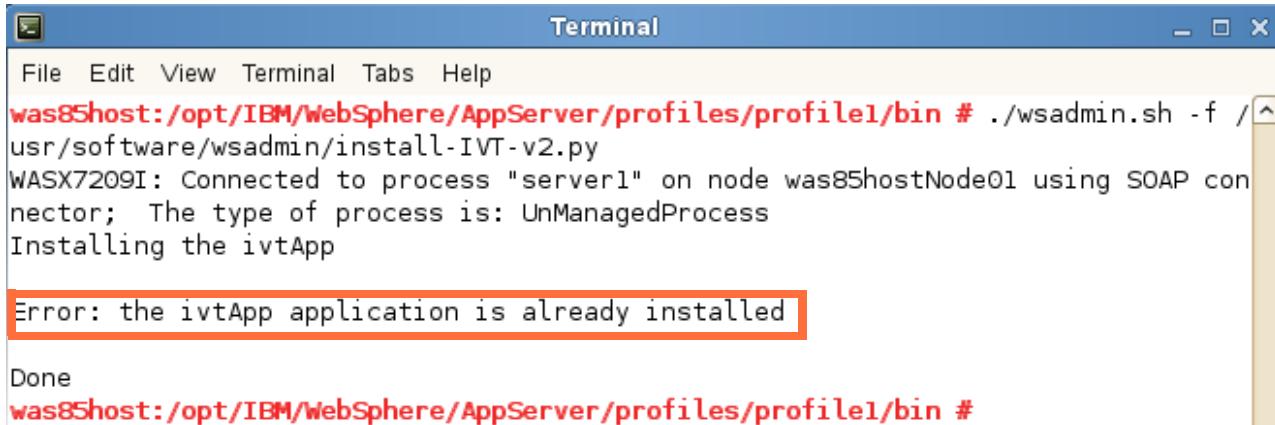
```
#++++
#
# install-IVT-v2.py:
#   - script to install the ivtApp
#   - adds some error handling to see if app is already installed
#
#----
#
print "Installing the ivtApp"

if "IVT Application" not in AdminApp.list().splitlines() :
    AdminApp.install( "/opt/IBM/WebSphere/AppServer/installableApps/ivtApp.ear" )
    AdminConfig.save()
else :
    print
    print "Error: the ivtApp application is already installed"
    print

print "Done"
```

- ___ b. Notice the additional logic, which uses the `AdminApp.list()` function to see whether the ivtApp is already installed.
- ___ c. From the command line, use the following command to run the `install-IVT-v2.py` script:

```
./wsadmin.sh -f /usr/software/wsadmin/install-IVT-v2.py
```



```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./wsadmin.sh -f /usr/software/wsadmin/install-IVT-v2.py
WASX7209I: Connected to process "server1" on node was85hostNode01 using SOAP connector; The type of process is: UnManagedProcess
Installing the ivtApp

Error: the ivtApp application is already installed

Done
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #
```

- ___ d. Notice that the script is successful in catching the error.

- ___ 6. Upgrade the script even further to complete an upgrade if the ivtApp is already installed.
- ___ a. Open /usr/software/wsadmin/install-IVT-v3.py in an editor.

```
#+++++
#
# install-IVT-v3.py:
#   - script to install the ivtApp
#   - adds some error handling to see if app is already installed
#   - if the app is already installed, do an update instead of an install
#
#-----
#
print "Installing the ivtApp"

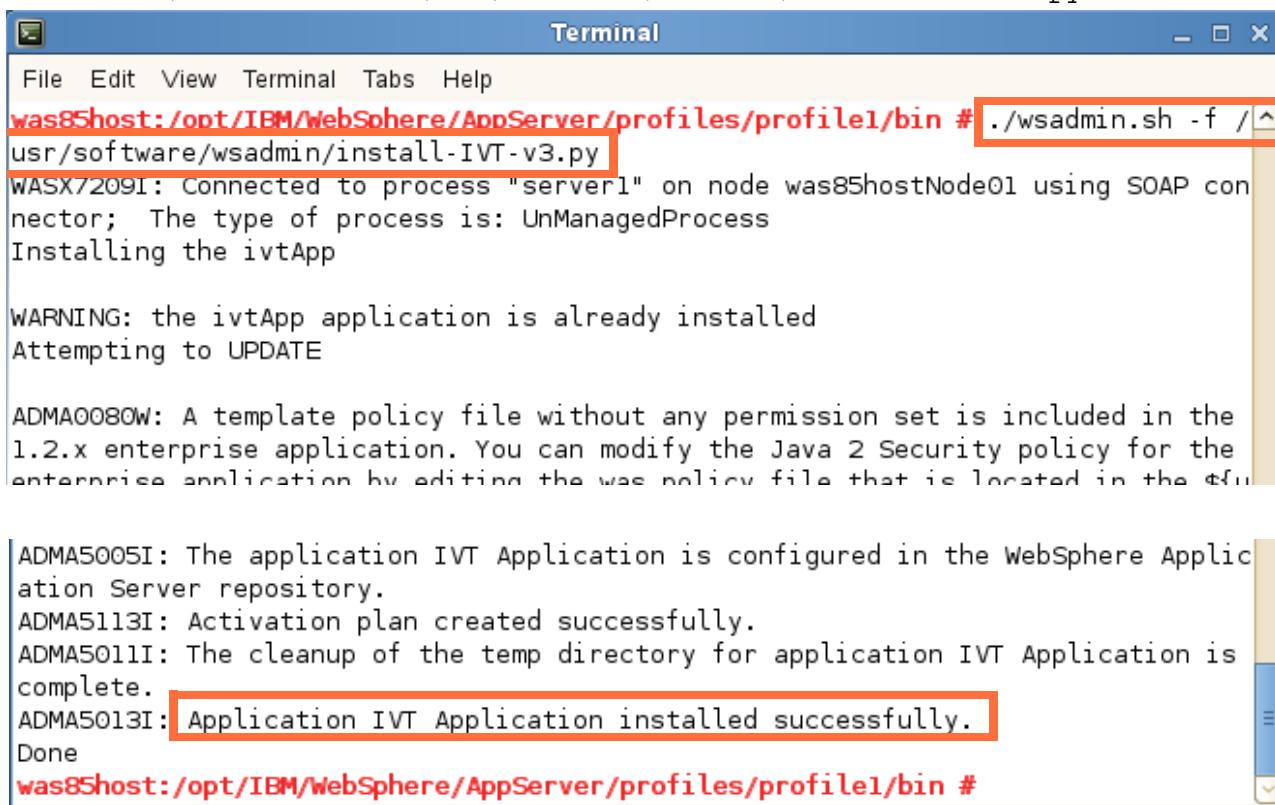
if "IVT Application" not in AdminApp.list().splitlines() :
    AdminApp.install( "/opt/IBM/WebSphere/AppServer/installableApps/ivtApp.ear" )
    AdminConfig.save()
else :
    print
    print "WARNING: the ivtApp application is already installed"
    print "Attempting to UPDATE"
    print
    AdminApp.update( "IVT Application", "app", "[ -operation update -contents /opt/IBM/WebSphere/AppServer/installableApps/ivtApp.ear ]" )
    AdminConfig.save()

print "Done"
```

- ___ b. Notice the new steps in the `else` block. Instead of printing an error message, a warning message is provided and the `AdminApp.update` function is used to update the application.

- ___ c. From the command line, use the following command to run the script install-IVT-v3.py:

```
./wsadmin.sh -f /usr/software/wsadmin/install-IVT-v3.py
```



```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./wsadmin.sh -f /usr/software/wsadmin/install-IVT-v3.py
WASX209I: Connected to process "server1" on node was85hostNode01 using SOAP connector; The type of process is: UnManagedProcess
Installing the ivtApp

WARNING: the ivtApp application is already installed
Attempting to UPDATE

ADMA0080W: A template policy file without any permission set is included in the 1.2.x enterprise application. You can modify the Java 2 Security policy for the enterprise application by editing the was.policy file that is located in the <fu>
```

ADMA5005I: The application IVT Application is configured in the WebSphere Application Server repository.

ADMA5113I: Activation plan created successfully.

ADMA5011I: The cleanup of the temp directory for application IVT Application is complete.

ADMA5013I: Application IVT Application installed successfully.

Done

was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin #

- ___ d. Notice that the script successfully updates the ivtApp.
- ___ 7. Examine the next version of the script, which supports entering arguments from the command line.
- ___ a. Open /usr/software/wsadmin/install-IVT-v4.py in an editor.
- ___ b. Notice that the code now supports defining the application as a command-line argument.



Information

Going through this process iteratively can help create automated scripts for your administrative needs. However, figuring out the specific commands and syntax for the necessary tasks can be a challenge. Helpful resources include the following examples:

- Console command assistance
- Script library examples
- AdminTask functions
- Information center
- Searching the web for examples

Section 6: Using console command assist

Sometimes it is challenging to figure out what wsadmin commands are necessary to use. One useful tool is the console itself. The administrative console, in some cases, can show the command that it uses to effect certain configuration changes. You can then copy these commands and use them in your scripts.



Information

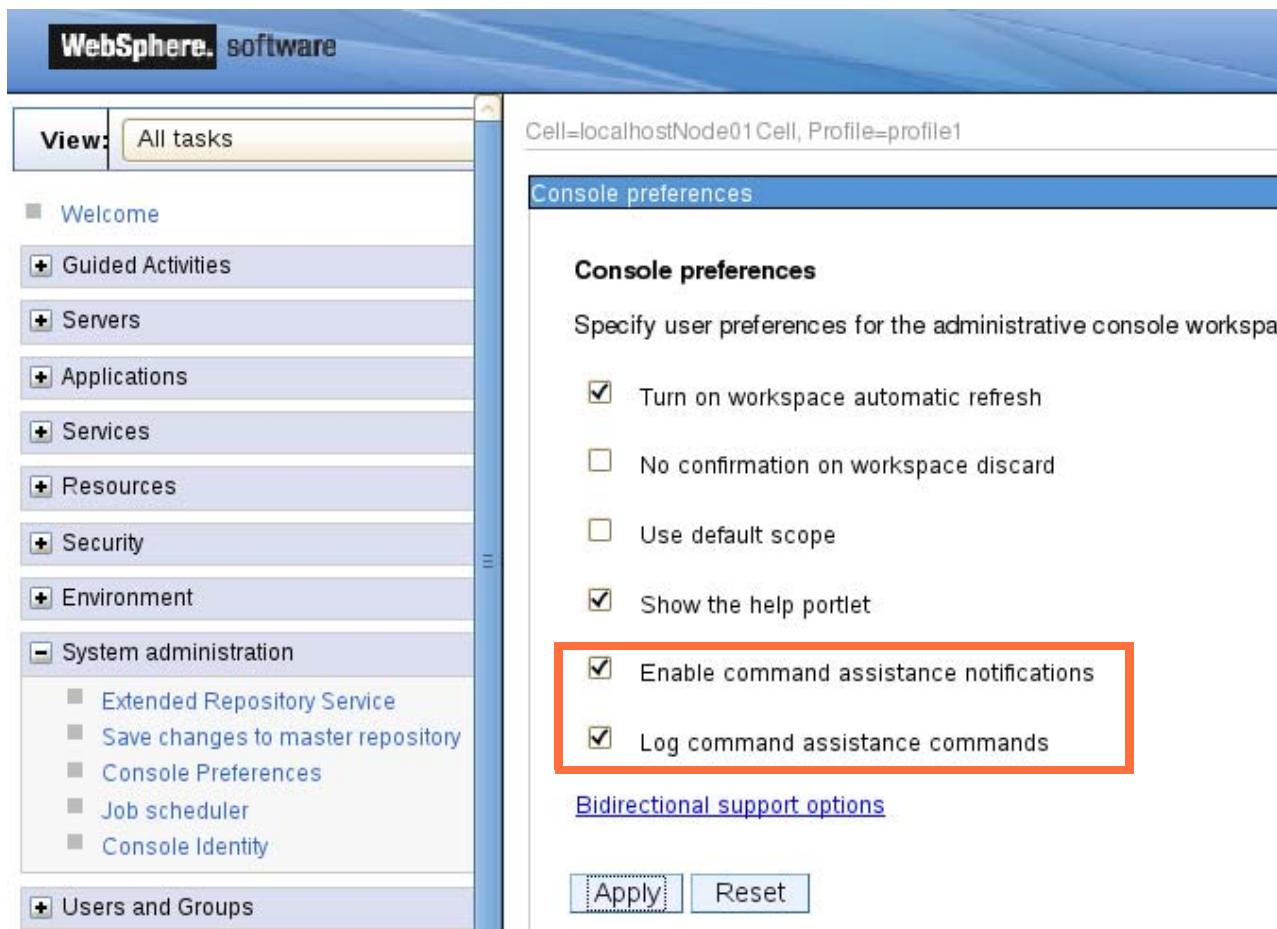
Not all configuration commands are shown in the console. With each new release of the WebSphere Application Server, the number of administrative console actions that show these commands increases.

To further simplify this process, if the IBM Assembly and Deploy Tools is running, it can communicate directly with the administrative console, receive the wsadmin commands, and make them available for inserting into your scripts.

In this exercise, you set up the administrative console and IBM Assembly and Deploy Tools to communicate with each other.

- 1. Enable command assistance in the administrative console.
 - a. Verify that server1 is running
 - b. Using the administrative console, click **System administration > Console Preferences**.

- __ c. Select both **Enable command assistance notifications** and **Log command assistance commands**.



- __ d. Click **Apply**.
__ 2. View a simple command assist output.
__ a. Click **Applications > Application Types > WebSphere enterprise applications**.

- ___ b. On the far right, under **Help**, click **View administrative scripting command for last action**.

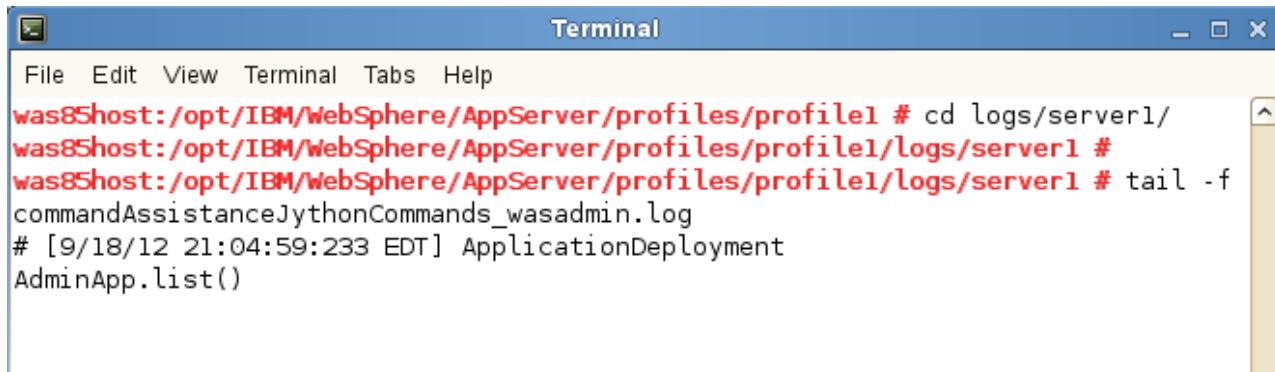
The screenshot shows the IBM WebSphere Administrative Console interface. At the top, it says "Welcome wasadmin". On the right, there are links for "Help" and "Logout". Below the header, there's a "Close page" button. The main content area has tabs for "Remove File", "Export", "Export DDL", and "Export File". A "Navigation Status" section shows a "Status" icon. To the right, a "Help" panel is open. It contains sections for "Field help", "Page help" (with a link to "More information about this page"), and "Command Assistance". In the "Command Assistance" section, the link "View administrative scripting command for last action" is highlighted with a red box.

- ___ c. A new browser window is displayed. Notice that the last operation that you clicked in the administrative console has the `wsadmin` command listed.

The screenshot shows a Mozilla Firefox browser window titled "Administrative Scripting Commands - Mozilla Firefox". The address bar shows "was85host https://was85host:9043/ibm/console/com.ibm.ws.console.core.commandassistance.forward". The main content area is titled "Administrative Scripting Commands". It lists "wsadmin scripting commands that map to actions on the administrative console display in the J" and includes a "Preferences" link. Below this, it shows "Administrative Scripting Command" and a list item "AdminApp.list()". The "AdminApp.list()" item is highlighted with a red box. At the bottom, it says "Total 1".

- ___ 3. View the command assist log. In a previous step, you configured command assistance, which writes commands to a log file.
- ___ a. Using a terminal window, navigate to the server1 log directory. This directory is at /opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1.
- ___ b. Use the tail command to monitor the updates to the command assist log file. Enter the following command:

```
tail -f commandAssitanceJythonCommands_wasadmin.log
```



The screenshot shows a terminal window titled "Terminal". The menu bar includes "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal window displays the following command and its output:

```
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1 # cd logs/server1/
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1 #
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/server1 # tail -f
commandAssistanceJythonCommands_wasadmin.log
# [9/18/12 21:04:59:233 EDT] ApplicationDeployment
AdminApp.list()
```

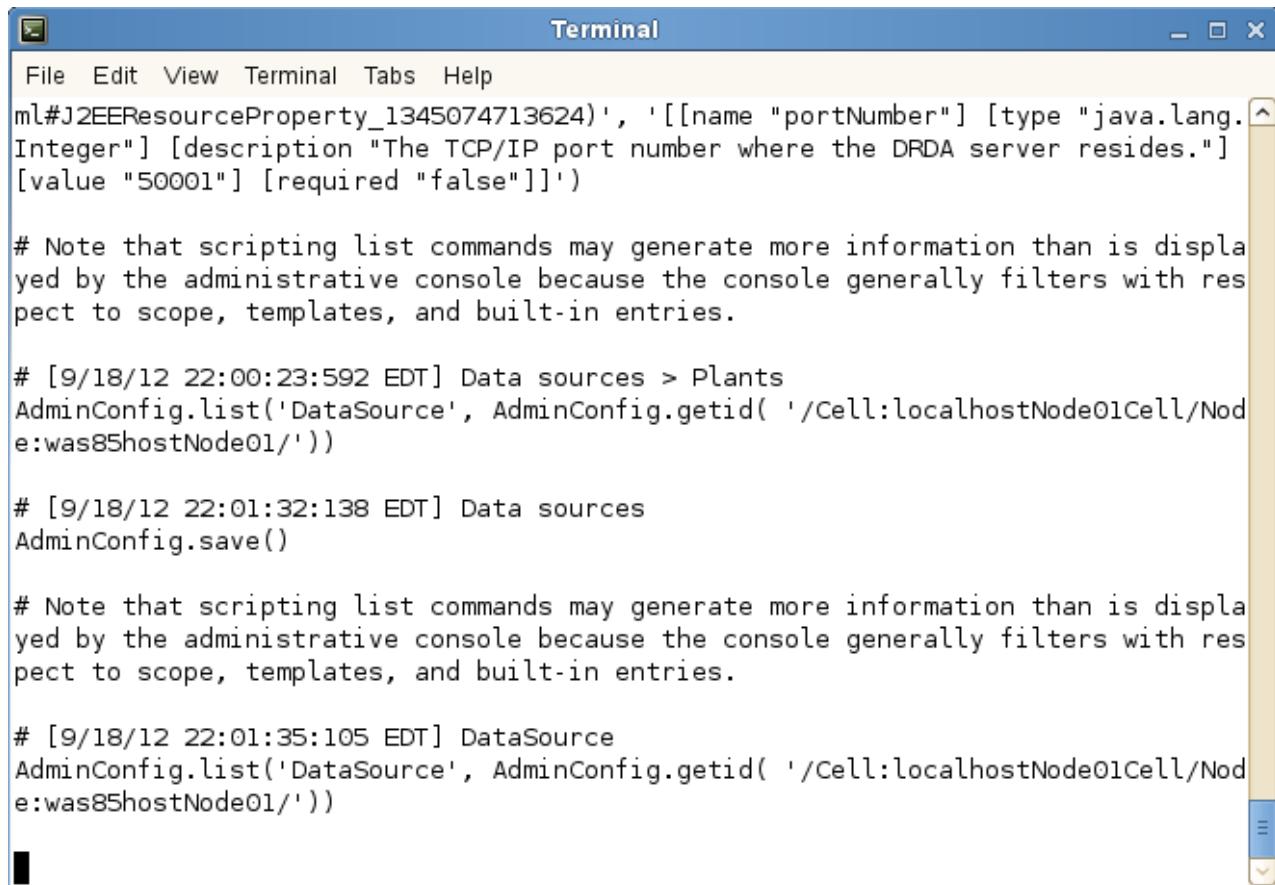


Information

The tail -f command initially displays the last 10 lines of a file. The -f option causes the tail utility to continue displaying new entries in the log file until the user stops the tail (usually with a Ctrl-C).

- ___ c. Leave the command window active, and return to the administrative console. While entering the next few commands in the administrative console, make sure to watch the updates in the command window.
- ___ d. In the administrative console:
- Navigate to **Resources > JDBC > JDBC providers**
 - Click **DB2 Universal JDBC Driver Provider (XA)**
 - From the navigation menu, click **Data sources**
 - Click **Plants**
 - Change the description of the data source to **Used by Plants application**
 - Click **OK**
 - **Save** the changes

- __ e. Return to the command window and notice all of the updates that are made in tail.



```

File Edit View Terminal Tabs Help
ml#J2EEResourceProperty_1345074713624)', '[[name "portNumber"] [type "java.lang.Integer"] [description "The TCP/IP port number where the DRDA server resides."] [value "50001"] [required "false"]]')
# Note that scripting list commands may generate more information than is displayed by the administrative console because the console generally filters with respect to scope, templates, and built-in entries.

# [9/18/12 22:00:23:592 EDT] Data sources > Plants
AdminConfig.list('DataSource', AdminConfig.getid( '/Cell:localhostNode01Cell/Node:was85hostNode01/'))

# [9/18/12 22:01:32:138 EDT] Data sources
AdminConfig.save()

# Note that scripting list commands may generate more information than is displayed by the administrative console because the console generally filters with respect to scope, templates, and built-in entries.

# [9/18/12 22:01:35:105 EDT] DataSource
AdminConfig.list('DataSource', AdminConfig.getid( '/Cell:localhostNode01Cell/Node:was85hostNode01/'))

```

- __ f. Enter **Ctrl-C** in the tail utility to stop it.



Information

Although the command assistance is not always complete, it provides the administrator a good starting point when attempting to create a wsadmin script.

The log file records all the action in the console for later viewing. The information can also be made available to various tools such as IBM Assembly and Deploy Tools.

Section 7: Exploring the scripting libraries

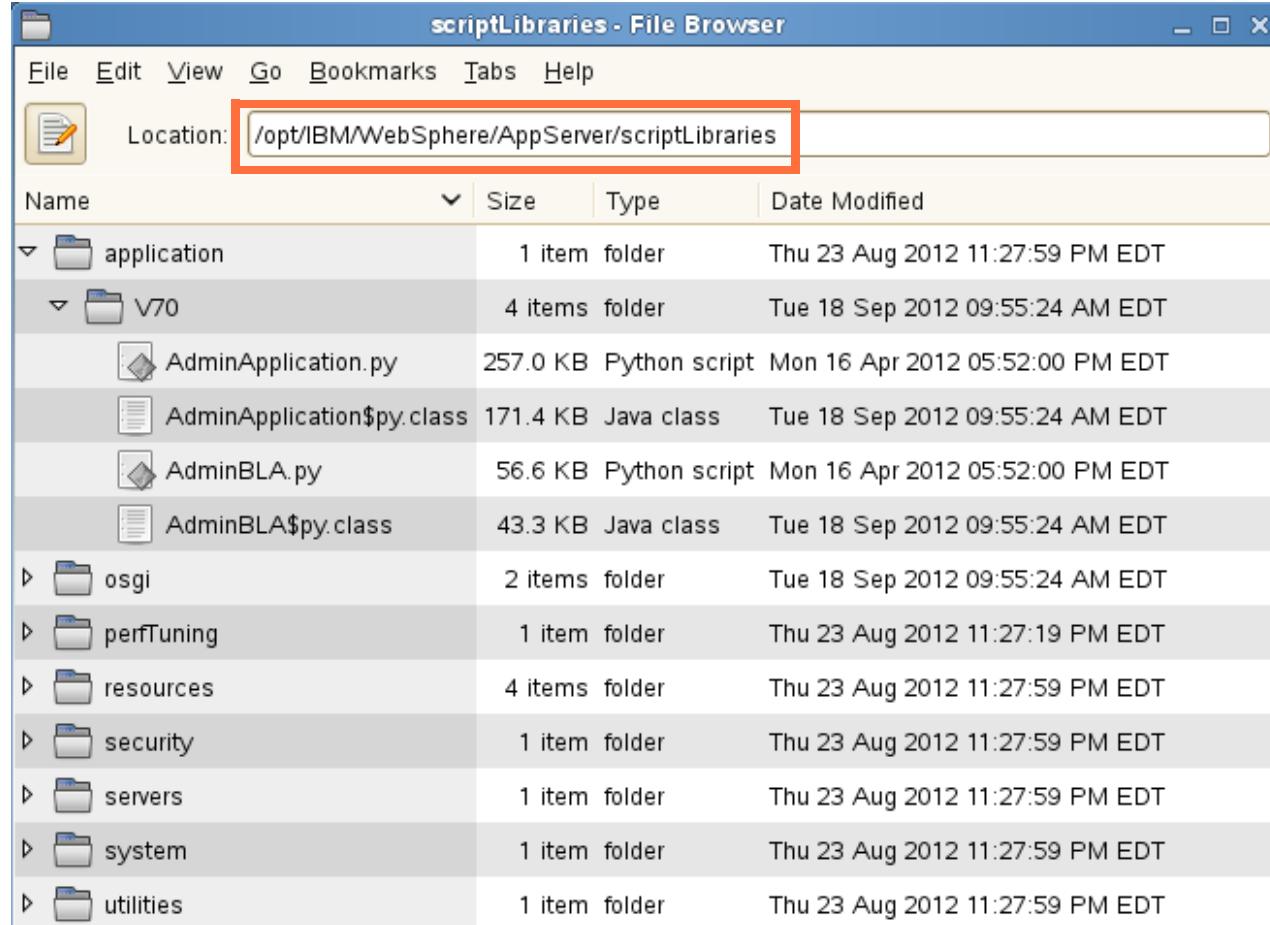
Many scripting operations are simple and straightforward. However, other operations might require a bit of investigation, reading, and trial and error, making it difficult for some users to take full advantage of scripting in their environments. To help users overcome the complexity of scripting, a new set of script libraries was introduced in WebSphere Application Server V7.

In this part of the lab, you examine the available libraries and explore what they provide.

— 1. Examine the libraries.

— a. Open File Browser and navigate to the following folder:

<was_root>/scriptLibraries



— b. Open each of the subfolders and discover the names of the libraries:

Table 5: Scripting libraries

Directory location	Script library
application\V70	AdminApplication AdminBLA
resources\J2C\V70	AdminJ2C
resources\JDBC\V70	AdminJDBC
resources\JMS\V70	AdminJMS
resources\Provider\V70	AdminResources
security\V70	AdminAuthorizations
servers\V70	AdminClusterManagement AdminServerManagement

Table 5: Scripting libraries

Directory location	Script library
system\V70	AdminNodeGroupManagement AdminNodeManagement
utilities\V70	AdminLibHelp AdminUtilities

**Note**

All of these libraries are loaded when wsadmin starts and are readily available from the wsadmin command prompt, or to be used from your own scripts. Even though source code is provided, it is not meant for the user to modify this code. Users of the libraries call code in the libraries from their own scripts. You can copy parts of the library code to other files and modify the copied code to improve it or better suit your needs.

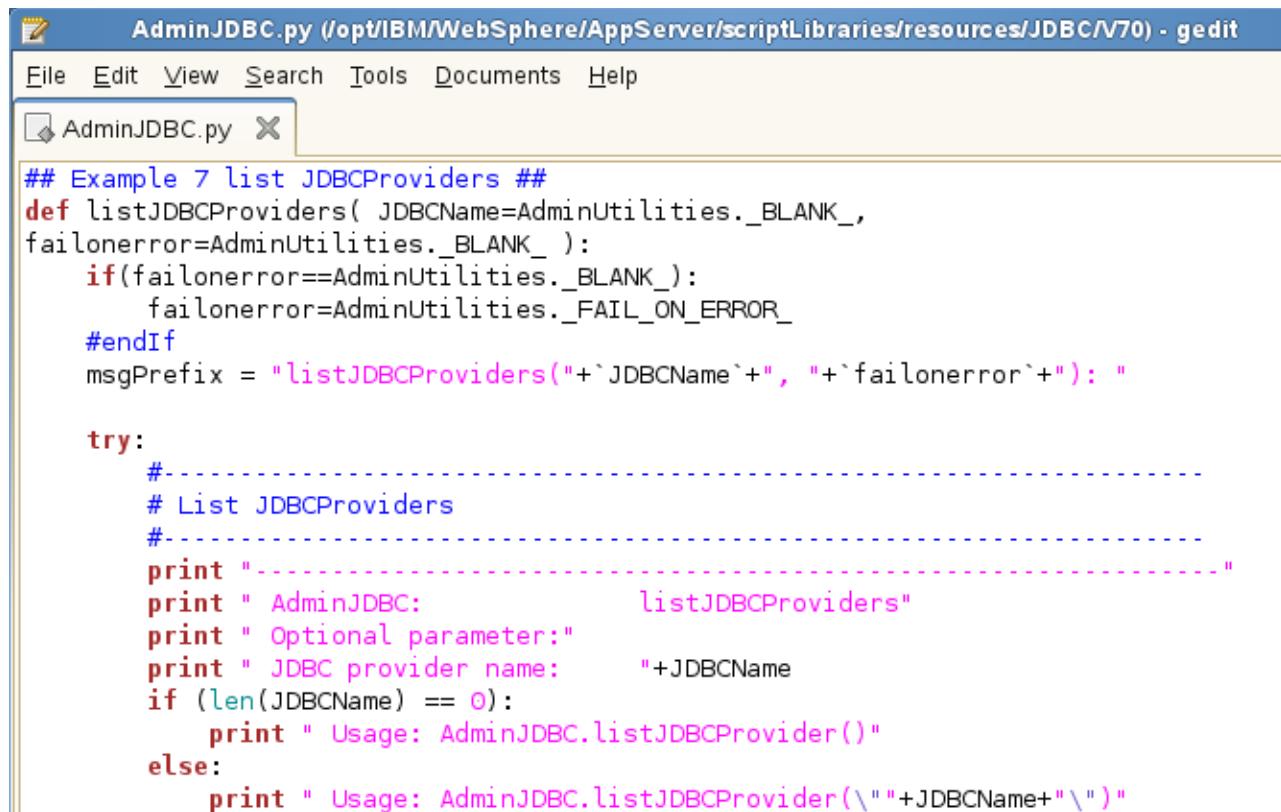
- c. You can open the libraries in a text editor and look at the code. This code is documented, and exceptions and other errors are handled smoothly by providing meaningful error messages to the calling scripts. Use a text editor to open the AdminJDBC script library:

```
<was_root>/scriptLibraries/resources/JDBC/V70/AdminJDBC.py
```

In each library, after the usual copyright and disclaimer statement, there is a list of procedure examples. The AdminJDBC library has 13 example functions:

- **Ex1:** createJDBCProvider
Create a JDBC provider in your environment.
The script returns the configuration ID of the new JDBC provider.
- **Ex2:** createJDBCProviderUsingTemplate
Use a template to create a JDBC provider.
- **Ex3:** listJDBCProviderTemplates
Show a list of configuration IDs for the JDBC provider templates.
- **Ex4:** createDataSource
Create a data source in your configuration.
The script returns the configuration ID of the new data source.
- **Ex5:** createDataSourceUsingTemplate
Use a template to create a data source in your configuration.
The script returns the configuration ID of the new data source.
- **Ex6:** listDataSourceTemplates
Show a list of configuration IDs for the data source templates.
- **Ex7:** listJDBCProviders
Show a list of configuration IDs for the JDBC providers.
- **Ex8:** listDataSources
Show a list of configuration IDs for the data sources.
- **Ex9:** help
Show AdminJDBC script library online help.
- **Ex10:** createJDBCProviderAtScope
Create a JDBC provider at scope.
- **Ex11:** createJDBCProviderUsingTemplateAtScope
Use the template at scope to create a JDBC provider.
- **Ex12:** createDataSourceAtScope
Create a data source at scope.
- **Ex13:** createDataSourceUsingTemplateAtScope
Use the template at scope to create a data source.

Go to **Example 7: listJDBCProviders**. These functions show operations at a higher level than the administrative objects provide. In addition, these scripts provide a better abstraction to the script writer.



```

AdminJDBC.py (/opt/IBM/WebSphere/AppServer/scriptLibraries/resources/JDBC/V70) - gedit
File Edit View Search Tools Documents Help
AdminJDBC.py X

## Example 7 list JDBCProviders ##
def listJDBCProviders( JDBCName=AdminUtilities._BLANK_,
failonerror=AdminUtilities._BLANK_ ):
    if(failonerror==AdminUtilities._BLANK_):
        failonerror=AdminUtilities._FAIL_ON_ERROR_
    #endif
    msgPrefix = "listJDBCProviders(" +`JDBCName` +", " +`failonerror` +")"

    try:
        #-----
        # List JDBCProviders
        #-----
        print "-----"
        print " AdminJDBC:           listJDBCProviders"
        print " Optional parameter:"
        print " JDBC provider name: " +JDBCName
        if (len(JDBCName) == 0):
            print " Usage: AdminJDBC.listJDBCProvider()"
        else:
            print " Usage: AdminJDBC.listJDBCProvider(\"" +JDBCName+ "\")"

```

Examine the example functions that the library provides. Close the file when you are done examining it.

- __ d. A similar listing of functions available in a library can be obtained with the `help()` method. At the wsadmin prompt, enter:

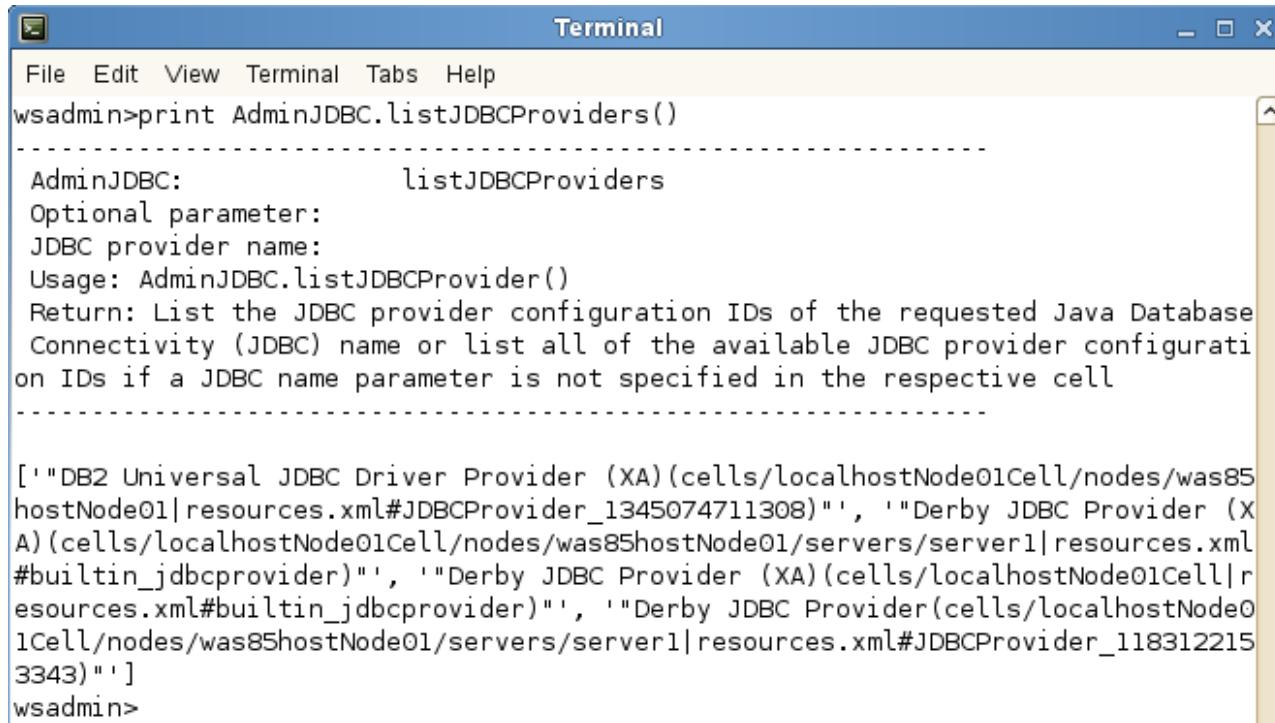
```
print AdminJDBC.help()
```

- __ e. As with the administrative objects, you can get help on a specific method. Enter:

```
print AdminJDBC.help("listJDBCProviders")
```

- ___ f. Not only can you use these commands in your own scripts, but you can run the commands directly. Enter:

```
print AdminJDBC.listJDBCPproviders()
```



The screenshot shows a terminal window titled "Terminal". The menu bar includes "File", "Edit", "View", "Terminal", "Tabs", and "Help". The command "wsadmin>print AdminJDBC.listJDBCPproviders()" is entered, followed by a list of JDBC providers and their configurations.

```

File Edit View Terminal Tabs Help
wsadmin>print AdminJDBC.listJDBCPproviders()
-----
AdminJDBC:           listJDBCPproviders
Optional parameter:
JDBC provider name:
Usage: AdminJDBC.listJDBCPprovider()
Return: List the JDBC provider configuration IDs of the requested Java Database Connectivity (JDBC) name or list all of the available JDBC provider configuration IDs if a JDBC name parameter is not specified in the respective cell
-----
[{"DB2 Universal JDBC Driver Provider (XA)": "cells/localhostNode01Cell/nodes/was85hostNode01/resources.xml#JDBCProvider_1345074711308"}, {"Derby JDBC Provider (XA)": "cells/localhostNode01Cell/nodes/was85hostNode01/servers/server1/resources.xml#builtin_jdbcprovider"}, {"Derby JDBC Provider (XA)": "cells/localhostNode01Cell/resources.xml#builtin_jdbcprovider"}, {"Derby JDBC Provider(cells/localhostNode01Cell/nodes/was85hostNode01/servers/server1/resources.xml#JDBCProvider_1183122153343)"}]
wsadmin>

```

As with any new library system, it takes a while to become familiar and comfortable with the functions available. By combining these library functions with your own scripting logic in your Jython scripts, you can write scripts to configure your application servers.

Section 8: Using properties file-based configuration

WebSphere Application Server provides a set of utilities for working with server configurations by using properties files. You can create a properties file of human readable key-value pairs that are based on your environment. You can modify the properties files and apply the updated configurations to a server. The objective of this portion of the lab is to provide you with a basic understanding of this technique for administering your environment by using property files.

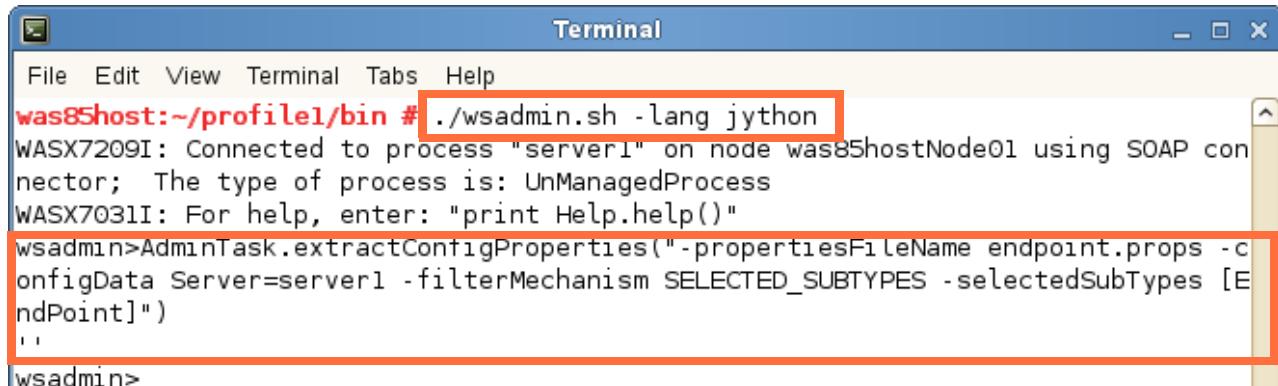
You extract the EndPoint resource for server1 that contains the list of port name-value pairs.

- ___ 1. Using wsadmin, extract the properties for the server1.
 - ___ a. In a terminal window, navigate to the bin directory for profile1.
 - ___ b. Use the following command to start wsadmin (if prompted to authenticate, use wasadmin and websphere):


```
./wsadmin.sh -lang jython
```

- ___ c. Enter the following command to extract the resource type EndPoint for server1 into a properties file called endpoint.props (you can copy and paste from the file /usr/software/wsadmin/properties.py):

```
AdminTask.extractConfigProperties( "-propertiesFileName
endpoint.props -configData Server=server1 -filterMechanism
SELECTED_SUBTYPES -selectedSubTypes [EndPoint]" )
```



The screenshot shows a terminal window titled "Terminal". The command `./wsadmin.sh -lang jython` is entered and executed. The output shows the connection to "server1" and the execution of the AdminTask command to extract configuration properties.

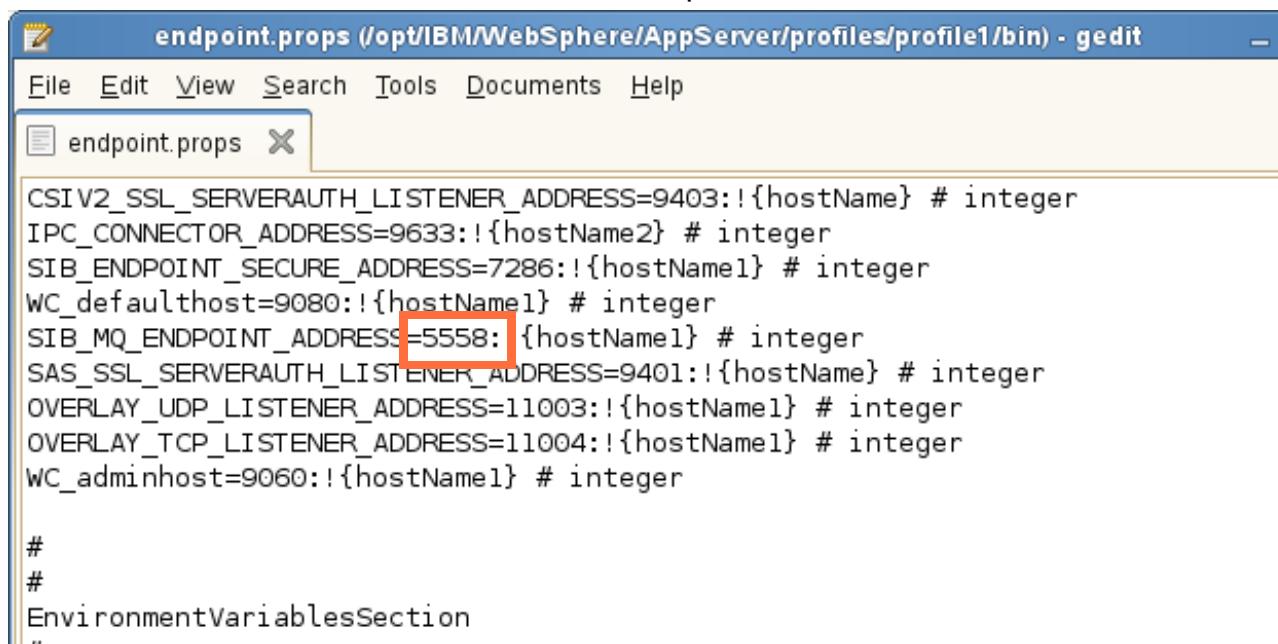
```
File Edit View Terminal Tabs Help
was85host:~/profile1/bin # ./wsadmin.sh -lang jython
WASX7209I: Connected to process "server1" on node was85hostNode01 using SOAP connector; The type of process is: UnManagedProcess
WASX7031I: For help, enter: "print Help.help()"
wsadmin>AdminTask.extractConfigProperties("-propertiesFileName endpoint.props -configData Server=server1 -filterMechanism SELECTED_SUBTYPES -selectedSubTypes [EndPoint]")
"
wsadmin>
```



Information

Properties files can be extracted for various configuration attributes at different levels: a cell, a node, a server, one container in that server, and others. If you know what type of properties you want to modify, you can extract a properties file for your server with an object type filter.

- ___ d. The properties file is in the current directory, in this case, in the `bin` directory for `profile1`. Open the `endpoint.props` file with `gedit`. Look for the entry: `SIB_MQ_ENDPOINT_ADDRESS`. Note the port number: _____



The screenshot shows a gedit window displaying the `endpoint.props` file. The file contains several configuration entries, including the `SIB_MQ_ENDPOINT_ADDRESS` entry which is highlighted with a red box. The value for this entry is `5558`.

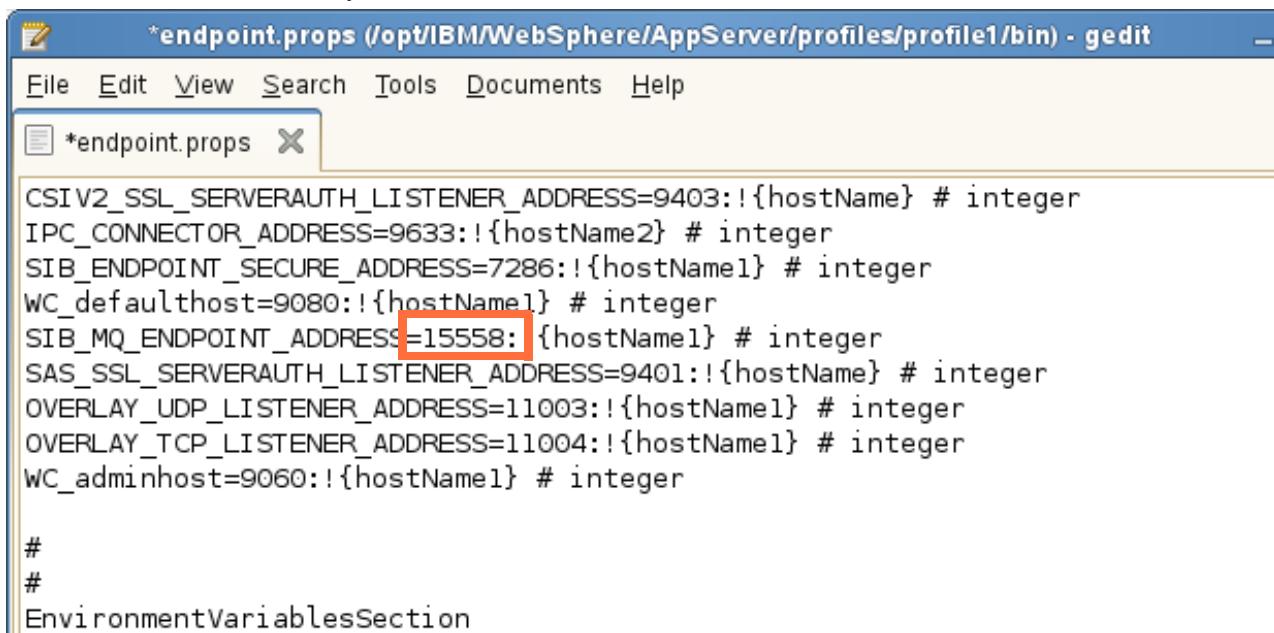
```
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=9403:{hostName} # integer
IPC_CONNECTOR_ADDRESS=9633:{hostName2} # integer
SIB_ENDPOINT_SECURE_ADDRESS=7286:{hostName1} # integer
WC_defaulthost=9080:{hostName1} # integer
SIB_MQ_ENDPOINT_ADDRESS=5558:{hostName1} # integer
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=9401:{hostName} # integer
OVERLAY_UDP_LISTENER_ADDRESS=11003:{hostName1} # integer
OVERLAY_TCP_LISTENER_ADDRESS=11004:{hostName1} # integer
WC_adminhost=9060:{hostName1} # integer

#
#
EnvironmentVariablesSection
```

- __ e. Open the administrative console and click **Servers > Server Types > WebSphere application servers > server1**.
- __ f. Under **Communications**, click **Ports**. Look for the entry **SIB_MQ_ENDPOINT_ADDRESS**. Note the port number: _____

<input type="checkbox"/>	SIB ENDPOINT SECURE ADDRESS	*	7286
<input type="checkbox"/>	SIB MQ ENDPOINT ADDRESS	*	5558
<input type="checkbox"/>	SIB MQ ENDPOINT SECURE ADDRESS	*	5578

- __ g. Notice that the port numbers match because they represent the same configuration item in two different views.
- __ 2. Modify the `endpoint.props` file.
- __ a. Locate the port and value pair for `SIB_MQ_ENDPOINT_ADDRESS`. Change the port value, currently 5558, to: 15558



```

*endpoint.props (/opt/IBM/WebSphere/AppServer/profiles/profile1/bin) - gedit
File Edit View Search Tools Documents Help
*endpoint.props X

CSIv2_SSL_SERVERAUTH_LISTENER_ADDRESS=9403:{hostName} # integer
IPC_CONNECTOR_ADDRESS=9633:{hostName2} # integer
SIB_ENDPOINT_SECURE_ADDRESS=7286:{hostName1} # integer
WC_defaulthost=9080:{hostName1} # integer
SIB_MQ_ENDPOINT_ADDRESS=15558:{hostName1} # integer
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=9401:{hostName} # integer
OVERLAY_UDP_LISTENER_ADDRESS=11003:{hostName1} # integer
OVERLAY_TCP_LISTENER_ADDRESS=11004:{hostName1} # integer
WC_adminhost=9060:{hostName1} # integer

#
#
EnvironmentVariablesSection

```

- __ b. Save the updated file.

- ___ c. Use the following command in wsadmin to validate the updated properties file:

```
AdminTask.validateConfigProperties( "-propertiesFileName
endpoint.props" )
```

```
File Edit View Terminal Tabs Help
wsadmin>AdminTask.validateConfigProperties("-propertiesFileName endpoint.props")
'true'
wsadmin>
```

- ___ d. If the file validation is successful, 'true' is shown.

- ___ 3. Apply the updated properties file to the configuration.

- ___ a. From the wsadmin command line, enter:

```
AdminTask.applyConfigProperties( "-propertiesFileName
endpoint.props" )
```

- ___ b. If the configuration is updated successfully, two quotation marks are shown.

- ___ 4. Save the changes.

- ___ a. From the wsadmin command line, enter:

```
AdminConfig.save()
```

```
File Edit View Terminal Tabs Help
wsadmin>AdminTask.applyConfigProperties("-propertiesFileName endpoint.props")
""
wsadmin>
wsadmin>AdminConfig.save()
""
wsadmin>
```

- ___ 5. Verify the changes.

- ___ a. From the administrative console, return to the **Ports** page (a page refresh might be needed). Notice that the port value for SIB_MQ_ENDPOINT_ADDRESS is now 15558, reflecting the change that you made in the properties file.

<input type="checkbox"/>	<u>SIB ENDPOINT SECURE ADDRESS</u>	*	7286
<input type="checkbox"/>	<u>SIB MQ ENDPOINT ADDRESS</u>	*	15558
<input type="checkbox"/>	<u>SIB MQ ENDPOINT SECURE ADDRESS</u>	*	5578

- ___ 6. Using the administrative console, change the port value for `SIB_MQ_ENDPOINT_ADDRESS` back to the original value (verify that it is 5558) and save the updates.
- ___ a. Click **Details**.
- ___ b. Select `SIB_MQ_ENDPOINT_ADDRESS` in the list of ports.

<input type="checkbox"/>	SIB_MQ_ENDPOINT_ADDRESS	*	15558	View associated transports
--------------------------	---	---	-------	--

- ___ c. Enter **5558** as the **Port**.

[Application servers](#) > [server1](#) > [Ports](#) > `SIB_MQ_ENDPOINT_ADDRESS`

Specifies the TCP/IP ports this server uses for connections.

The screenshot shows the 'Configuration' tab for the 'SIB_MQ_ENDPOINT_ADDRESS' port. Under 'General Properties', the 'Port Name' is set to 'SIB_MQ_ENDPOINT_ADDRESS'. The 'Host' field contains an asterisk (*). The 'Port' field is set to '5558' and is highlighted with a red box. At the bottom are 'Apply', 'OK', 'Reset', and 'Cancel' buttons.

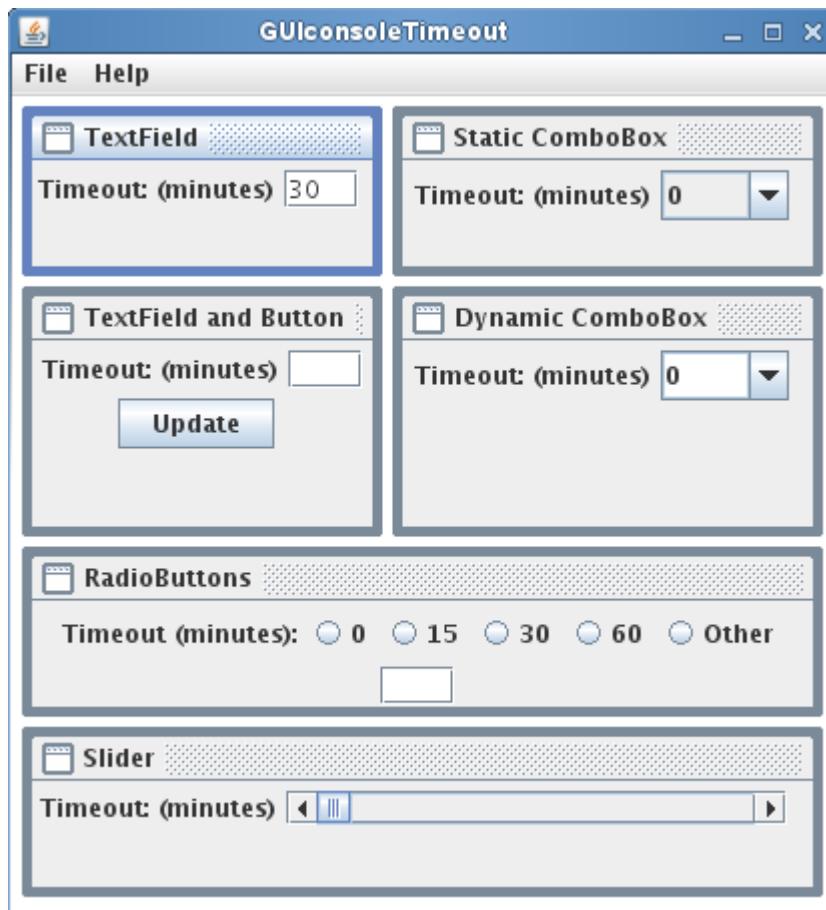
- ___ d. Click **Apply** and **save** the update.

Section 9: Using SWING with Jython (optional)

When creating wsadmin scripts, it might be useful to add a graphical interface. Doing so is not trivial, but it is possible. Look at the following example if you are interested in examining a script that uses the Java SWING libraries from within Jython to create a graphical user interface.

- ___ 1. Using wsadmin, run the `GUIconsoleTimeout.py` script.
- ___ a. From the command line, from the `server1 bin` directory, run the following command (if prompted, use `wasadmin` and `websphere` to authenticate):
- ```
./wsadmin.sh -f /usr/software/wsadmin/GUIconsoleTimeout.py
```
- This script is similar to the `consoleTimeout.py` script, which is used in an earlier exercise, except that it adds a graphic interface. Functionally, it allows you

to set the console timeout value for an application server or deployment manager. But in this case, it completes this function graphically.



#### Note

The interface completes the same function multiple times. Each box uses a different type of graphical widget to set the console timeout value. Feel free to explore the different options.

- \_\_\_ b. If you are interested, open the script file with an editor and investigate how the graphical interface is coded.



## Information

For more information about creating graphical user interfaces for wsadmin scripts, see the following topics:

Step into the Swing era - How to easily add a GUI to your WebSphere Application Server administrative scripts.

[http://www.ibm.com/developerworks/websphere/techjournal/1205\\_col\\_gibson/1205\\_col\\_gibson.html](http://www.ibm.com/developerworks/websphere/techjournal/1205_col_gibson/1205_col_gibson.html)

[http://www.ibm.com/developerworks/websphere/techjournal/1206\\_col\\_gibson/1206\\_col\\_gibson.html](http://www.ibm.com/developerworks/websphere/techjournal/1206_col_gibson/1206_col_gibson.html)

- c. Press Enter in the command window to end the script.

## End of exercise

## Exercise review and wrap-up

In this exercise, you learned to access wsadmin objects from the command line and from a wsadmin Jython shell. You also learned how to create a wsadmin administrative script.

You also learned how to activate command assist and reference the script libraries. Finally, you learned how to use property file based configurations to configure your environment.



# Exercise 9. Creating a federated cell

## What this exercise is about

In this lab exercise, you experience the process of creating a WebSphere cell through the generation of a deployment manager profile and by the federation of application server profiles.

## What you should be able to do

At the end of the lab, you should be able to:

- Create a deployment manager profile
- Back up the deployment manager configuration
- Use the deployment manager administrative console
- Federate a node into the deployment manager cell
- Create a custom profile
- Create an unmanaged web server node
- Use the administrative console to start and stop a web server
- Map an application to a web server

## Introduction

This exercise examines the process of creating and federating a cell. The initial steps include creating two more profiles, the first of which is a deployment manager profile. After the deployment manager profile is created, profile1 is federated into the cell. Then, a custom profile is created and federated at the same time.

This exercise demonstrates the process of creating a cell and prepares the lab environment for other important steps, including creating a node to manage a remote web server and clustering an application server.

## Requirements

To complete this exercise, you must have the application server named server1 started. The DefaultApplication and PlantsByWebSphere applications must be installed and running on server1.

## Exercise instructions

### Preface

To do this exercise, you must complete the Installing IBM Installation Manager and Installing WebSphere Application Servers exercises as the exercises set up the environment that is used in this exercise.



#### Important

The labs use two variables to define various installation paths. On Linux, the variable definitions are as follows:

```
<was_root>: /opt/IBM/WebSphere/AppServer
<profile_root>: /opt/IBM/WebSphere/AppServer/profiles
```

During this exercise, you change your stand-alone application server environment to a cell environment that contains two federated nodes and an unmanaged node for a web server. It is important as you progress through the exercise that you have a good understanding of what you are creating.

As you begin the exercise, you have one stand-alone application server, named `server1`, contained in a node, named `was85hostNode01`.

When you complete the exercise, you have a cell, named `was85hostCell01`, containing the following nodes:

- Deployment manager node, named `was85hostCellManager01`
- A federated node, named `was85hostNode01`, containing a node agent and an application server, named `server1`
- A federated node, named `was85hostNode02`, containing only a node agent
- An unmanaged node, named `ihsnode`, containing an IBM HTTP Server administrative process and a web server, named `webserver1`

### Section 1: Resetting the WebSphere environment



#### Note

To reset your WebSphere environment, read **Appendix A** for instructions on how to complete this procedure.

## Section 2: Use the Profile Management Tool to create a deployment manager profile

During this section of the exercise, you use the Profile Management Tool to create a deployment management profile. The deployment manager profile defines a cell, named was85hostCell01, containing a deployment manager node, named was85hostCellManager01. The existing application server, server1, continues to be a stand-alone server that is contained in the node was85hostNode01.

The Profile Management Tool is part of the WebSphere Customization Toolbox, and is a GUI tool for creating WebSphere profiles. Using the profile wizard, you can create an application server profile, deployment manager profile, custom profile, or cell profile (which creates both a deployment manager and managed node).

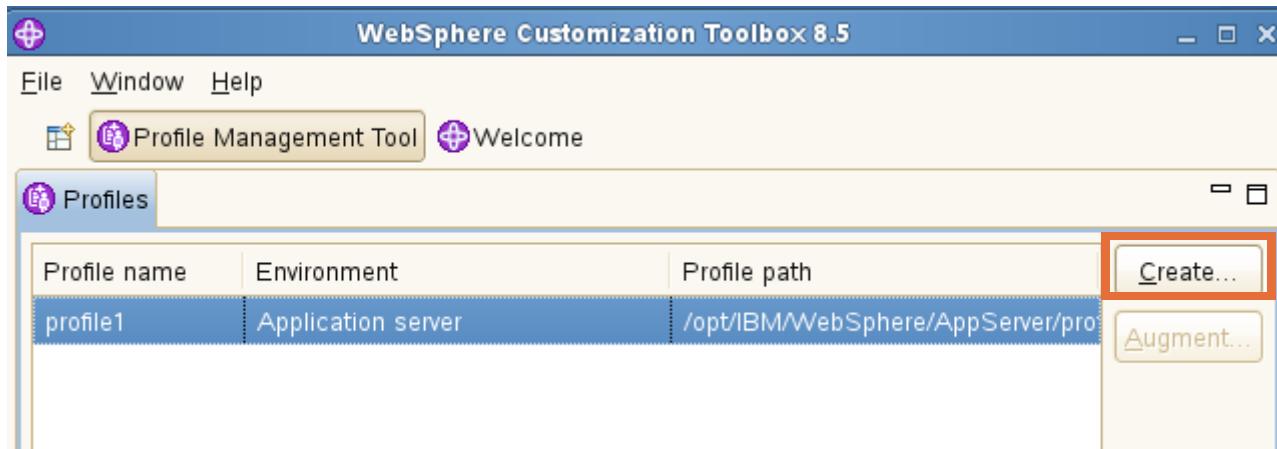


### Information

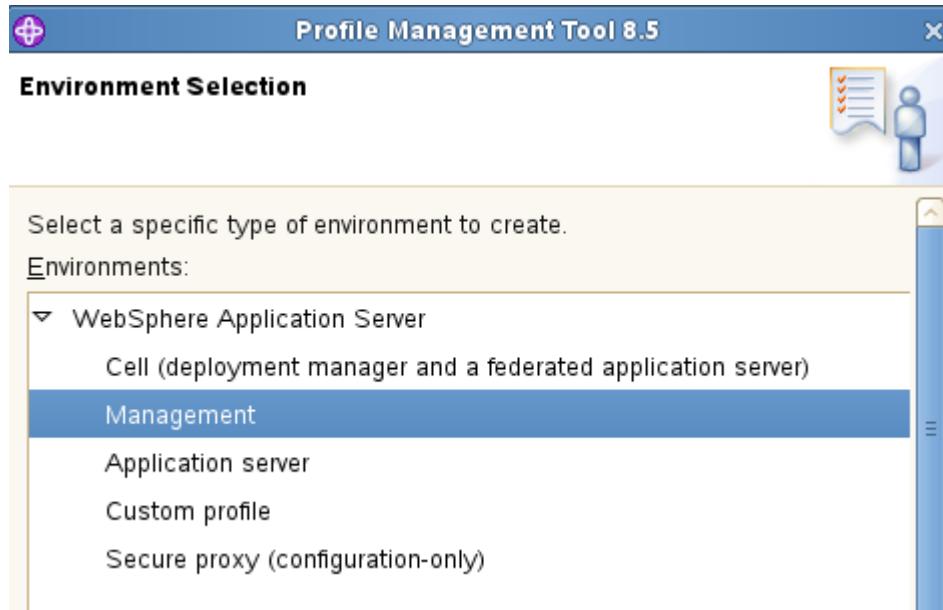
It is possible to create profiles from the command line by using the `manageprofiles -create` script that is in the `/opt/IBM/WebSphere/AppServer/bin` directory.

```
./manageprofiles.sh -create -profileName profile2
-profilePath /opt/IBM/WebSphere/AppServer/profiles/profile2
-templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/default
-nodeName was85hostNode02 -cellName was85hostCell02
-hostName was85host
```

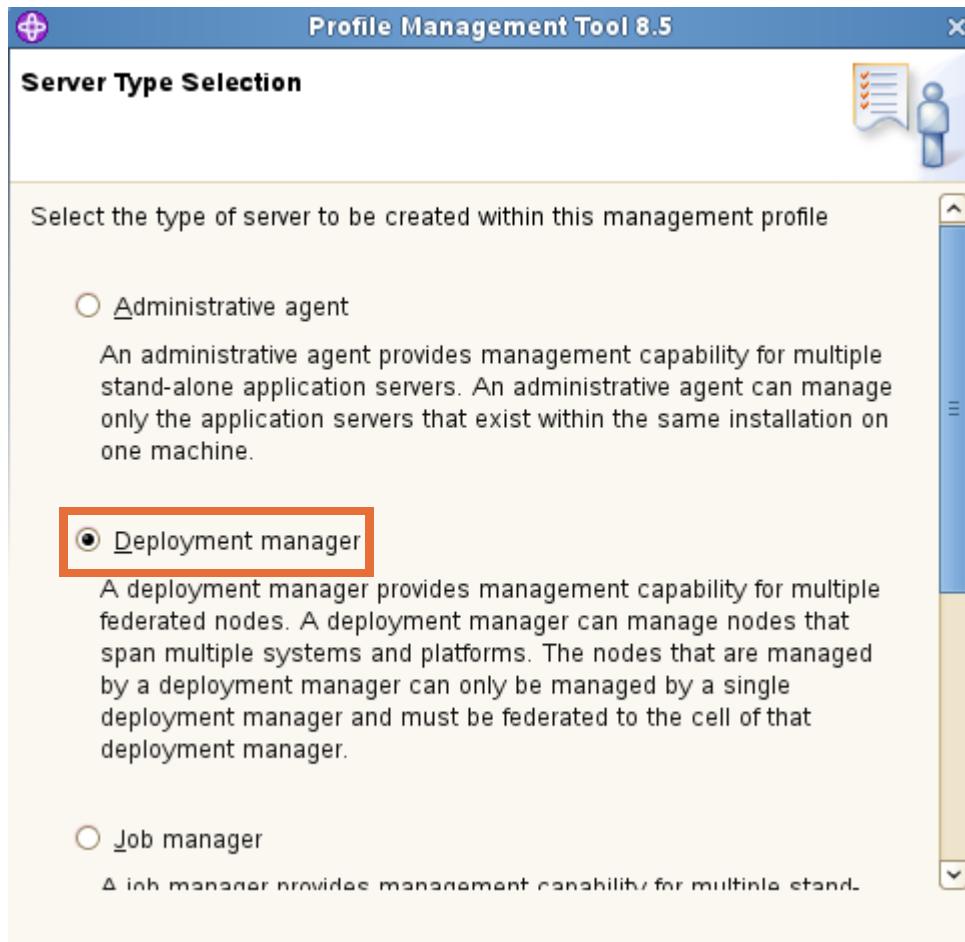
- \_\_\_ 1. Start the WebSphere Customization Toolbox.
  - \_\_\_ a. The command to start the WebSphere Customization Toolbox is:  
`/opt/IBM/WebSphere/AppServer/bin/ProfileManagement/wct.sh`
  - \_\_\_ b. The WebSphere Customization Toolbox opens on the Profile Management Tool tab. Click **Create**.



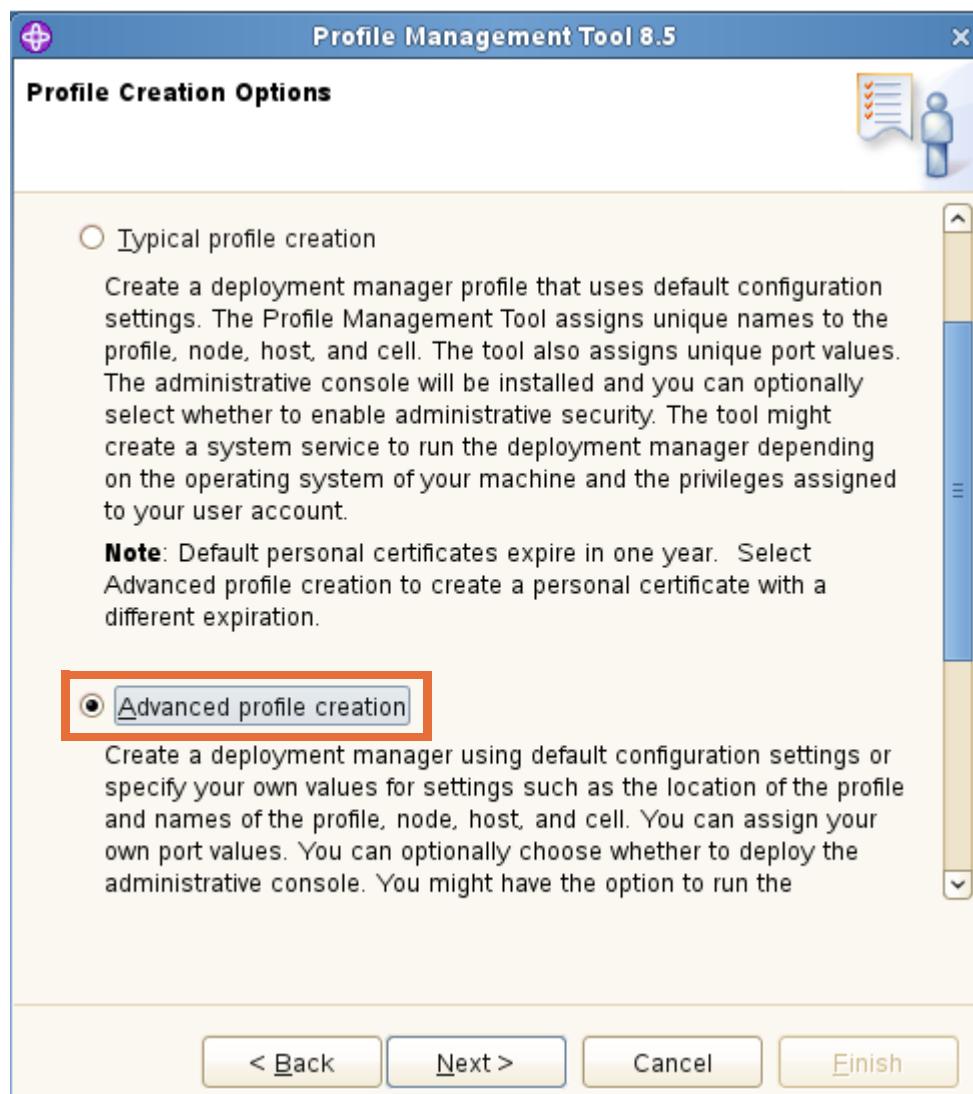
- \_\_ 2. Create a deployment manager profile called Dmgr.  
\_\_ a. From the Environment Selection panel, select **Management** and click **Next**.



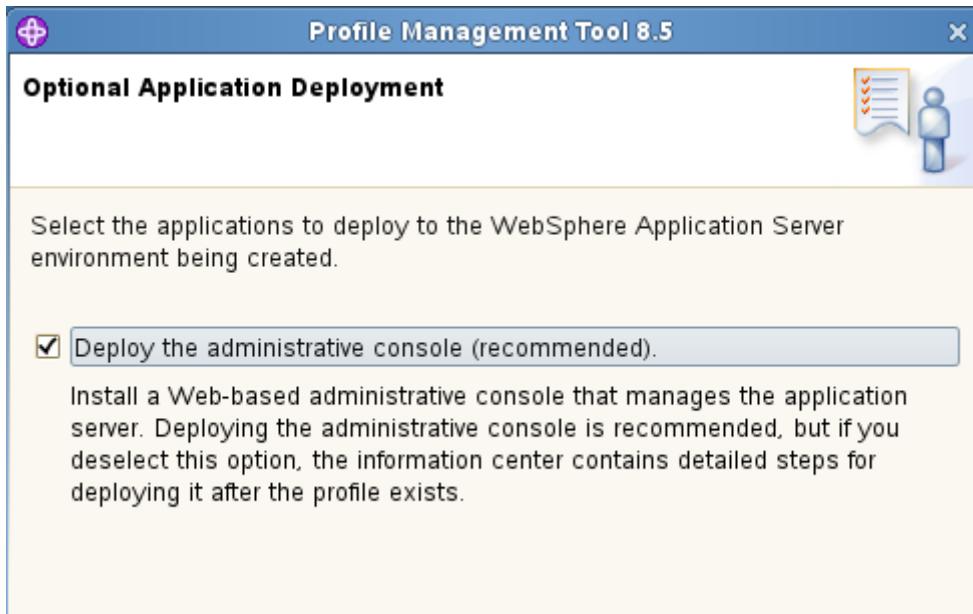
- \_\_ b. From the Server Type Selection panel, select **Deployment manager**.



- \_\_\_ c. Click **Next**.
- \_\_\_ d. Select **Advanced profile creation** to specify your own configuration values during profile creation. Click **Next**.

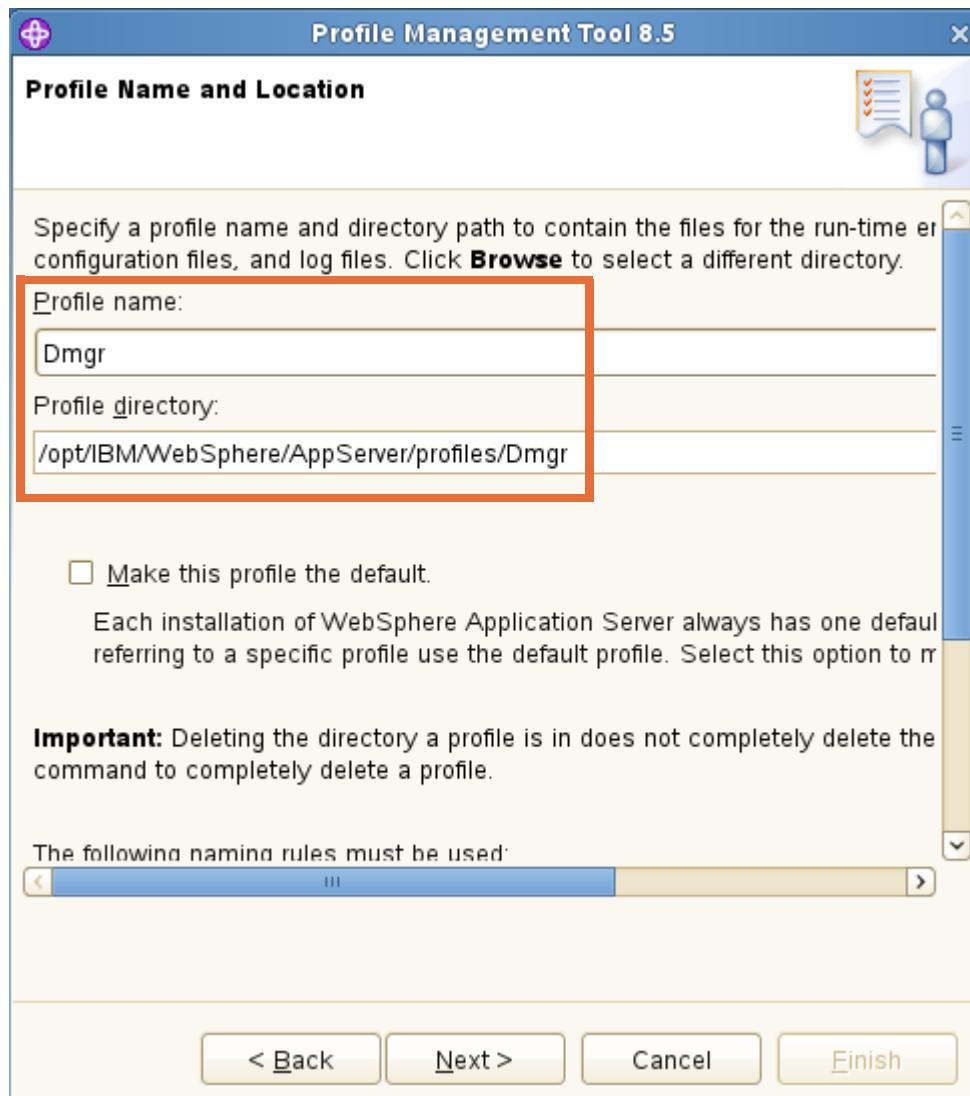


- \_\_ e. Ensure that the **Deploy the administrative console (recommended)** check box is selected. The administrative console is needed for this class. Click **Next**.



- \_\_ f. From the Profile Name and Location panel, provide the following name and location information:
- Profile name: Dmgr
  - Profile directory: /opt/IBM/WebSphere/AppServer/profiles/Dmgr
  - Do **not** select the **Make this profile the default** option.

- g. Click **Next**.



### Information

The default profile is the first profile created. It is also possible to change which profile is designated as the default with the Profile Management Tool or the `manageprofiles` command.

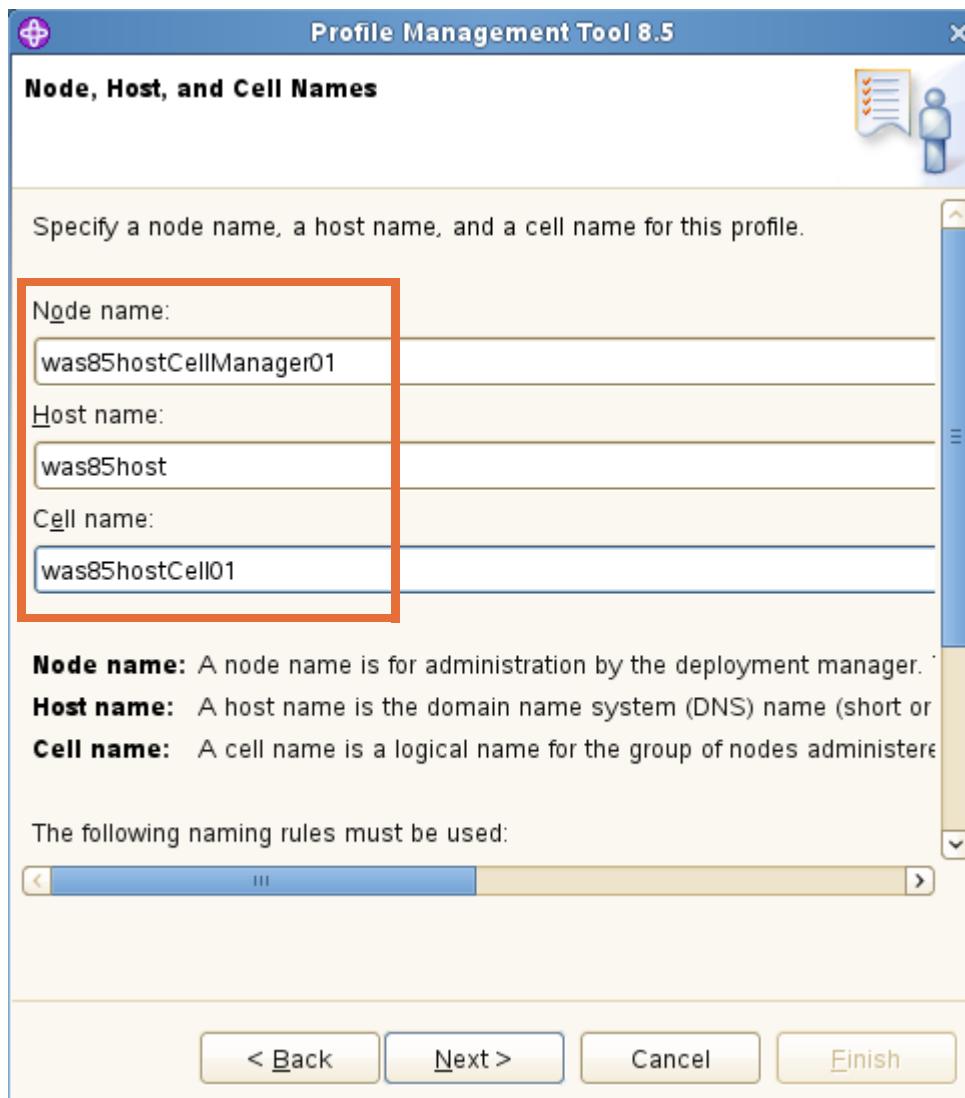
When running commands from the `/opt/IBM/WebSphere/AppServer/bin` directory, commands are run against the runtime that the default profile defines. It is also possible to use the `-profileName` argument to specify a particular profile.

- h. The **Node, Host, and Cell Names** panel allows you to set the node name, cell name, and host name. Default values are completed based on the detected host name for your server.

**Note**

On UNIX systems, the host name can be the long name (`was85host.ibm.com`). Accept whatever the default is. Make sure that you are consistent in later exercises.

Ensure that the **Node name**, **Host name**, and **Cell name** are correct (they are based on the short form of the host name and not localhost: for example, `was85hostCellManager01`, `was85host`, and `was85hostCell01`). If the fields include `localhost` as part of the name, replace `localhost` with the **host name**. For example, change `localhostCellManager01` to: `was85hostCellManager01`



- \_\_ i. Click **Next**.

- \_\_ j. From the Administrative Security panel, you choose whether to enable administrative security. Verify that the **Enable administrative security** option check box is selected. Enter the following information:

- User name: wasadmin
- Password: web1sphere
- Confirm password: web1sphere

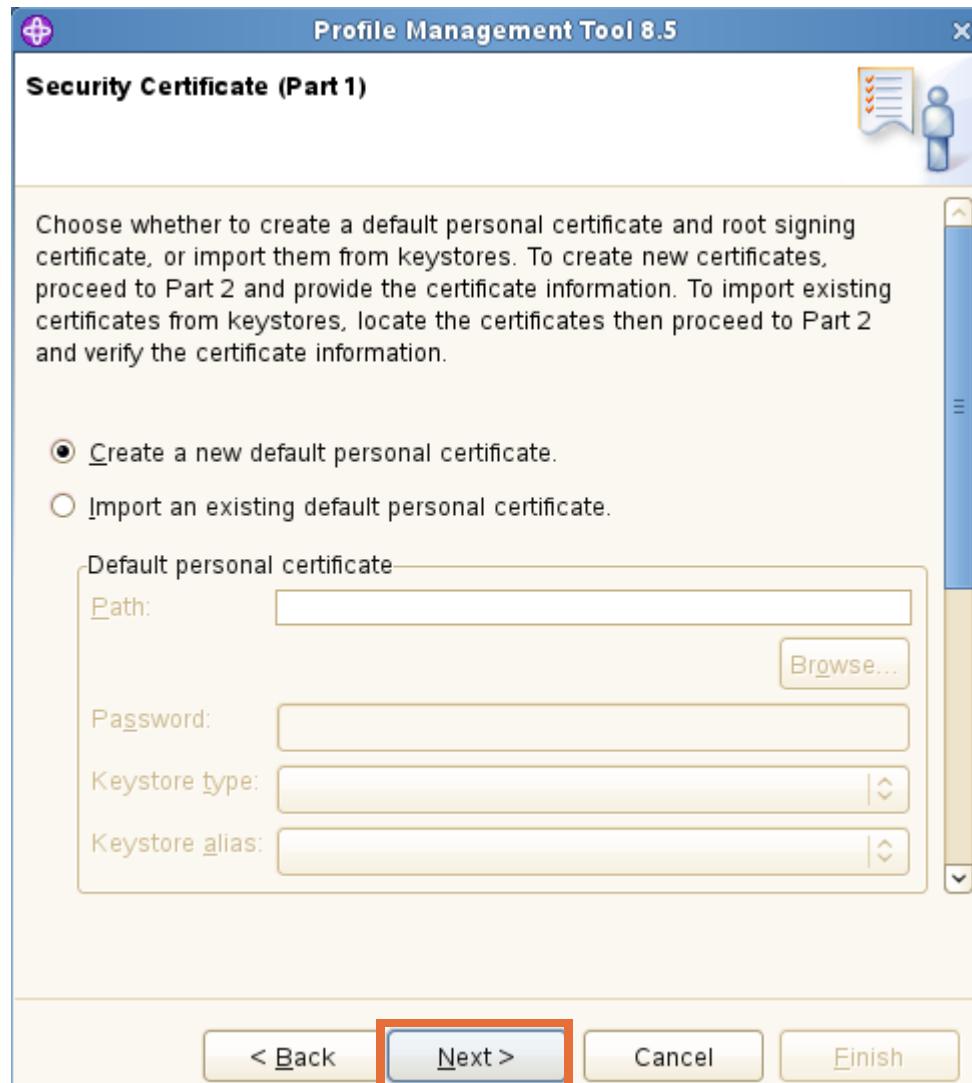
- \_\_ k. Click **Next**.



\_\_ I. From the Security Certificate (Part 1) panel, accept the default selections:

- Create a new default personal certificate
- Create a root signing certificate

Click **Next**.



### Note

The **Issued to distinguished name** and the **Issued by distinguished name** on the Security Certificate (Part 2) panel have a common name (CN) that can take different forms, depending on your environment:

- IP address (such as 192.168.192.128)
- Fully qualified domain name (FQDN) (such as was85host.localdomain or was85host.ibm.com)

- \_\_ m. Accept the Security Certificate (Part 2) panel defaults. Click **Next**.



- \_\_\_ n. The Port Values Assignment panel allows you to set any ports for the deployment manager to prevent conflicts with other profiles. Accept the default port values (which can be different from the example shown).

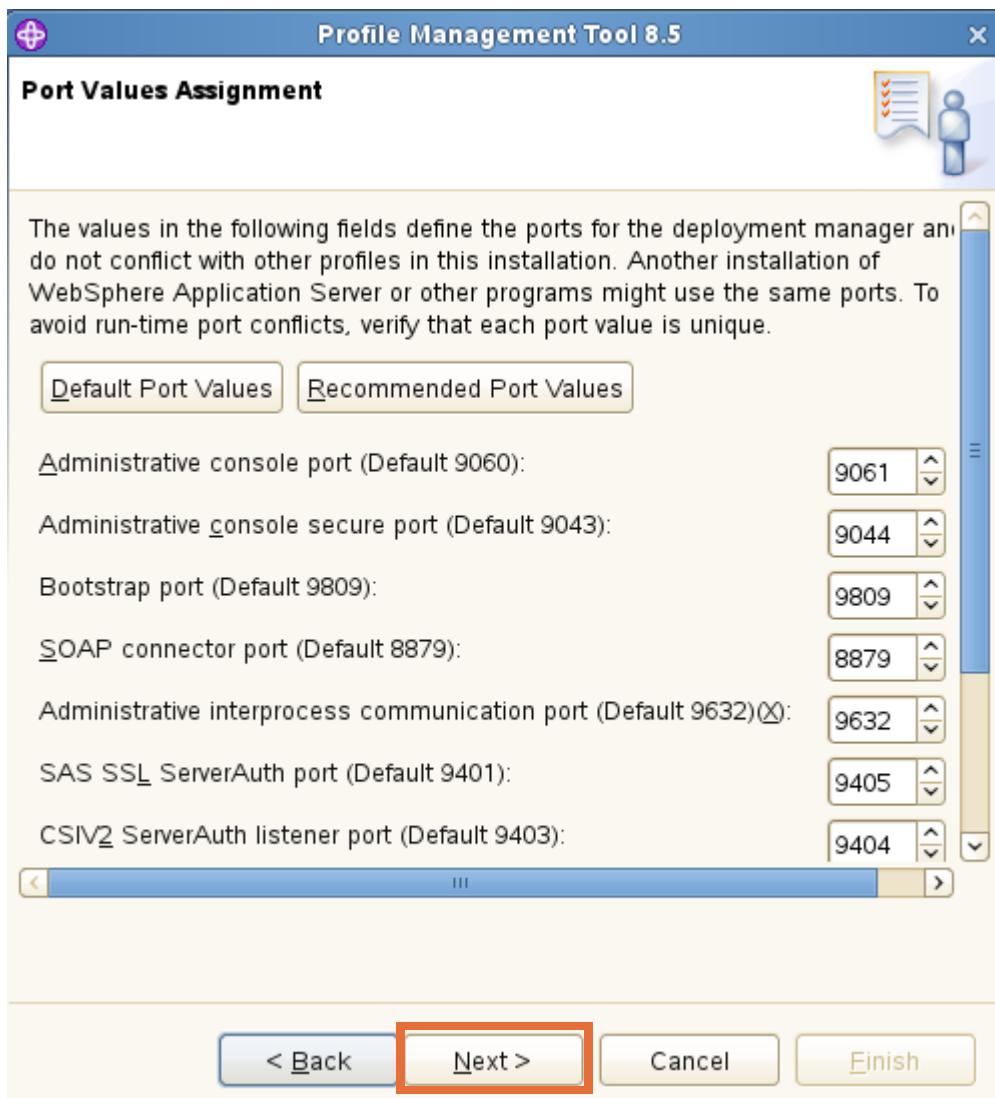


### Information

Note the administrative console port for the deployment manager. This port is used later in this exercise.

Ordinarily, the administrative console port would use port 9060. However, since a stand-alone application server is installed, the Profile Management Tool avoids reuse of any ports. It uses port 9061 instead.

- \_\_\_ o. Click **Next**.

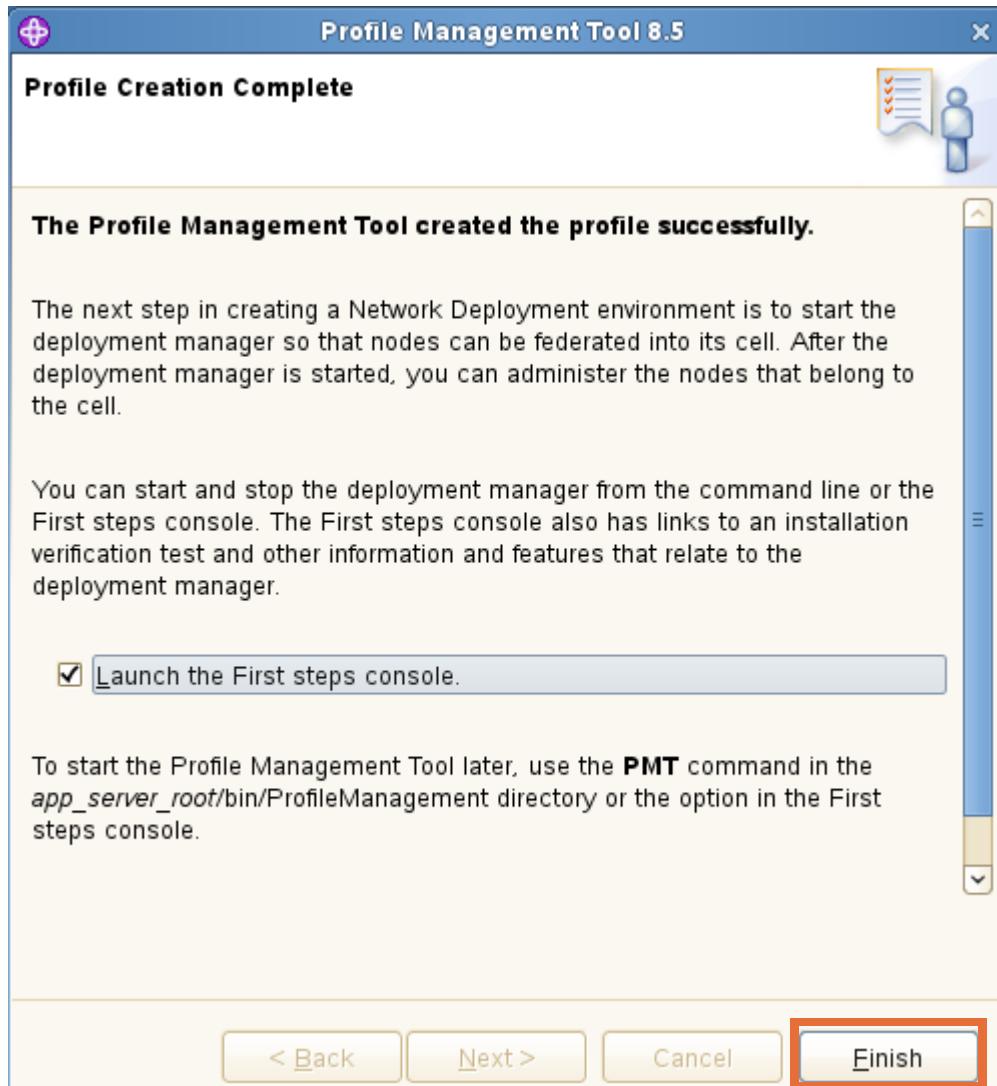


- \_\_\_ p. On the Linux Service Definition panel, take the defaults and click **Next**.

- \_\_\_ q. The Profile Creation Summary panel shows all of the choices you made on previous panels. Verify the summary information with what you entered previously. Click **Create**.

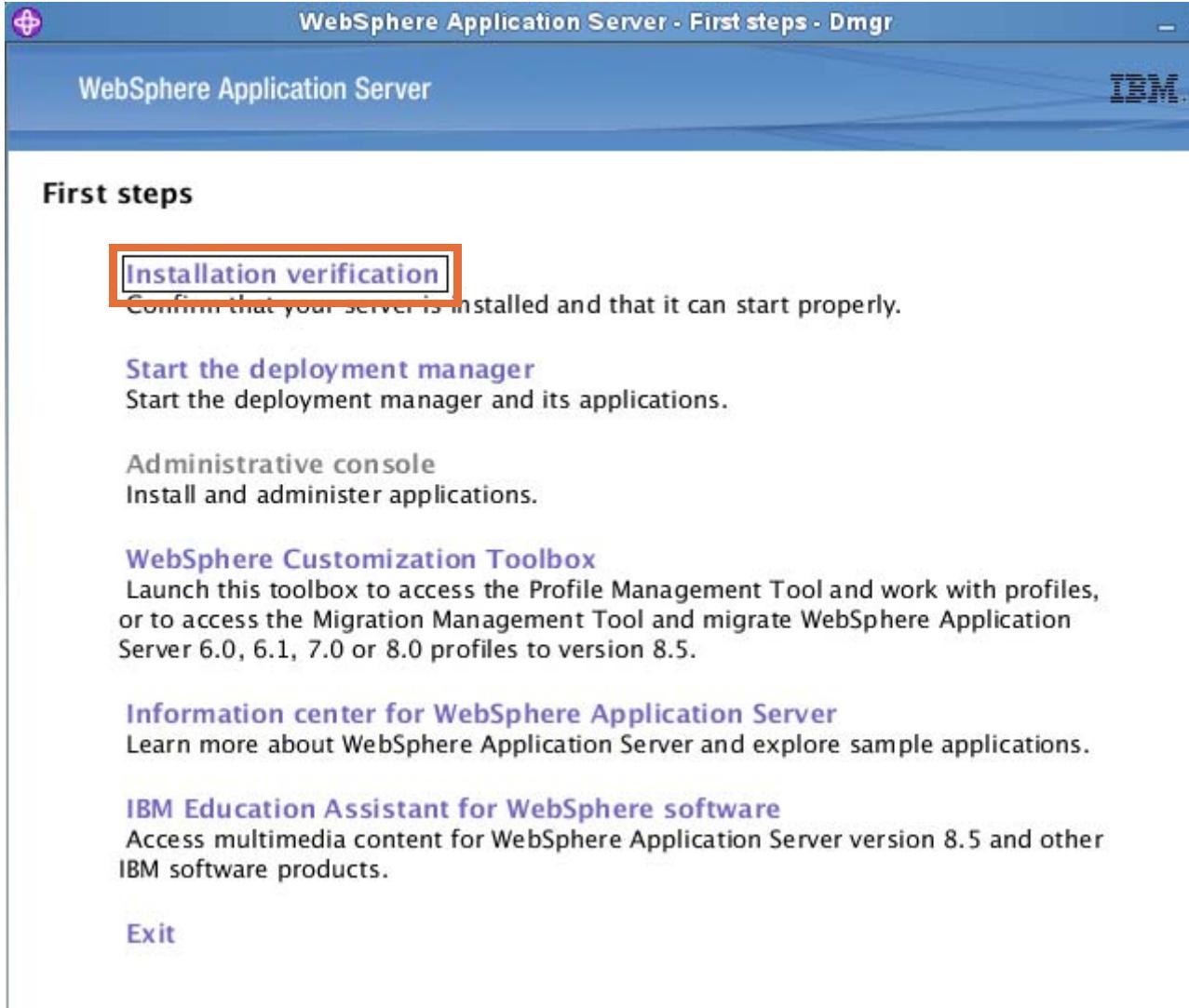
Creation of the profile usually takes several minutes to complete.

- \_\_\_ r. The profile creation completes and the Dmgr profile is created. Notice that the **Launch the First steps console** check box is selected. Click **Finish**, and the First steps console launches.



- \_\_\_ 3. The First steps console is associated with the deployment manager profile, Dmgr, that was created. Each profile has its own First steps console.

Click **Installation verification** from the First steps console.



The screenshot shows the 'WebSphere Application Server - First steps - Dmgr' interface. At the top, there's a navigation bar with icons for Home, Help, and Log Out. Below it, the title 'WebSphere Application Server' is displayed next to the 'IBM' logo. The main content area is titled 'First steps'. Inside this section, several links are listed: 'Installation verification' (which is highlighted with a red rectangular border), 'Start the deployment manager' (with the sub-instruction 'Start the deployment manager and its applications.'), 'Administrative console' (with the sub-instruction 'Install and administer applications.'), 'WebSphere Customization Toolbox' (with the sub-instruction 'Launch this toolbox to access the Profile Management Tool and work with profiles, or to access the Migration Management Tool and migrate WebSphere Application Server 6.0, 6.1, 7.0 or 8.0 profiles to version 8.5.'), 'Information center for WebSphere Application Server' (with the sub-instruction 'Learn more about WebSphere Application Server and explore sample applications.'), 'IBM Education Assistant for WebSphere software' (with the sub-instruction 'Access multimedia content for WebSphere Application Server version 8.5 and other IBM software products.'), and 'Exit'.

**First steps**

**Installation verification**

Confirm that your server is installed and that it can start properly.

**Start the deployment manager**

Start the deployment manager and its applications.

**Administrative console**

Install and administer applications.

**WebSphere Customization Toolbox**

Launch this toolbox to access the Profile Management Tool and work with profiles, or to access the Migration Management Tool and migrate WebSphere Application Server 6.0, 6.1, 7.0 or 8.0 profiles to version 8.5.

**Information center for WebSphere Application Server**

Learn more about WebSphere Application Server and explore sample applications.

**IBM Education Assistant for WebSphere software**

Access multimedia content for WebSphere Application Server version 8.5 and other IBM software products.

**Exit**

- \_\_\_ a. The installation verification test tool runs and shows messages to indicate verification status. Use the scroll bar to scroll to the bottom to see all the messages. The following messages are shown if the installation verification was successful:

IVTL00701: The Installation Verification Tool verification succeeded.

IVTL00801: The installation verification is complete.



### Information

It is possible that several warnings might be shown. These warning messages can be ignored.

- \_\_\_ b. Close the **First steps output - Installation verification** window.  
\_\_\_ c. Click **Exit** to close the First steps console.  
\_\_\_ d. Click **File > Exit** to close the WebSphere Customization Toolbox.

## Section 3: Back up the Dmgr profile configuration

Before continuing, it is a good practice to back up the configuration for the Dmgr profile that was created.

- \_\_\_ 1. Create a backup.



### Information

In a previous lab, the `backupConfig` command was used to create a backup. Another WebSphere tool makes backups of a profile as well (other than operating system-level backups). The `backupConfig` tool backs up only the configuration directory of a profile. The command `manageprofiles -backupProfile` backs up the configuration directory and other metadata.

The information center article on the `manageprofiles` command defines the `-backupProfile` attribute as follows:

This attribute forms a file system backup of a profile folder and the profile metadata from the profile registry file. Any servers that use the profile that you want to back up must first be stopped before starting the `manageprofiles` command with the

-backupProfile option. The -backupProfile parameter must be used with the -backupFile and -profileName parameters, for example:

```
./manageprofiles.sh -backupProfile -profileName
/opt/IBM/WebSphere/AppServer -backupFile <backupFile_name>
```

- \_\_ a. In a terminal window, navigate to the /opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin directory.
- \_\_ b. Verify the status of the deployment manager process by entering the following command:

```
./serverStatus.sh -all
```

Enter the user ID **wasadmin** and password **websphere** in the dialog box when you are prompted.

- \_\_ c. Stop the deployment manager process if it is running by entering the following command:

```
./stopManager.sh
```

Enter the user ID **wasadmin** and password **websphere** in the dialog box when prompted.

- \_\_ d. If the /usr/software/backups directory does not exist, create it.
- \_\_ e. After the deployment manager stops, enter the following command to back up the entire profile:

```
./manageprofiles.sh -backupProfile -profileName Dmgr -backupFile
/usr/software/backups/Dmgr_initial_backup.zip
```

Wait for the message:

INSTCONFSUCCESS: Success: The profile backup operation was successful.

- \_\_ 2. Since profile1 is federated later, create a backup for it as well.

- \_\_ a. Navigate to the /opt/IBM/WebSphere/AppServer/profiles/profile1/bin directory, and stop server1 by entering the following command:

```
./stopServer.sh server1
```

- \_\_ b. Make sure to enter the manageprofiles command from the <profile\_root>/profile1/bin directory. After server1 stops, enter the following command to back up the entire profile:

```
./manageprofiles.sh -backupProfile -profileName profile1
-backupFile /usr/software/backups/Profile1_prefederation.zip
```

Wait for the message:

INSTCONFSUCCESS: Success: The profile backup operation was successful.

- \_\_\_ 3. Start the deployment manager.
  - \_\_\_ a. In a terminal window, navigate to /opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin and enter the following command to start the deployment manager:
 

```
./startManager.sh
```

## **Section 4: Federate profile1 into the cell of the deployment manager**

During this section of the exercise, you federate the application server node, which profile1 defines (and is named `was85hostNode01`), into the cell named `was85hostCell01`, which the deployment manager profile defines. The federation process adds a node agent to the application server node.

- \_\_\_ 1. Verify that server `server1` is running.
  - \_\_\_ a. If you backed up the profile for profile1 in the previous section, the server `server1` must be started. Backing up a profile causes the profile server to stop. Navigate to the `/opt/IBM/WebSphere/AppServer/profiles/profile1/bin` folder and enter the following command:
 

```
./serverStatus.sh server1
```
  - \_\_\_ b. If `server1` is not running, start `server1`.
- \_\_\_ 2. Open the administrative console for the deployment manager.



### **Reminder**

The port in this case is not the default port, but instead is port 9061. This difference is because the system already has profile1, which is using port 9060, for its administrative console port. The profile creation process therefore chose the next available port (9061) for the deployment manager administrative console.

- \_\_\_ a. Open a web browser and enter the following address:  
`http://was85host:9061/ibm/console`
- \_\_\_ b. The browser might show a message that the server connection is untrusted. (This process is reviewed in an optional section of a previous exercise.) Click “**I understand the Risks**” and then click **Add Exception...**
- \_\_\_ c. Log in to the administrative console with `wasadmin` as the user name and `web1sphere` as the password.

\_\_\_ 3. Federate a node into the cell.

This process takes the existing application server within profile1 and federates it to the deployment manager. This action means that it no longer is a stand-alone application server, but instead is part of the newly created cell.



### Information

In this lab environment, synchronizing clocks is not an issue since the cell is running on a single computer. But, when federating distributed computers, it is necessary to make sure that the clocks on all nodes are within 5 minutes of each other.

- \_\_\_ a. From the deployment manager administrative console, click **System administration > Nodes**.
- \_\_\_ b. Click **Add Node**.

#### Nodes

Use this page to manage nodes in the application server environment. A node corresponds to a physical computer system with a distinct IP host address. The following table lists the managed and unmanaged nodes in this cell. The first node is the deployment manager. Add new nodes to the cell and to this list by clicking Add Node.

The screenshot shows the 'Nodes' page of the WebSphere Application Server administrative console. At the top, there is a toolbar with buttons for Preferences, Add Node (which is highlighted with a red box), Remove Node, Force Delete, Synchronize, Full Resynchronize, and Stop. Below the toolbar is a row of icons for managing nodes. The main area has columns for Select, Name, Host Name, Version, Discovery Protocol, and Status. A message says 'You can administer the following resources:' followed by a table row for 'was85hostCellManager01'. At the bottom, it says 'Total 1'.

| Select | Name                                   | Host Name | Version    | Discovery Protocol | Status |
|--------|----------------------------------------|-----------|------------|--------------------|--------|
|        | <a href="#">was85hostCellManager01</a> | was85host | ND 8.5.5.0 | TCP                |        |

- \_\_\_ c. The **Managed node** option is selected as default. A managed node contains an application server and a node agent. The application server runs as part of the network deployment environment. Keep the default setting and click **Next**.

Use this page to add either a managed or an unmanaged node.

Managed node  
Specifies the creation of a managed node. A managed node contains an application server process that runs within the deployment manager cell. The managed node is associated with a node agent process that maintains the configuration for the node and controls its operation. Choosing this option results in running the add node utility to federate an existing stand-alone application server.

Unmanaged node  
Specifies the creation of an unmanaged node. An unmanaged node represents a node in the topology that does not have an application server process or a node agent process. Unmanaged nodes are for other server processes, such as web servers that exist on their own node in the topology.

Recover an existing node  
Specifies to replace a damaged node in the cell. First, create a new profile to replace the damaged node and give it the same profile and node names. Then use this option to replace the damaged node in the cell with the new node.

**Next** | **Cancel**



### Information

As the description on the screen capture indicates, a managed node is a node with an application server and a node agent that belongs to a deployment manager cell. However, a managed node contains a node agent, but initially might not contain an application server.

In this part of the exercise, you are adding a stand-alone node. Adding a managed node in this way, you have an application server, server1, and it is the node agent process that makes it a managed node.

- \_\_\_ d. Enter your host name `was85host` for the host. Specify security user names and passwords for both profile1 and the deployment manager. Enter `wasadmin` and `web1sphere` for the user name and password.

- \_\_ e. Select the options to **Include applications** and **Include buses**. Keep all remaining defaults.

Add Managed Node

Use this page to identify a stand-alone application server process that is running. Start the application server, if necessary, or add the node from the command line by running the addNode command from the bin directory of the stopped application server profile.

**Node connection**

\* Host: was85host

\* JMX connector type: SOAP

\* JMX connector port: 8880

Application server user name: wasadmin

Application server password: \*\*\*\*\*

\* Deployment manager user name: wasadmin

\* Deployment manager password: \*\*\*\*\*

Config URL: file:\${USER\_INSTALL\_ROOT}/properties/sas.client.props

**Options**

Include applications

Include buses

- \_\_ f. Click **OK**. The federation process can take several minutes to complete.



### Information

The port number is the JMX connector port of the node you want to federate to the cell.



### Note

The **Include buses** option is selected in this example, which specifies whether to move the bus configuration at the node to the deployment manager. However, no buses currently exist in the cell. When federating a stand-alone node, it is a good idea to select this option to ensure that any bus configuration at the node level is moved over into the cell.

**Note**

If you see the following message in the console, proceed as described:

"The console has not received information for the add operation in a timely manner. The state of the operation is indeterminate. Check the add node log for details."

Check the `/opt/IBM/WebSphere/AppServer/profiles/profile1/logs/addNode.log` file for error messages. Look for a message that indicates the node is successfully federated.

If you see the following message, proceed as explained:

ADMC0009E: The system failed to make the SOAP RPC call: invoke

- One option is to increase the timeout value for the type of connection you are using, in this case SOAP, by editing the file `/opt/IBM/WebSphere/AppServer/profiles/profile1/properties/soap.client.properties` and changing the value of the `com.ibm.SOAP.requestTimeout` property to 6000. The value is in seconds.
- If using an RMI connection, edit the file `/opt/IBM/WebSphere/AppServer/profiles/profile1/properties/sas.client.props`, and increase the value for the `com.ibm.CORBA.requestTimeout` property to 6000.

Repeat the previous steps to add a node.

\_\_\_ 4. Verify the cell configuration.

- \_\_\_ a. Click **System Administration > Nodes**. Two nodes are listed: the deployment manager (`was85hostCellManager01`), and the `was85hostNode01` node that was added.

#### Nodes

Use this page to manage nodes in the application server environment. A node corresponds to a physical computer system with a distinct IP host address. The following table lists the managed and unmanaged nodes in this cell. The first node is the deployment manager. Add new nodes to the cell and to this list by clicking Add Node.

Preferences

| Add Node                                    | Remove Node                            | Force Delete | Synchronize | Full Resynchronize | Stop   |
|---------------------------------------------|----------------------------------------|--------------|-------------|--------------------|--------|
|                                             |                                        |              |             |                    |        |
| Select                                      | Name                                   | Host Name    | Version     | Discovery Protocol | Status |
| You can administer the following resources: |                                        |              |             |                    |        |
| <input checked="" type="checkbox"/>         | <a href="#">was85hostCellManager01</a> | was85host    | ND 8.5.5.0  | TCP                |        |
| <input type="checkbox"/>                    | <a href="#">was85hostNode01</a>        | was85host    | ND 8.5.5.0  | TCP                |        |
| Total 2                                     |                                        |              |             |                    |        |

- \_\_\_ b. Click **System Administration > Node agents**. Verify that the node agent on `was85hostNode01` started.

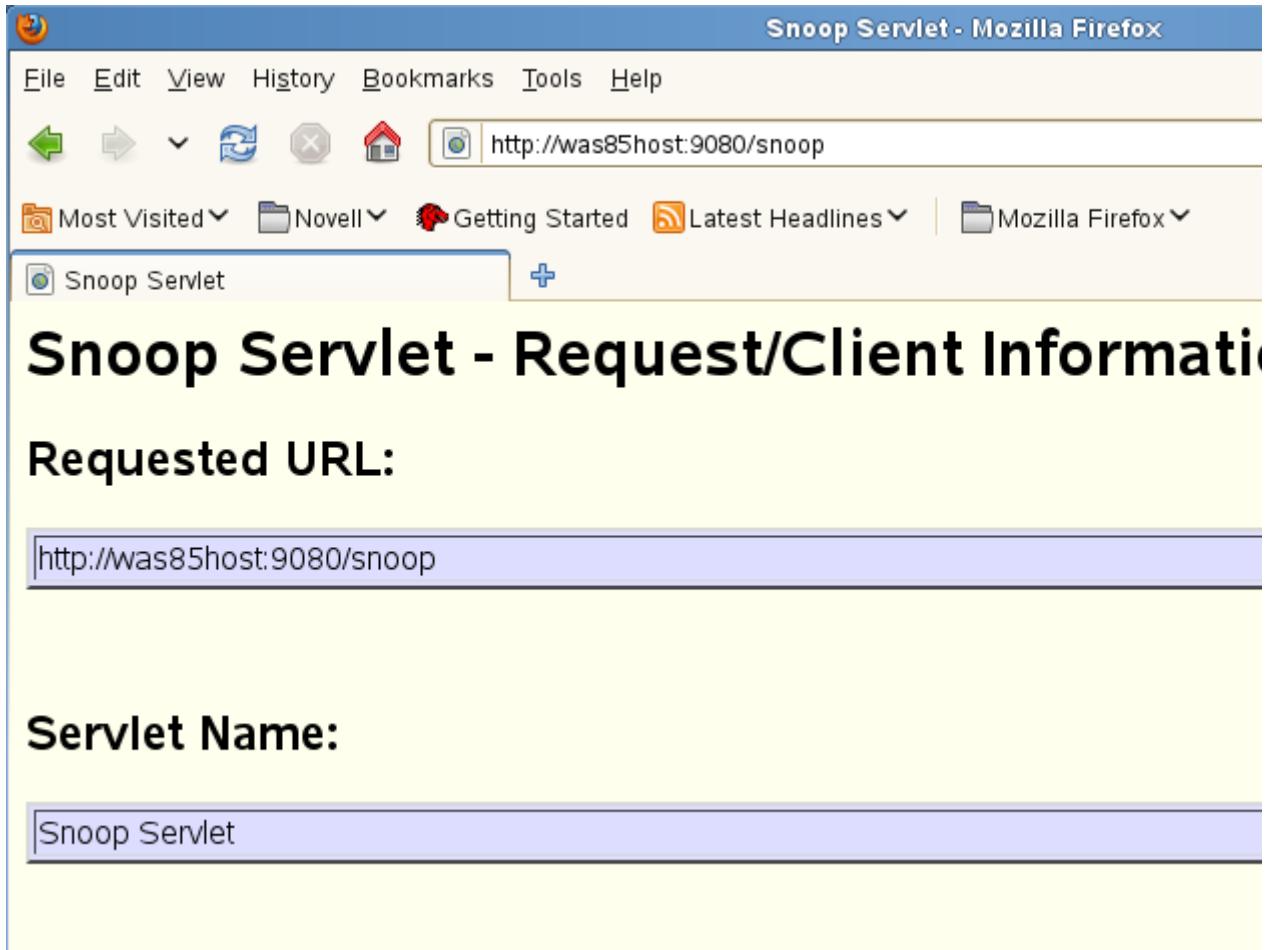
#### Node agents

Use this page to manage node agents and application servers on the node that a node agent manages. The node agent process serves as an intermediary between the application servers on the node and the deployment manager. The node agent process runs on every node and is specialized to perform node-specific administration functions, such as server process monitoring, configuration synchronization, file transfer, and request routing.

Preferences

| Stop                                        | Restart                   | Restart all Servers on Node |           |            |        |
|---------------------------------------------|---------------------------|-----------------------------|-----------|------------|--------|
|                                             |                           |                             |           |            |        |
| Select                                      | Name                      | Node                        | Host Name | Version    | Status |
| You can administer the following resources: |                           |                             |           |            |        |
| <input type="checkbox"/>                    | <a href="#">nodeagent</a> | was85hostNode01             | was85host | ND 8.5.5.0 |        |
| Total 1                                     |                           |                             |           |            |        |

5. Start the application server and test the Snoop servlet.
  - a. Click **Servers > Server Types > WebSphere application servers**.
  - b. Select **server1** and click **Start**. Wait for server1 to start.
  - c. Click **Applications > Application Types > WebSphere enterprise applications**. Check the status for the DefaultApplication. Verify that the DefaultApplication is running.
  - d. Open another browser window and enter the following address:  
**http://was85host:9080/snoop**



The screenshot shows a Mozilla Firefox browser window titled "Snoop Servlet - Mozilla Firefox". The address bar contains "http://was85host:9080/snoop". The main content area displays the following text:

**Snoop Servlet - Request/Client Information**

**Requested URL:**

**Servlet Name:**

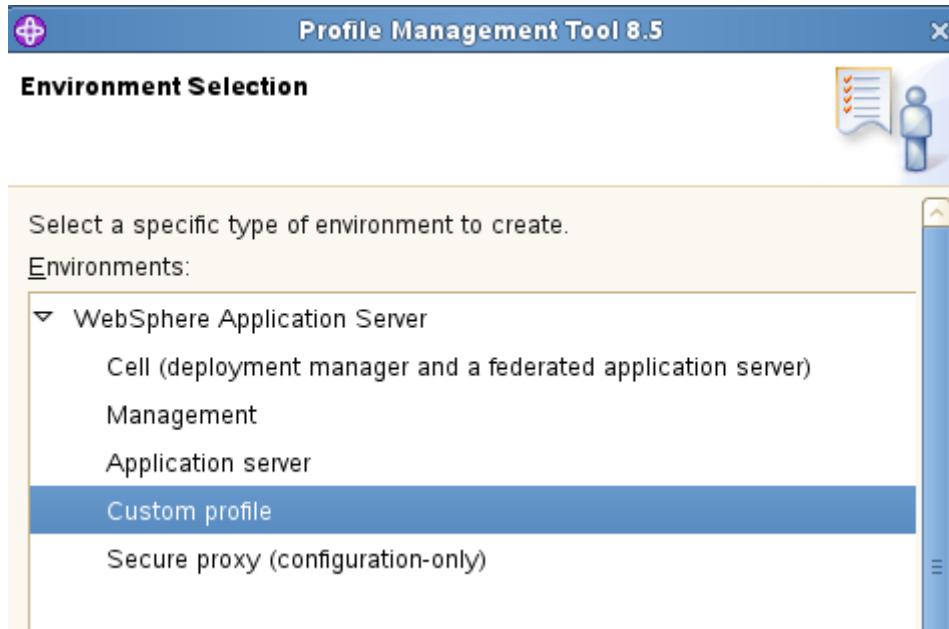
If you get information back regarding the server, it verifies that the Snoop servlet works.

### **Section 5: Create a custom profile and federate it into the deployment manager cell**

During this section of the exercise, you are going to create a custom profile, `profile2`, that defines a node, named `was85hostNode02`. The custom profile is automatically federated into the cell `was85hostCell01`.

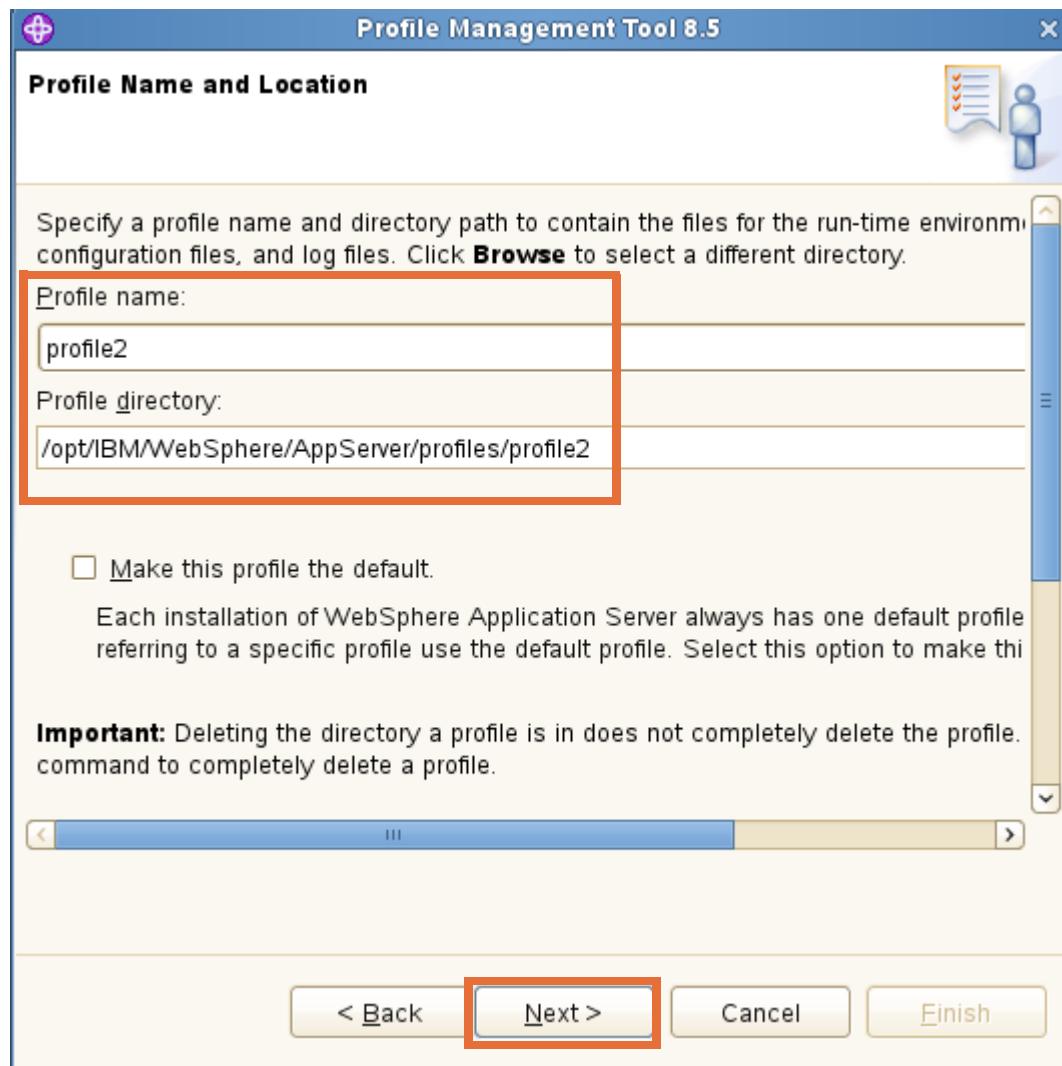
A custom profile is useful because it does not create any application servers on the node; it creates the configuration and the node agent only. Consequently, no server1 is created on that node. This feature is helpful for expanding clusters.

- \_\_ 1. Start the WebSphere Customization Toolbox.
  - \_\_ a. Enter the following command to start the WebSphere Customization Toolbox:  
`/opt/IBM/WebSphere/AppServer/bin/ProfileManagement/wct.sh`
  - \_\_ b. The Welcome window for the WebSphere Customization Toolbox opens, and you are placed on the Profile Management Tool tab.
- \_\_ 2. Create a custom profile named profile2, and federate it to the deployment manager's configuration.
  - \_\_ a. Click **Create** from the Profiles list panel.
  - \_\_ b. On the Environment Selection panel, click **Custom profile**. Click **Next**.



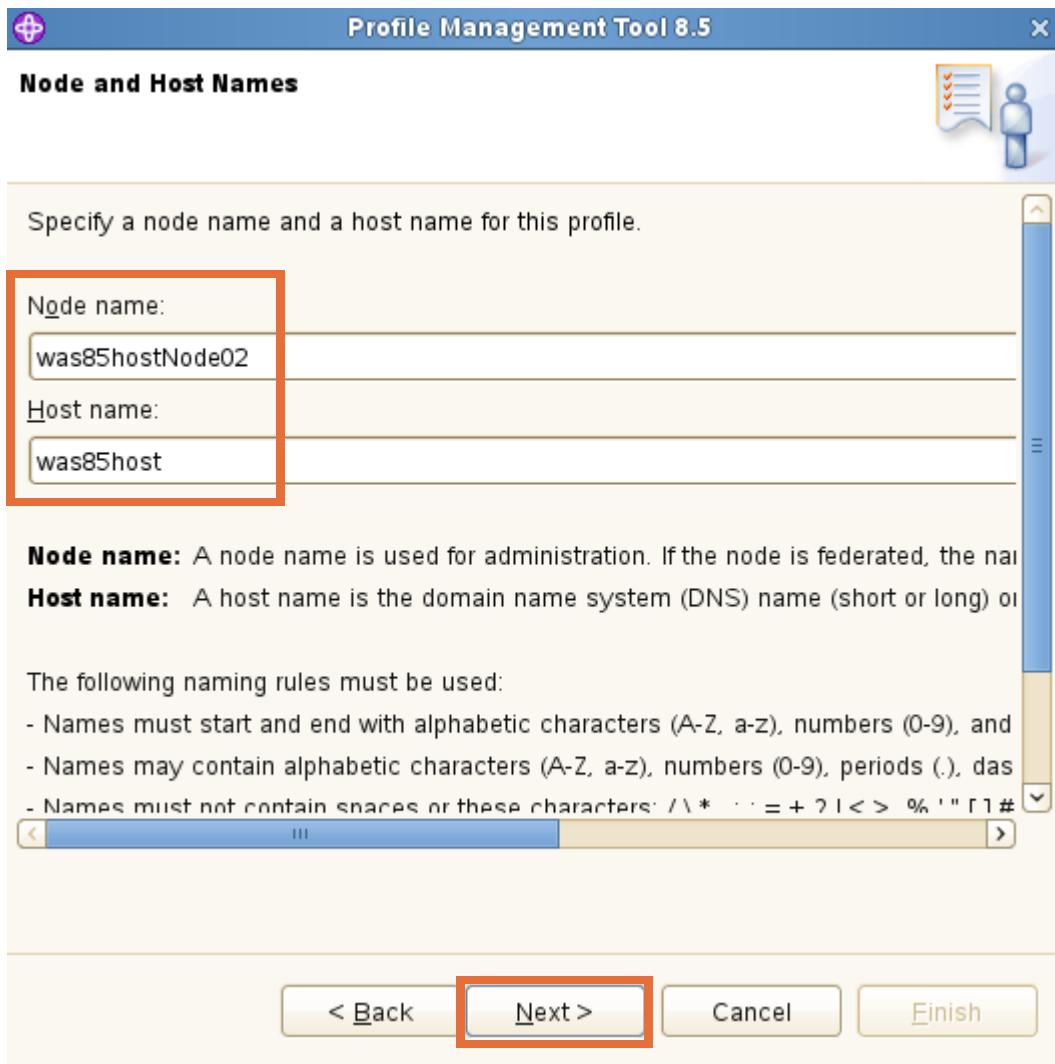
- \_\_ c. On the Profile Creation Options page, select **Advanced profile creation** to specify your own configuration values during profile creation. Click **Next**.

- \_\_\_ d. For the profile name and location, enter the following information:
- Profile name: profile2
  - Profile directory: /opt/IBM/WebSphere/AppServer/profiles/profile2



- \_\_\_ e. Click **Next**.

- \_\_\_ f. Ensure that the **Node name** and **Host name** are was85hostNode02 and was85host (they are based on the short form of the host name and not localhost: for example, was85hostNode02). Notice that the **Node name** ends with a **02** (not **01**, as that would conflict with the existing node). Click **Next**.



### Information

For these labs, use the short name for your host. Although it is acceptable to use another form of the host name, it is important to be consistent. Since the short name was used in the initial WebSphere installation lab, the short name is used here as well.

\_\_ g. On the Federation panel, enter the following information:

- Deployment manager host name: was85host
- User name: wasadmin
- Password: web1sphere
- Do not click the **Federate this node later** check box.

**Federation**

Deployment manager host name or IP address:  
was85host

Deployment manager SOAP port number (Default 8879):  
8879

Deployment manager authentication

Provide a user name and password that can be authenticated, if administrative security is enabled for the deployment manager.

User name:  
wasadmin

Password:  
[REDACTED]

Federate this node later.

You must federate this node later using the **addNode** command if the deployment manager:

- is not running
- has the SOAP connector disabled

\_\_ h. Click **Next**.



### Information

By not selecting **Federate this node later**, the wizard process federates the node now.

- \_\_ i. Accept the defaults on the Security Certificate (Part 1) panel. Click **Next**.
- \_\_ j. Accept the defaults on the Security Certificate (Part 2) panel. Click **Next**.

- \_\_ k. Accept the default on the Port Values Assignment panel. Click **Next**.
  - \_\_ l. On the Profile Creation Summary panel, click **Create**.
  - \_\_ m. After a couple of minutes, the profile creation is complete. On the Profile Creation Complete panel, clear the **Launch the First steps console** option. Do not use the First steps console for profile2.
  - \_\_ n. Click **Finish** to exit the wizard.
  - \_\_ o. **Close** the WebSphere Customization Toolbox.
- \_\_ 3. Verify that the node is added to the deployment manager's configuration.
- \_\_ a. Using the administrative console, click **System administration > Nodes**. The federated node was85hostNode02 is listed. If the node is not displayed, refresh the administrative console.

#### Nodes

Use this page to manage nodes in the application server environment. A node corresponds to a physical computer system with a distinct IP host address. The following table lists the managed and unmanaged nodes in this cell. The first node is the deployment manager. Add new nodes to the cell and to this list by clicking Add Node.

| Select                   | Name                                   | Host Name | Version    | Discovery Protocol | Status |
|--------------------------|----------------------------------------|-----------|------------|--------------------|--------|
| <input type="checkbox"/> | <a href="#">was85hostCellManager01</a> | was85host | ND 8.5.5.0 | TCP                |        |
| <input type="checkbox"/> | <a href="#">was85hostNode01</a>        | was85host | ND 8.5.5.0 | TCP                |        |
| <input type="checkbox"/> | <a href="#">was85hostNode02</a>        | was85host | ND 8.5.5.0 | TCP                |        |

Total 3



#### Information

Using a custom profile does not create a server instance. This feature is useful when adding nodes to a cell. The intention of federating a new node into a cell is normally to either add cluster members to the node or create servers named something other than server1.

- \_\_ 4. Verify that both node agents started.
  - \_\_ a. From the navigation tree, click **Node agents**.
  - \_\_ b. Start the node agent that is not started by entering the following command in a terminal window:

/opt/IBM/WebSphere/AppServer/profiles/profileX/bin/startNode.sh  
(where X is 1 or 2)

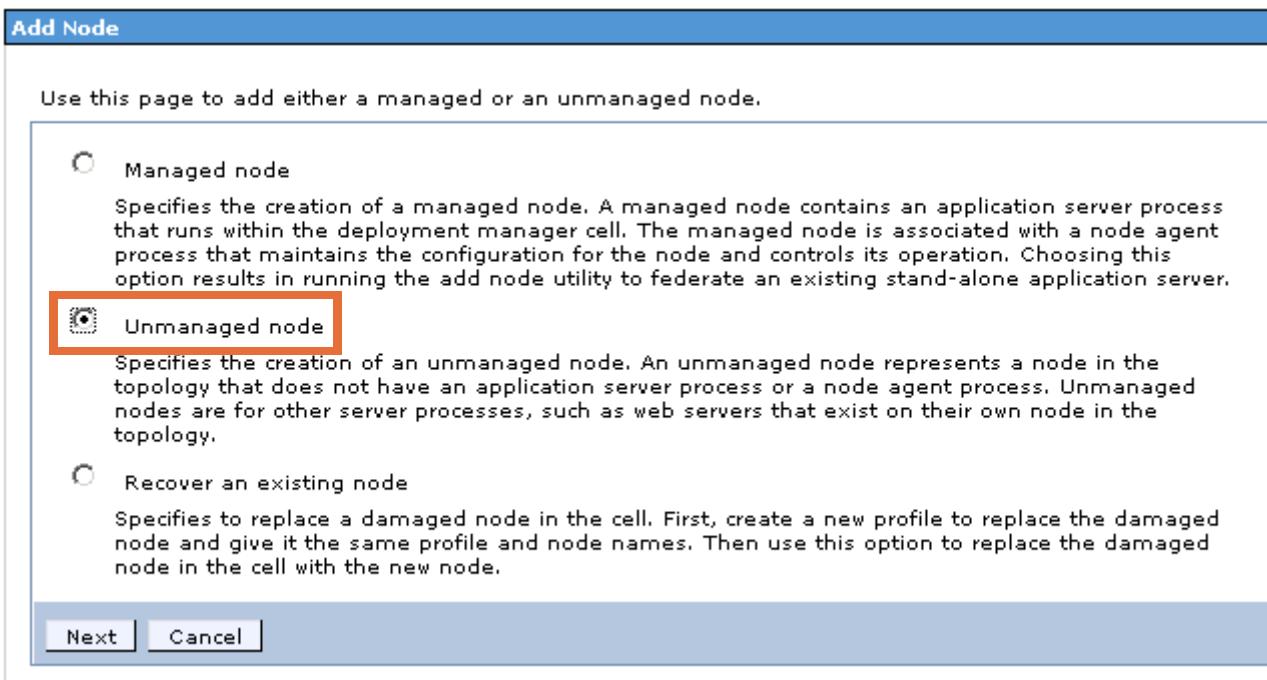
Wait for the node agent to start.

## Section 6: Add the IBM HTTP Server to the cell

During this section of the exercise, you add an unmanaged node, ihsnode, to the cell was85hostCell01. You also add a web server, webserver1, to the unmanaged node. Information about the web server is communicated to the deployment manager through the IBM HTTP Server administrative process.

Create a node and add the web server to the node. When adding a node, you can create either a managed node or an unmanaged node. A managed node contains a WebSphere Application Server and a node agent. An unmanaged node does not have a node agent and is used for defining remote web servers in the topology.

- \_\_\_ 1. Create a new unmanaged node for the web server. The web server definition uses this new node definition to define the host on which it lives.
  - \_\_\_ a. Click **System administration > Nodes**.
  - \_\_\_ b. Click **Add Node**.
  - \_\_\_ c. In the Add Node window, select **Unmanaged node** and click **Next**.



\_\_ d. In the Nodes window, enter configuration information for the node:

- Name: ihsnode
- Host Name: was85host
- Platform Type: Linux

[Nodes > New...](#)

Use this page to view or change the configuration for an unmanaged node. An unmanaged node is a node defined in the topology that does not have a node agent running to manage the process. Unmanaged nodes are typically used to manage servers.

The screenshot shows a configuration dialog for a new node. The 'General Properties' section is highlighted with a red box. It contains three fields: 'Name' (ihsnode), 'Host Name' (was85host), and 'Platform Type' (Linux). Below these fields are four buttons: 'Apply', 'OK', 'Reset', and 'Cancel'. To the right of the fields, a note states: 'The additional properties will not be available until the general properties for this item are applied or saved.' Below this note is a 'Custom Properties' section with a single button: 'Custom Properties'.

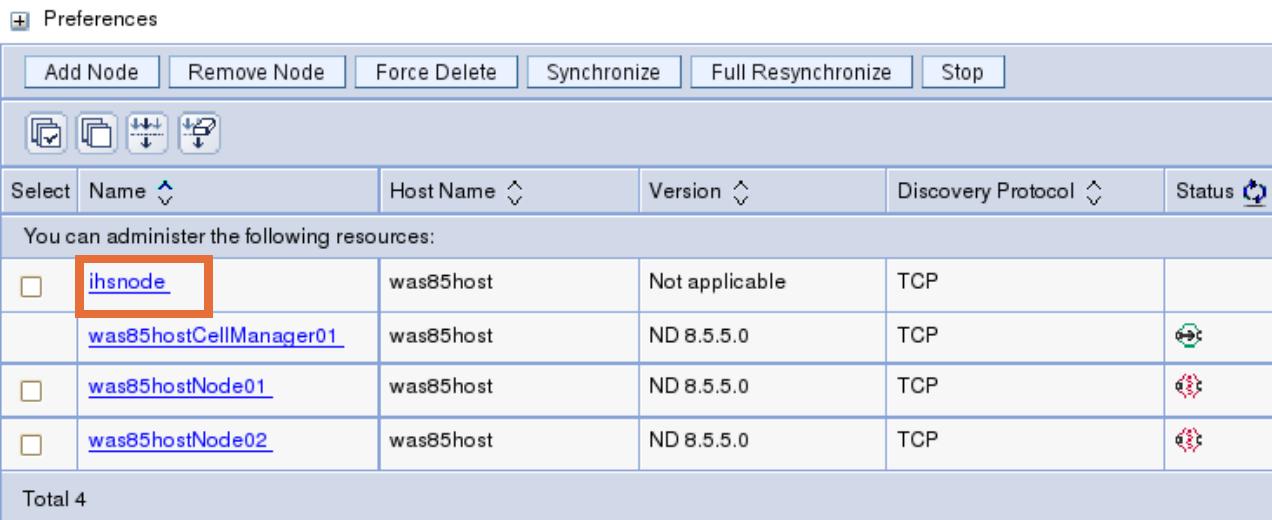
\_\_ e. Click **OK**.

\_\_ f. **Save** the changes.

- \_\_\_ g. The node **ihsnode** is now shown in the list of nodes.

#### Nodes

Use this page to manage nodes in the application server environment. A node corresponds to a physical computer system with a distinct IP host address. The following table lists the managed and unmanaged nodes in this cell. The first node is the deployment manager. Add new nodes to the cell and to this list by clicking Add Node.



The screenshot shows the 'Nodes' administrative console interface. At the top, there are buttons for 'Add Node', 'Remove Node', 'Force Delete', 'Synchronize', 'Full Resynchronize', and 'Stop'. Below these are four icons: a magnifying glass, a folder, a double arrow, and a double arrow with a minus sign. The main area has columns for 'Select', 'Name', 'Host Name', 'Version', 'Discovery Protocol', and 'Status'. A message says 'You can administer the following resources:'. Below is a table with four rows:

| Select                   | Name                                   | Host Name | Version        | Discovery Protocol | Status |
|--------------------------|----------------------------------------|-----------|----------------|--------------------|--------|
| <input type="checkbox"/> | <a href="#">ihsnode</a>                | was85host | Not applicable | TCP                |        |
| <input type="checkbox"/> | <a href="#">was85hostCellManager01</a> | was85host | ND 8.5.5.0     | TCP                |        |
| <input type="checkbox"/> | <a href="#">was85hostNode01</a>        | was85host | ND 8.5.5.0     | TCP                |        |
| <input type="checkbox"/> | <a href="#">was85hostNode02</a>        | was85host | ND 8.5.5.0     | TCP                |        |

Total 4

## Section 7: Add the web server to the configuration

In this section, the web server definition is added to the ihsnode.

- \_\_\_ 1. Add the web server to the ihsnode configuration. This action allows the web server to be managed from the administrative console.
- \_\_\_ a. Click **Servers > Server Types > Web servers**.
- \_\_\_ b. Click **New** to add a web server.
- \_\_\_ c. On **Step 1** of creating a web server, enter the following information:
- Select **ihsnode** from the **Select node** list.
  - For **Server name**, enter: `webserver1`
  - Select **IBM HTTP Server** from the **Type** list.



### Information

The web server name must match the name that you assigned during the IBM HTTP Server installation. You can check the web server name by looking in `/opt/IBM/WebSphere/Plugins/config/`.

**Click Next.**

Create new Web server definition

Use this page to create a new web server.

→ Step 1: Select a node for the Web server and select the Web server type

Step 2: Select a Web server template

Step 3: Enter the properties for the new Web server

Step 4: Confirm new Web server

Select a node for the Web server and select the Web server type

Select a node that corresponds to the Web server you want to add.

Select node

ihsnode

\* Server name  
webserver1

\* Type  
IBM HTTP Server

Next | Cancel

- \_\_ d. On **Step 2** of selecting a web server template, verify that **IHS** is selected and click **Next**.

Create new Web server definition

Use this page to create a new web server.

Step 1: Select a node for the Web server and select the Web server type

→ Step 2: Select a Web server template

Step 3: Enter the properties for the new Web server

Step 4: Confirm new Web server

Select a Web server template

Select the template that corresponds to the server that you want to create.

| Select                           | Template Name | Type   | Description                 |
|----------------------------------|---------------|--------|-----------------------------|
| <input checked="" type="radio"/> | IHS           | System | The IHS Web Server Template |

Previous | Next | Cancel

- \_\_ e. On **Step 3**, specify the properties for the new web server. Enter the following information in the fields as provided:

**Table 6: Web server configuration details**

| Field name                       | Value               |
|----------------------------------|---------------------|
| Port                             | 80                  |
| Web server Installation location | /opt/IBM/HTTPServer |

**Table 6: Web server configuration details**

| <b>Field name</b>             | <b>Value</b>               |
|-------------------------------|----------------------------|
| Plug-in installation location | /opt/IBM/WebSphere/Plugins |
| Application mapping           | All                        |
| Port                          | 8008                       |
| User name                     | ihsadmin                   |
| Password                      | web1sphere                 |
| Confirm password              | web1sphere                 |

**Warning**

The plug-in installation location defaults to an incorrect value. The default value is /opt/IBM/HTTPServer/Plugins, but the correct value for the Plug-in installation location is /opt/IBM/WebSphere/Plugins.

It is important that the directories entered for the details are the same directories that are used during the web server and plug-in installation in Exercise 3.

Create new Web server definition

Use this page to create a new web server.

Step 1: Select a node for the Web server and select the Web server type  
Step 2: Select a Web server template  
→ Step 3: Enter the properties for the new Web server  
Step 4: Confirm new Web server

**Enter the properties for the new Web server**

Enter the Web server properties.

\* Port

\* Web server installation location

\* Plug-in installation location

Application mapping to the Web server

Enter the IBM Administration Server properties.

\* Administration Server Port

\* Username

\* Password

\* Confirm password

Use SSL

[Previous] [Next] [Cancel]

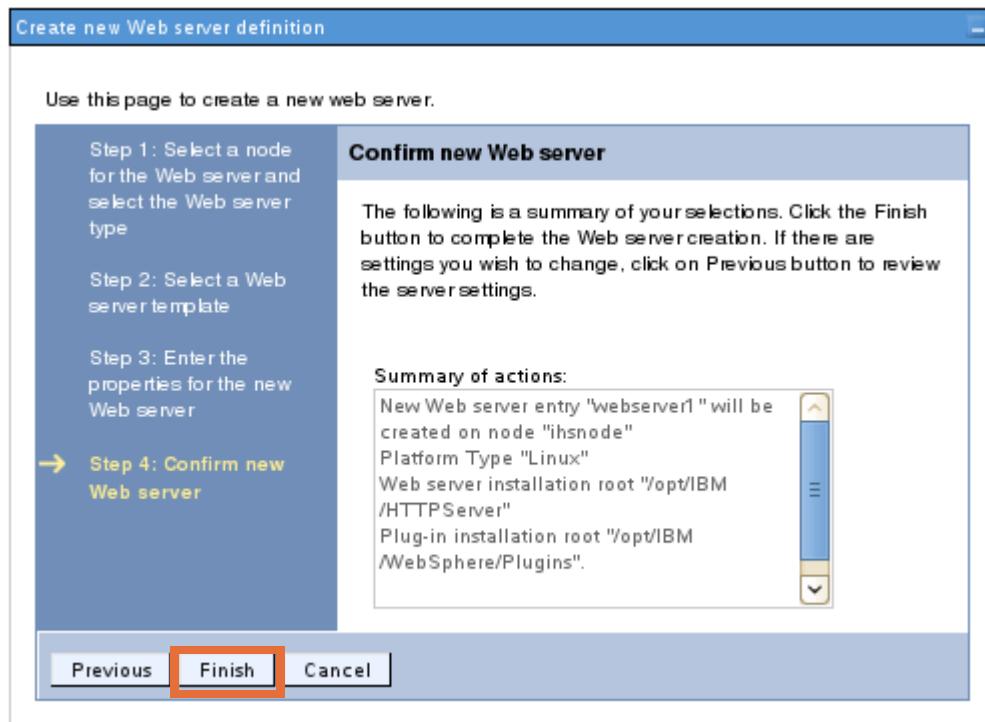


### Important

Make sure that you modify the default value for the plug-in installation location to the correct location.

- \_\_ f. Click **Next** when complete.

- \_\_ g. On Step 4, the summary, click **Finish**.



- \_\_ h. **Save** the changes.
- \_\_ i. Minimize the administrative console browser window.
- \_\_ 2. Use a terminal window to verify the web server status.
- \_\_ a. Open a terminal window and navigate to:  
/opt/IBM/HTTPServer/bin/
- \_\_ b. Use the following command to check whether the web server processes are running:  
`ps -ef | grep -i httpd`
- \_\_ c. If the web server or IBM HTTP Server administrative process is not running, you can use the following commands to start them:  
`./apachectl start`  
`./adminctl start`
- \_\_ d. Again, use the following command to check whether the web server processes are running:  
`ps -ef | grep -i httpd`  
Verify that several processes are now running.
- \_\_ e. Exit the terminal window.

- \_\_\_ 3. Use the administrative console to examine the web server status.
  - \_\_\_ a. Maximize the administrative console browser window.
  - \_\_\_ b. The web server status should be Started. If it indicates a Stopped status, refresh the administrative console.
  - \_\_\_ c. To verify you can use the administrative console, select the box next to **webserver1** and click **Stop**. Wait for the web server to stop.
  - \_\_\_ d. Start the web server again, select the box next to **webserver1**, and click **Start**. Wait for the web server to start.
  - \_\_\_ e. Another way to verify that the web server started is to open a web browser and connect to the IBM HTTP Server welcome page. Enter the following address:

`http://was85host`

## **Section 8: Mapping modules to servers**

Each module of an application is mapped to one or more target servers. The target server can be an application server, a cluster of application servers, or a web server. Web servers that are specified as targets have the routing information for the application, which is generated in their plug-in configuration files.

This mapping usually takes place during application deployment. But since the DefaultApplication is deployed when this particular web server was added, the DefaultApplication still must be mapped to your new web server. That, in fact, is done for you during the last step of defining the web server properties when **All** is selected for the **Application mapping to the web server**. That step mapped all installed applications to the new web server.

This section of the lab verifies that the applications are correctly mapped to the new web server.

- \_\_\_ 1. Using the deployment manager administrative console, verify the mapping of the DefaultApplication modules to the web server.
  - \_\_\_ a. From the administrative console, click **Applications > Application Types > WebSphere enterprise applications**.
  - \_\_\_ b. Click **DefaultApplication**.

- \_\_\_ c. Under Modules, click **Manage Modules**.

[Enterprise Applications](#) > [DefaultApplication](#)

Use this page to configure an enterprise application. Click the links to access pages for further configuring of the application or its modules.

The screenshot shows the 'Default Application' configuration page. The 'Configuration' tab is selected. In the 'General Properties' section, the 'Name' field is set to 'DefaultApplication'. In the 'Modules' section, the 'Manage Modules' link is highlighted with a red box. Below it are 'Display module build IDs' and 'Web Module Properties' (which contains a 'Session management' link). The 'Web Module Properties' section is also highlighted with a red box.

- \_\_\_ d. Notice that the Default Web Application module maps to both the application server **server1** and the **webserver1**. This page can also be used to modify the mappings manually if they do not exist.

Configure Modules

Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed through.

Targets and servers:

The screenshot shows the 'Manage Modules' configuration page. It includes a 'Targets and servers' section with dropdown menus for 'Cell' (set to 'WebSphere:cell=was85hostCell01') and 'Node' (set to 'ihsnode'). Below this are 'Remove', 'Update', 'Remove File', and 'Export File' buttons. A table lists module mappings:

| Module                                  | URI                                       | Module Type | Server                                                                                                                              |
|-----------------------------------------|-------------------------------------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Increment EJB module</a>    | Increment.jar,META-INF/ejb-jar.xml        | EJB Module  | WebSphere:cell=was85hostCell01,node=ihsnode,server=webserver1<br>WebSphere:cell=was85hostCell01,node=was85hostNode01,server=server1 |
| <a href="#">Default Web Application</a> | DefaultWebApplication.war,WEB-INF/web.xml | Web Module  | WebSphere:cell=was85hostCell01,node=ihsnode,server=webserver1<br>WebSphere:cell=was85hostCell01,node=was85hostNode01,server=server1 |

- \_\_\_ e. Click **DefaultApplication** in the breadcrumb trail to return to the configuration window.

- \_\_ f. Under Detail Properties, click **Target specific application status**.

**General Properties**

- \* Name: DefaultApplication
- Application reference validation: Issue warnings

**Detail Properties**

- Target specific application status** (highlighted with a red box)
- Startup behavior

**Modules**

- Manage Modules
- Display module build IDs

**Web Module Properties**

- Session management
- Context Root For Web Modules
- JSP and JSF options
- Virtual hosts

- \_\_ g. This view shows the mapping of a deployed object to servers.

[Enterprise Applications](#) > [DefaultApplication](#) > Target specific application status

Use this page to view a mapping of a deployed object, such as an application or module, into a target server or cluster environment. This page displays the status of the enterprise application or module on each server or cluster.

| Select                   | Target                     | Node            | Version        | Auto Start | Application Status |
|--------------------------|----------------------------|-----------------|----------------|------------|--------------------|
| <input type="checkbox"/> | <a href="#">server1</a>    | was85hostNode01 | ND 8.5.0.0     | Yes        |                    |
| <input type="checkbox"/> | <a href="#">webserver1</a> | ihsnode         | Not applicable | Yes        |                    |

Total 2

## Section 9: Working with the plug-in configuration file

The plug-in configuration file contains routing information for all applications that are mapped to the web server. The plug-in configuration file must be regenerated and propagated to the web server whenever changes that are made to the WebSphere configuration affect how requests are routed from the web server to the application server.

1. Regenerate the plug-in configuration file. This process generates a plug-in configuration file that is specific to the web server that is defined within the cell. If multiple web servers are defined within the cell, you can generate customized plug-in configuration files for each of those web servers.
  - a. From the administrative console, click **Servers > Server Types > Web servers**.

- \_\_ b. Select the web server and click **Generate Plug-in**.

#### Web servers

Use this page to view a list of the installed web servers.

Preferences

| Select                              | Name       | Web server Type | Node    | Host Name | Version        | Status |
|-------------------------------------|------------|-----------------|---------|-----------|----------------|--------|
| <input checked="" type="checkbox"/> | webserver1 | IBM HTTP Server | ihsnode | was85host | Not applicable |        |

Total 1



#### Information

This step is not necessary because the default behavior is to automatically generate a new plug-in configuration file whenever an update is made. However, this step confirms that the setup is working correctly.

- \_\_ c. Verify that the generation was successful by viewing the messages.
- \_\_ 2. View the plug-in configuration file, `plugin-cfg.xml`, from the administrative console. This `plugin-cfg.xml` file is specific to the web server. If you have multiple web servers, it is possible for each `plugin-cfg.xml` file to be unique.
- \_\_ a. Click **webserver1**.
- \_\_ b. Under Additional Properties, click **Plug-in properties**.

- \_\_ c. Under Plug-in properties, click **View** to see the `plugin-cfg.xml` file.

**Web servers > webserver1 > Plug-in properties**

Use this page to configure a web server plug-in. The plug-in passes HTTP requests from a web server to WebSphere(R)

**Runtime Configuration**

**Plug-in properties**

Ignore DNS failures during Web server startup

\* Refresh configuration interval  
60 seconds

**Repository copy of Web server plug-in files:**

- \* Plug-in configuration file name  
`plugin-cfg.xml` **View**
- Automatically generate the plug-in configuration file
- Automatically propagate plug-in configuration file

- \_\_ d. The next window shows the plug-in configuration file. Verify that the element

```
<UriGroup Name="default_host_server1_was85hostNode01_Cluster_URIs">
 includes the element
```

```
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
 Name="/snoop/*" />
```

This element ensures that the plug-in recognizes URLs containing `/snoop` and that they get forwarded to the application server.

```
<UriGroup Name="default_host_server1_was85hostNode01_Cluster_URIs">
 <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/ivt/*"/>
 <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/snoop/*"/>
 <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/hello"/>
 <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/hitcount"/>
 <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="*.jsp"/>
 <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="*.jsv"/>
 <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="*.jsw"/>
 <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/j_security_check"/>
 <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/ibm_security_logout"/>
 <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/servlet/*"/>
 <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/PlantsByWebSphere/*"/>
</UriGroup>
```

- \_\_ e. Look through the list of URIs for the `/PlantsByWebSphere` entry.

- \_\_\_ 3. Propagate the plug-in configuration file. After a plug-in configuration file is regenerated, it must be propagated to the web server. You can propagate manually by copying the file from the application server to the web server, or you can do it from the administrative console.



### Information

The default is to automatically generate the plug-in configuration file and to propagate the plug-in to the web server (if the propagation is configured). The administrative console can be used to view these settings. From the navigation tree, click **Servers > Server Types > Web servers > webserver1**. Under Additional Properties, click **Plug-in properties**.

- \_\_\_ a. Click **Servers > Server Types > Web servers**.  
 \_\_\_ b. Select the web server and click **Propagate Plug-in**.

#### Web servers

Use this page to view a list of the installed web servers.

##### Preferences

Select	Name	Web server Type	Node	Host Name	Version	Status
<input checked="" type="checkbox"/>	<a href="#">webserver1</a>	IBM HTTP Server	ihsnode	was85host	Not applicable	

Total 1

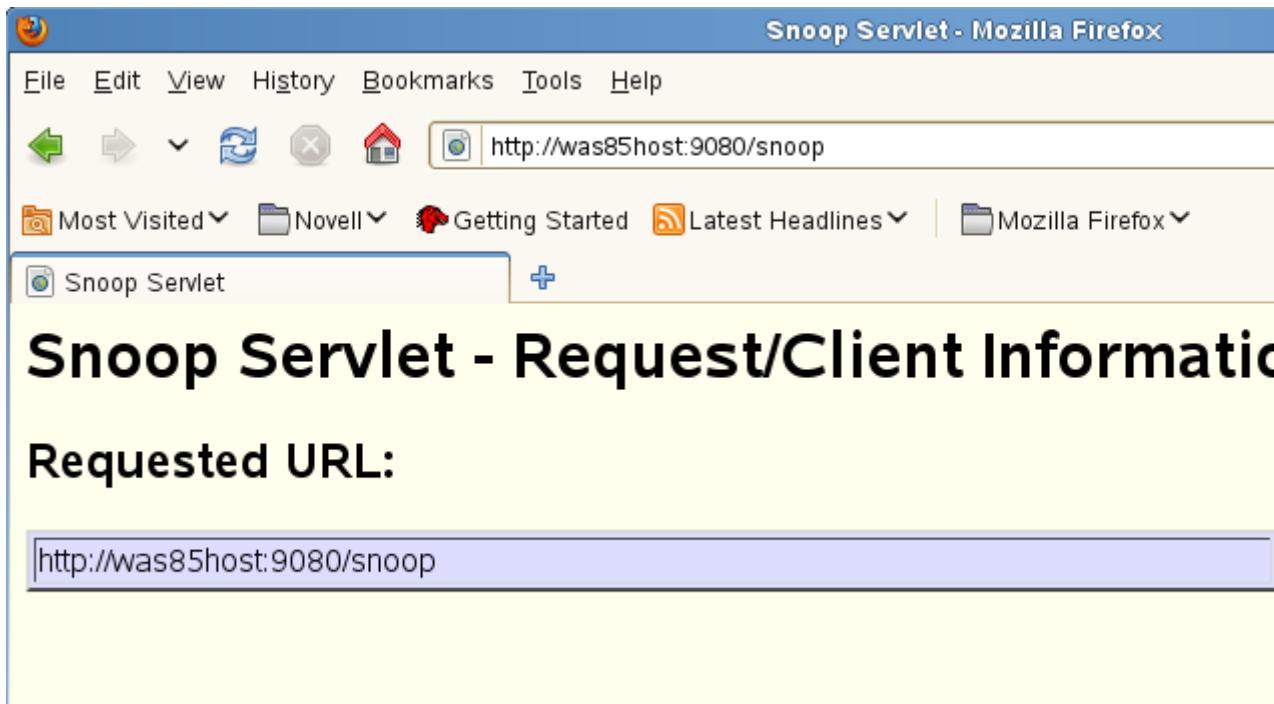
- \_\_\_ c. Verify that the propagation was successful by viewing the messages.

## Section 10: Test the plug-in configuration

By default, the web server plug-in module checks for a new configuration file every 60 seconds. You can wait for the plug-in to find the changes, or you can restart the web server to pick up the changes immediately.

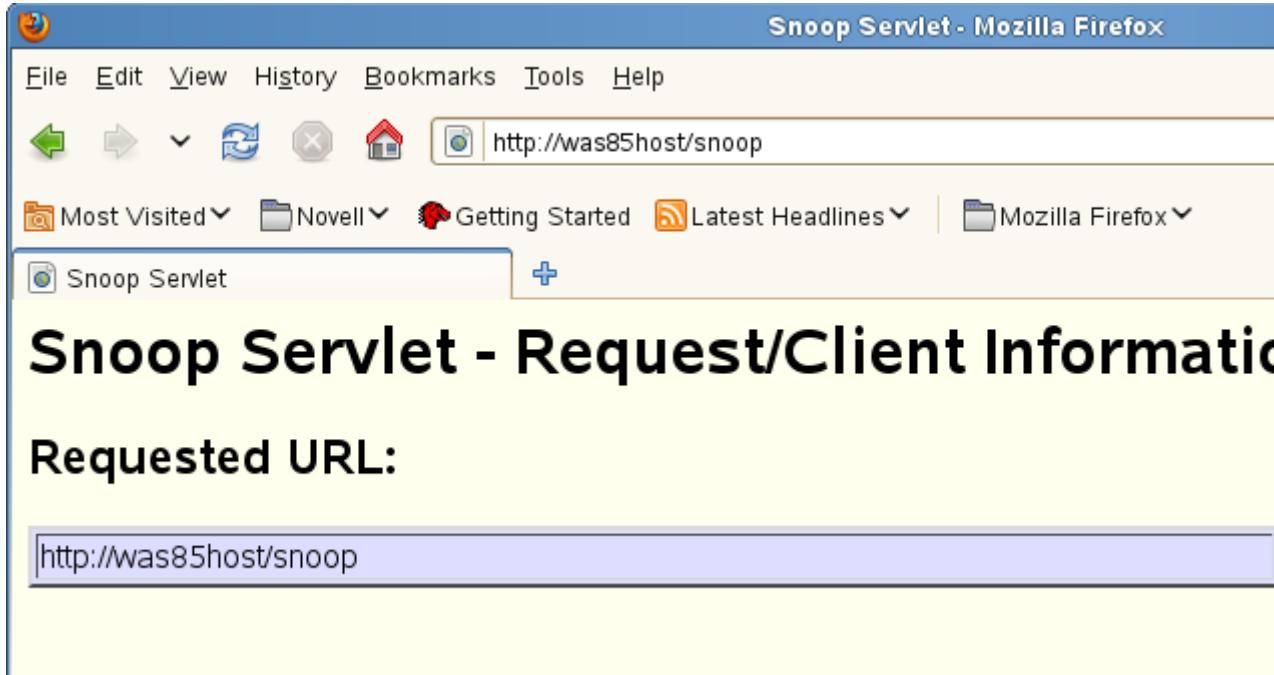
- \_\_\_ 1. Verify that the application server, server1, is running.  
 \_\_\_ a. Click **Servers > Server Types > WebSphere application servers**. If the server is not running, start server1.  
 \_\_\_ b. Minimize the administrative console.  
 \_\_\_ 2. Access the Snoop servlet.  
 \_\_\_ a. Open a new web browser and enter the following address:  
 http://was85host:9080/snoop

By using the port 9080, you are bypassing the external IBM HTTP Server.



- \_\_\_ b. The details for the Snoop servlet are visible in the browser window.
- \_\_\_ 3. Verify that the web server is forwarding requests to the application server.
- \_\_\_ a. Using a browser, enter the following address:

http://was85host/snoop



- \_\_\_ b. The details are visible in the Snoop servlet. This request first goes to the external web server.

## **End of exercise**

## Exercise review and wrap-up

In this lab exercise, you experience the process of creating a WebSphere cell through the generation of a deployment manager profile, and then the federation of application server profiles.

# Exercise 10.Clustering and workload management

## What this exercise is about

This exercise covers the creation of a cluster. While creating the cluster, two cluster members are added. After the cluster is created, the PlantsByWebSphere application is configured to run in the cluster.

A replication domain is set up to use the memory-to-memory replication mechanism, and the application is tested to ensure that session failover works as expected. Testing is achieved by stopping one of the two servers in the cluster and watching the requests fail over to the remaining running server.

## What you should be able to do

At the end of the lab, you should be able to:

- Create a cluster and add cluster members
- Map modules to clusters and web servers
- Test load balancing and failover between two cluster members
- Configure a data replication domain for session management

## Introduction

Up to this point you worked with WebSphere Application Server in a single server environment. In this lab, after federating, you work with a cell and use the deployment manager. You create a cluster so that the workload can be managed between two servers, one on each node you already have.

You also set up a memory-to-memory replication domain so that HTTP sessions can be shared in case of failure of one of the servers.

## Requirements

The lab requires that you successfully completed the previous lab on federation.

# Exercise instructions

## Preface

To do this exercise, you must complete the Federating a cell exercise as it sets up the environment of the nodes, node agents, and servers that are clustered in this exercise.



### Important

The labs use two variables to define various installation paths. On Linux, the variable definitions are as follows:

```
<was_root>: /opt/IBM/WebSphere/AppServer
<profile_root>: /opt/IBM/WebSphere/AppServer/profiles
```

## Section 1: Resetting the WebSphere environment



### Note

To reset your WebSphere environment, read **Appendix A** for instructions on how to complete this procedure.

## Section 2: Check nodes and node agents

Before you can begin creating the cluster, make sure that both node agents are running and the nodes are synchronized.

- \_\_\_ 1. Log in to the deployment manager's administrative console.
  - \_\_\_ a. Log in with `wasadmin` and `web1sphere` as the user name and password.
- \_\_\_ 2. Make sure that both federated nodes, `was85hostNode01` and `was85hostNode02`, are started and synchronized.
  - \_\_\_ a. Click **System Administration > Node agents**.
  - \_\_\_ b. Click **System Administration > Nodes**.



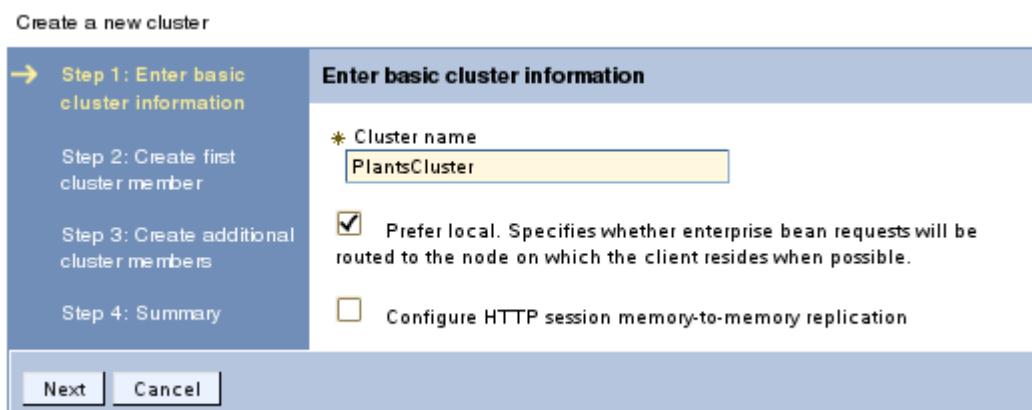
### Information

If the node agents are not started, use the `startNode.sh` script from a terminal window to start them. Make sure that you are in the correct `bin` folder for the profile you are trying to start.

## Section 3: Creating the PlantsCluster cluster

A cluster is composed of two or more servers in a cell, which are assigned to run the same applications. Clusters are logical abstractions that are equivalent to servers. In this section, you create the cluster that contains the cluster members that participate in workload management of the Plants application. You create a cluster that is called PlantsCluster. This cluster is created based on the existing server1 application server. This action means that all of the applications that are already deployed to server1 are included in the cluster.

- 1. Create the PlantsCluster.
  - a. Click **Servers > Clusters > WebSphere application server clusters**.
  - b. Click **New**.
  - c. Enter the following basic cluster information:
    - Enter PlantsCluster for the **Cluster name**.
    - Select the **Prefer local** option.
    - Clear the **Configure HTTP session memory-to-memory replication** option.



Click **Next**.

- \_\_\_ d. Under Select basis for first cluster member, click **Create the member by converting an existing application server**, and from the list, select the existing **server1** server.

\* Member name

Select node

\* Weight  
 (0..100)

Generate unique HTTP ports

Select how the server resources are promoted in the cluster.

**Select basis for first cluster member:**

- Create the member using an application server template.
- Create the member using an existing application server as a template.
- Create the member by converting an existing application server.
- None. Create an empty cluster.

Click **Next**.

- \_\_\_ 2. Add a server, **server2**, to the cluster. This server is created in node **was85hostNode02**. More cluster members can be created either during or after the cluster creation process.
- \_\_\_ a. Enter **server2** for the **Member name**. This name becomes the name of a new server that is created.
- \_\_\_ b. Select **was85hostNode02** from the list for the node name. This node was created in the previous lab by using a custom profile.

- \_\_ c. Make sure that **Generate unique HTTP ports** is selected.

**Create additional cluster members**

Enter information about this new cluster member, and click Add Member to add this cluster member to the member list. A server configuration template is created from the first member, and stored as part of the cluster data. Additional cluster members are copied from this template.

\* Member name

Select node

\* Weight  
 (0..100)

Generate unique HTTP ports

**Add Member**

- \_\_ d. Click **Add Member**.



### Information

Notice that the first server of the cluster is already listed at the bottom of the page. As new servers are added to the cluster, they are also listed here.

- \_\_\_ e. Notice that the new server now shows at the bottom of the page. More cluster members can be created now or after cluster creation.

**Create additional cluster members**

Enter information about this new cluster member, and click Add Member to add this cluster member to the member list. A server configuration template is created from the first member, and stored as part of the cluster data. Additional cluster members are copied from this template.

\* Member name

Select node

\* Weight  
 (0..100)

Generate unique HTTP ports

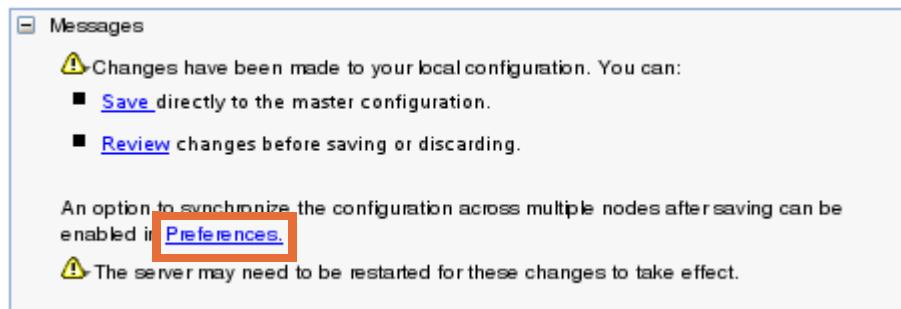
**Add Member**

Use the Edit function to modify the properties of a cluster member in this list. Use the Delete function to remove a cluster member from this list. You are not allowed to edit or remove the first cluster member.

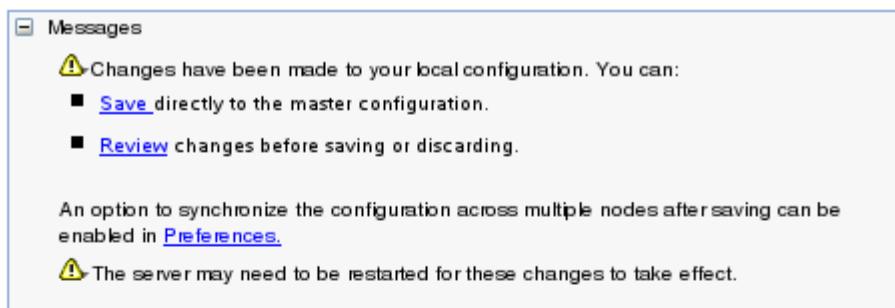
	<b>Edit</b>	<b>Delete</b>		
Select	Member name	Nodes	Version	Weight
<input type="checkbox"/>	server1	was85hostNode01	ND 8.5.5.0	2
<input type="checkbox"/>	server2	was85hostNode02	ND 8.5.5.0	2
Total 2				

- \_\_\_ f. Click **Next**, and then click **Finish** on the Summary page.

- g. Before saving the changes, if not already done, set the console preferences to synchronize configuration changes with the nodes when saving. Click the **Preferences** link.



- h. On the Preferences page, select **Synchronize changes with Nodes**.



#### [WebSphere application server clusters](#) > Console preferences

Specify user preferences for the administrative console workspace.

- Turn on workspace automatic refresh
- No confirmation on workspace discard
- Use default scope
- Show the help portlet
- Enable command assistance notifications
- Log command assistance commands
- Synchronize changes with Nodes

#### [Bidirectional support options](#)

[Apply](#)

[Reset](#)

- i. Click **Apply**, and then click **Save** to save and synchronize with the nodes.

**Information**

From now on, any saves are automatically synchronized with the nodes during a save. Preferences settings are persistent and are retained throughout browser invocations.

- \_\_\_ j. Click **OK** on the Synchronize changes with Nodes page.
- \_\_\_ 3. Modify the default\_host virtual host configuration. This action allows browsers to have direct access to server2 without being forced to use the external IBM HTTP Server.
  - \_\_\_ a. View the HTTP Transport for server2. Click **Servers > Server Types > WebSphere application servers > server2**.
  - \_\_\_ b. Expand **Ports** under Communications.

**Communications****Ports**

Port Name	Port	Details
BOOTSTRAP_ADDRESS	9811	
SOAP_CONNECTOR_ADDRESS	8882	
ORB_LISTENER_ADDRESS	9103	
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	9407	
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	9408	
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	9409	
WC_adminhost	9062	
WC_defaulthost	9081	
DCS_UNICAST_ADDRESS	9356	
WC_adminhost_secure	9045	
WC_defaulthost_secure	9444	
SIP_DEFAULTHOST	5062	
SIP_DEFAULTHOST_SECURE	5063	
SIB_ENDPOINT_ADDRESS	7278	
SIB_ENDPOINT_SECURE_ADDRESS	7287	
SIB_MQ_ENDPOINT_ADDRESS	5559	
SIB_MQ_ENDPOINT_SECURE_ADDRESS	5579	
IPC_CONNECTOR_ADDRESS	9634	

The ports for server2 are listed. The **WC\_defaulthost** for server2 is **9081**. You must add this port number to the host aliases list for the default\_host.

- \_\_\_ c. Click **Environment > Virtual Hosts > default\_host**.

- \_\_\_ d. Click **Host Aliases** under Additional Properties. The host aliases for default\_host are listed. Look for the port 9081 for server2.

[Virtual Hosts > default\\_host > Host Aliases](#)

Use this page to edit, create, or delete a domain name system (DNS) alias by which the virtual host is known.

[+ Preferences](#)

New...	Delete	
Select	Host Name	Port
You can administer the following resources:		
<input type="checkbox"/>	:	9080
<input type="checkbox"/>	:	80
<input type="checkbox"/>	:	9443
<input type="checkbox"/>	:	5080
<input type="checkbox"/>	:	5081
<input type="checkbox"/>	:	443
<input type="checkbox"/>	:	9081
<input type="checkbox"/>	:	9444
Total 8		



**Note**

If 9081 is not already defined, add it by clicking **New**. Leave the default \* for the **host name** and specify 9081 for **Port**. Click **OK** and **Save** the changes.

- \_\_\_ 4. Verify that the new cluster is added to the server configuration and start the cluster.
- \_\_\_ a. Click **Servers > Clusters > WebSphere application server clusters**. The PlantsCluster cluster is shown on the page.
- \_\_\_ b. Select **PlantsCluster** and click **Start** to start the servers on the cluster. Starting the cluster has the same result as starting all the application servers that are cluster members.



## Information

Ripplestart is used when you want to restart a cluster without having all members of the cluster stopped at the same time. This function restarts the individual cluster members one at a time, ensuring that the other members are available to handle requests.

- \_\_\_ c. Make sure that both application servers in the cluster are started. This operation can take a few minutes. You can look at the **Servers > Server Types > WebSphere application servers** page to see the status of the cluster members.

## **Section 4: Set the applications to run on the cluster**

Now that the cluster is defined, the next step is to configure the applications to run on the cluster, rather than on individual servers. Since the web server is used to workload manage the web containers, the web server also must be mapped to the applications. This step is important as it allows the customized `plugin.cfg.xml` files to include the appropriate URIs for each of the applications they are supposed to host.

- \_\_\_ 1. For the PlantsByWebSphere application, verify the next series of steps to map the modules to the PlantsCluster cluster and the webserver1 web server.
  - \_\_\_ a. Click **Applications > Application Types > WebSphere enterprise applications > PlantsByWebSphere**.
  - \_\_\_ b. Under Modules, click **Manage Modules**. The server area indicates to which server or servers the modules are mapped.

[Enterprise Applications > PlantsByWebSphere > Manage Modules](#)

### Manage Modules

Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (`plugin-cfg.xml`) for each Web server is generated, based on the applications that are routed through.

#### Clusters and servers:

WebSphere:cell=was85hostCell01,cluster=PlantsCluster	<input type="button" value="^"/>			
WebSphere:cell=was85hostCell01,node=ihsnode,server=webserver1	<input type="button" value="v"/>			
<input type="button" value="Apply"/>				
<input type="button" value="Remove"/> <input type="button" value="Update"/> <input type="button" value="Remove File"/> <input type="button" value="Export File"/>				
<input checked="" type="checkbox"/> <input type="checkbox"/>				
Select	Module	URI	Module Type	Server
<input type="checkbox"/>	<a href="#">PlantsByWebSphere</a>	PlantsByWebSphereWeb.war,WEB-INF/web.xml	Web Module	WebSphere:cell=was85hostCell01,node=ihsnode,server=webserver1 WebSphere:cell=was85hostCell01,cluster=PlantsCluster



## Information

WebSphere automatically mapped the application to the cluster and web server since there are no other reasonable choices. The steps to map an application are provided to show you the steps to map an application to a server or servers. For example, to map the application to the PlantsCluster, complete the following steps:

- 1) Select the module of the application. Then, in the Clusters and servers list, select both the **PlantsCluster** cluster and the **webserver1** web server (use the **Ctrl** key to select multiple servers).
- 2) Click **Apply**. This action creates the mapping.
- 3) Click **OK**.
- 4) Make sure that the modules are mapped to both the **PlantsCluster** and the web server.
- 5) **Save** the configuration changes.
- 6) **Start** the application if necessary.

- 2. Regenerate and propagate the web server plug-in configuration file.

Although this process happens automatically, do it manually so that you can see the status results and verify that the propagation succeeded.

- a. Click **Servers > Server Types > Web servers**.
- b. Select the web server and click **Generate Plug-in**.
- c. Select the web server and click **Propagate Plug-in**.

## **Section 5: Create a cluster scoped JDBC resource**

When creating the first cluster member from the existing server1, all resources that are already defined at the server and node scope are maintained. Unfortunately, when adding the second server on the was85hostNode02 node, the resource definitions from server1 and was85hostNode01 are not automatically defined. You now have a problem: since both servers run the same applications, by virtue of being on the same cluster, they both need access to the same resources.

One solution is to re-create the resources at the node scope for each additional node as its servers are added to the cluster. That solution works, but the disadvantage is that you must do it every time a new node server is added to the cluster. A better solution is to define resources at the cluster scope.



## Information

Resources can be added at the cluster scope only if the cluster members are running in similar operating environments. Since many resources require pointers to a file system location, it does not work to define resources at the cluster scope for cluster members that run in both Windows and Linux. In that case, you must define the resources at the node level.

- 1. Remove the existing node scoped DB2 Universal JDBC Driver Provider (XA) provider.
  - a. Click **Resources > JDBC > JDBC providers**.
  - b. From the Scope list, select **Node=was85hostNode01**.
  - c. Select the **DB2 Universal JDBC Driver Provider (XA)** provider that is defined at the Node=was85hostNode01 scope, and click **Delete**.

### JDBC providers

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=was85hostCell01, Node=was85hostNode01

Show scope selection drop-down list with the all scopes option

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

### Preferences

New...	Delete			
Select	Name ▾	Scope ▾	Description ▾	
You can administer the following resources:				
<input checked="" type="checkbox"/>	<a href="#">DB2 Universal JDBC Driver Provider (XA)</a>	Node=was85hostNode01		Two-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on the application server for z/OS is not supported for data sources created under this provider.
Total 1				



## Information

Cluster scope takes precedence over node scope, so node scoped resources do not have to be deleted. However, deleting them does avoid ambiguity.

Deleting the JDBC provider also deletes any data sources that are defined under it.

- \_\_ d. **Save** your changes.
  - \_\_ e. Click **OK** on the dialog to confirm that the nodes are synchronized.
2. Create a cluster-scoped JDBC provider and data source.
- \_\_ a. Click **Resources > JDBC > Data sources**.
  - \_\_ b. From the Scope list, select **Cluster=PlantsCluster**.
  - \_\_ c. Click **New**.
  - \_\_ d. Enter **Plants** for the data source name and **jdbc/PlantsByWebSphereDataSource** for the JNDI name.

**Enter basic data source information**

Set the basic configuration values of a datasource for association with your JDBC provider. A datasource supplies the physical connections between the application server and the database.

Requirement: Use the Datasources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.

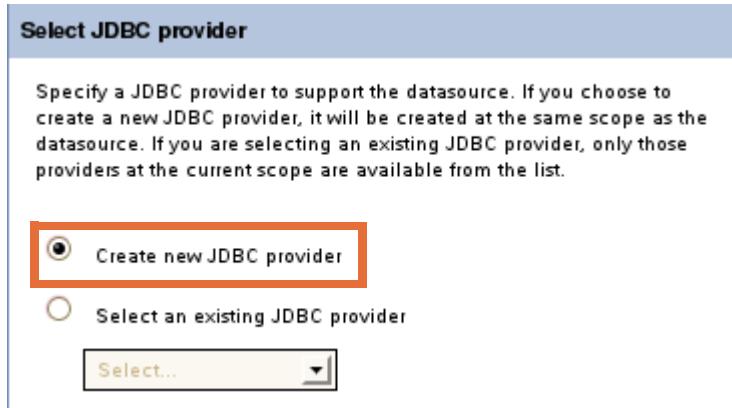
**Scope**

**\* Data source name**

**\* JNDI name**

- \_\_ e. Click **Next**.

- \_\_ f. Since you do not yet have a cluster-scoped JDBC provider, select **Create new JDBC provider**.



- \_\_ g. Click **Next**.



**Note**

The Create Data source wizard temporarily goes to the Create JDBC provider wizard.

\_\_ h. Enter the following selections:

- Select **DB2** for the Database type.
- Select **DB2 Universal JDBC Driver Provider** for the Provider type.
- Select **XA data source** for the Implementation type.

**Create new JDBC provider**

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

**Scope**  
cells:was85hostCell01 :clusters:PlantsCluster

\* **Database type**  
DB2

\* **Provider type**  
DB2 Universal JDBC Driver Provider

\* **Implementation type**  
XA data source

\* **Name**  
DB2 Universal JDBC Driver Provider (XA)

**Description**  
Two-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on the application server for z/OS is not supported for data sources created under this provider.

Click **Next**.

\_\_ i. On the next step, enter the database class path information in both directory location fields. On Linux, this path is /opt/ibm/db2/V9.7/java. Click **Next**.

- \_\_ j. Enter the following database-specific properties for the data source:

- Driver type: 4
- Database name: PLANTS
- Server name: dbhost
- Port number: 50001

**Enter database specific properties for the data source**

Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* Driver type	<input type="text" value="4"/>
* Database name	<input type="text" value="PLANTS"/>
* Server name	<input type="text" value="dbhost"/>
* Port number	<input type="text" value="50001"/>

Use this data source in container managed persistence (CMP)

Click **Next**.

- \_\_ k. From the Component-managed authentication alias list, select **was85hostNode01/PlantsApp**. Click **Next**.
- \_\_ l. On the Summary page, click **Finish**.
- \_\_ m. **Save** your changes. Click **OK**.
- \_\_ 3. Test the data source connection.
- \_\_ a. If you are not placed on the Data sources page, click **Resources > JDBC > Data sources**.
- \_\_ b. Make sure the **Cluster=PlantsCluster** scope is selected.
- \_\_ c. Select the **Plants** data source and click **Test connection**. Examine the messages that are generated to make sure that both node agents were able to connect.

**Messages**

**I** The test connection operation for data source Plants on server nodeagent at node was85hostNode01 was successful.  
**I** The test connection operation for data source Plants on server nodeagent at node was85hostNode02 was successful.

**Note**

For the test to succeed, DB2 and the node agents must be running.

This exercise also assumes that the dbhost definition is added to your hosts file. This procedure is done in a previous exercise. If not, add dbhost as an alias for your host.

- 4. Disable the HTTPOnly setting.
  - a. Click **Security > Global security**.
  - b. Under Authentication, expand **Web and SIP Security** and click **Single sign-on (SSO)**.
  - c. Clear the box for **Set security cookies to HTTPOnly to help prevent cross-site scripting attacks**.

[\*\*Global security\*\*](#) > **Single sign-on (SSO)**

Specifies the configuration values for single sign-on.

**General Properties**

Enabled

Requires SSL

**Domain name**

Interoperability mode

LTPA V1 cookie name

LTPA V2 cookie name

Web inbound security attribute propagation

Set security cookies to HTTPOnly to help prevent cross-site scripting attacks

**Apply**

**OK**

**Reset**

**Cancel**

- d. Click **OK** and **Save** the changes.

**Note**

This setting is enabled by default to help limit the ability of JavaScript to access your cookies. Although this setting is beneficial from a security perspective, it prevents you from seeing the JSESSIONID cookie, which is interesting to see later in this exercise.

This setting is **not** something that you would typically want to disable in your production environment.

## Section 6: Test the application

In this section of the exercise the application is tested in a clustered environment. The application is served from both application servers (cluster members) until the application creates an HTTP session object. At that point, affinity is established. This condition means that from that point on, all requests are directed to the same application server. This action is done so that the user's session information is available locally.

If the cluster member that is holding the user session becomes unavailable, the web server plug-in reroutes the request to another cluster member. However, this situation presents a problem because the new application server does not (by default) have access to the session information.

The exercise initially demonstrates this “problem,” but then later configures a solution that allows the cluster members to share their session information. As a result, even if a cluster member fails, users are still able to access their session through another cluster member.

- 1. Restart your environment.
  - a. Using the administrative console, stop all the application servers (**Servers > Server Types > WebSphere application servers**).
  - b. Using the administrative console, stop all the node agents (**System administration > Node agents**).
  - c. Using the administrative console, stop the deployment manager (**System administration > Deployment manager**).
  - d. Use the `startManager` and `startNode` scripts in the `bin` directories to start your cell. Or use the following script, which starts the deployment manager and both node agents:  
`/usr/software/scripts/start_cell.sh`
  - e. Start the PlantsCluster (**Servers > Clusters > WebSphere application server clusters**).
  - f. Verify that the web server is running (**Servers > Server Types > Web servers**).
  - g. Log out of the administrative console and close the browser window.

- \_\_\_ 2. Access the PlantsByWebSphere application.
  - \_\_\_ a. Open a new browser window and access the PlantsByWebSphere application by entering the following URL:

<http://was85host/PlantsByWebSphere>



### Important

It is important to note that the URL does not include a port number. Consequently, the request is going to go through the external web server. This condition is important because it is the WebSphere plug-in that is running within the web server process that has the intelligence to route the requests to the various cluster members.

- \_\_\_ b. Click **Flowers** in the upper left.
- \_\_\_ c. Click any of the available flowers, and then click **Add to cart**.
- \_\_\_ d. This action takes you to the shopping cart. Notice the flower that you selected is listed.
- \_\_\_ e. Click the **Trees** tab in the upper left, select a tree entry, and add it to your cart. Notice that both your flower and your tree are in your shopping cart.



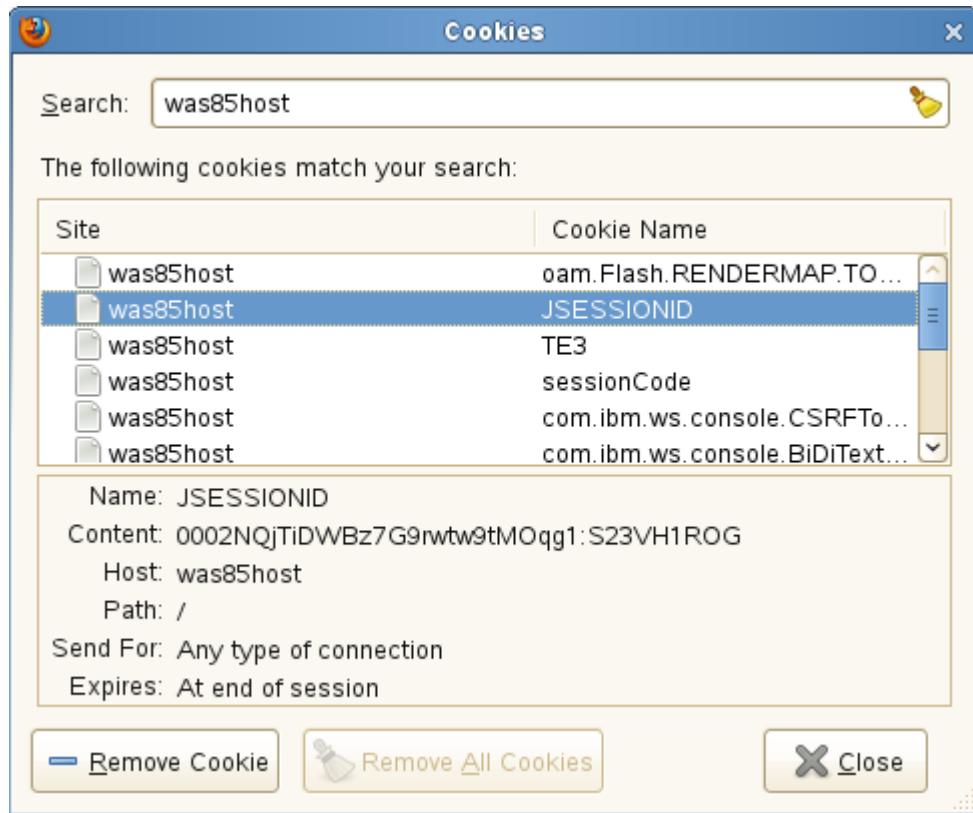
### Information

Based on the content of the JSESSIONID cookie, the web server plug-in knows which server is hosting your information. Given that affinity, the plug-in makes sure to route all of your requests to the same server (or cluster member).

This feature allows the server to store what is in your shopping cart and make it available as you continue to shop.

- \_\_\_ 3. Look for the session cookie.
  - \_\_\_ a. To view the cookies for the application, click **Tools > Page Info > Security > View Cookies** in the browser tools.

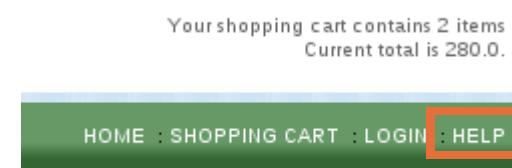
- \_\_ b. Click the **JSESSIONID** cookie in the list to see its contents.



### Information

The value of the JSESSIONID cookie contains various pieces of information, including the cluster member (called the CloneID) for which you have affinity. In some cases, the CloneID is obscured because it actually contains a list of CloneIDs.

- \_\_ c. **Close** the Cookies and Page Info windows.
- \_\_ 4. View the server runtime information. This information is useful to understand the failover testing that is done later in this exercise.
- \_\_ a. To view the runtime server information, click **Help** in the navigation bar at the top of the screen.



- \_\_ b. On the Help page, click the **View Server Info** link near the bottom of the page.

**PLANTS BY WEBSPHERE**

Flowers Fruits & Vegetables Trees Accessories

Home >

## Help

Plants By WebSphere provides limited help support. See the sample docs directory for documentation on the design, building, and installation of the sample.

Debug mode has been tied to the JSF project stage declaration. Debug messages will be displayed when the web app's javax.faces.PROJECT\_STAGE context param is set to either Development or UnitTest. A value of SystemTest or Production will turn off debug output. The current state of debugging is indicated in the check box below.

Debug messages enabled

If the database becomes corrupted for some reason, the button below can be used to delete all data currently in the database and populate it with a fresh set of data. If this does not work, stop the server and repeat the prerequisite steps found in the docs directory to unzip the Derby database.

[Reset database](#)

[View Server Info](#)

[Admin Home](#)

Powered by  
 Flowers : Fruits & Vegetables : Trees : Accessories : Home : Shopping Cart

- \_\_\_ c. On the server information page, notice that the **Process** field shows the server name and that the **Session Data** and **Session Created** fields are **null**.

## PLANTS BY WEBSPHERE

The screenshot shows a web application interface. At the top, there's a green header bar with the title 'PLANTS BY WEBSPHERE'. Below it is a white content area with a green horizontal bar at the top labeled 'Runtime server information'. A table follows, with columns: Cell, Node, Process, Session Data, and Session Created. The data is as follows:

Cell	Node	Process	Session Data	Session Created
was85hostCell01	was85hostNode02	server2	null	null

Below the table are three buttons: 'Session Data' (highlighted in orange), 'Update', and 'Refresh'. Underneath these buttons is a link 'Show cookies'. At the bottom of the page, there's a footer section with the text 'Thu Sep 06 15:12:24 EDT 2012' and a green horizontal bar. The footer also includes the IBM WebSphere logo and links to 'Home', 'Admin Home', and 'Help'.

- \_\_\_ d. Take note of the server name in the Process field: \_\_\_\_\_



### Important

This information is important because this server is the server to which your browser currently has affinity. Thus the session information is stored specifically on that server. If that server fails, your session information can be lost.

- \_\_\_ 5. Continue shopping.

- \_\_\_ a. Click **HOME** in the top navigation bar of the View Server Info page.

- \_\_\_ b. Add several more items to your shopping cart. Notice that the shopping cart continues to include your previously added items.

## Shopping Cart

Here are the items you have selected. To recalculate your total after changing the quantity of an item, select the 'Recalculate' button. To remove an item from your cart, enter "0" as the quantity. Select 'Checkout Now' to begin the checkout process.

ITEM #	ITEM DESCRIPTION	PACKAGING	QUANTITY	PRICE	SUBTOTAL
F0003	Black-eyed Susan	2 plants	<input type="text" value="1"/>	\$9.00	\$9.00
T0004	Crabapple	10 gallon seedling	<input type="text" value="1"/>	\$57.00	\$57.00
A0009	Hand Rake	Assembled	<input type="text" value="1"/>	\$4.50	\$4.50

Order Subtotal:\$70.50

- \_\_\_ c. Return to the View Server Info page (**Help > View Server Info**) and notice that the server name in the **Process** field remains the same. This information does not change because of the affinity with that server.

## PLANTS BY WEBSPHERE

### Runtime server information

Cell	Node	Process	Session Data
was85hostCell01	was85hostNode02	server2	null

Session Data

Thu Sep 06 15:12:24 EDT 2012



[Home](#) : [Admin](#) [Home](#) : [Help](#)

- \_\_\_ 6. Stop the server to which you have affinity.

- \_\_\_ a. Leave your browser window to PlantsByWebSphere open.  
 \_\_\_ b. Maximize the administrative console.

- \_\_\_ c. Click **Servers > Server types > WebSphere application servers**, select the server to which you have affinity, and click **Stop**. Click **OK** if you are prompted to confirm the stop in the **Stop server** panel.



### Information

This action simulates a server failure and forces a failover to the other cluster member.

- \_\_\_ d. Make sure that the server is stopped before returning to the PlantsByWebSphere window. Click the **Status** refresh icon until the Status shows a complete stop.

#### Application servers

Use this page to view a list of the application servers in your environment and the status of each of these servers. You can also use this page to change the status of a specific application server.

Preferences

Application Servers						
		New...	Delete	Templates...	Start	Stop
					Restart	ImmediateStop
					Terminate	
<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>						
Select	Name	Node	Host Name	Version	Cluster Name	Status
You can administer the following resources:						
<input type="checkbox"/>	<a href="#">server1</a>	was85hostNode01	was85host	ND 8.5.0.0	PlantsCluster	
<input type="checkbox"/>	<a href="#">server2</a>	was85hostNode02	was85host	ND 8.5.0.0	PlantsCluster	
Total 2						

- \_\_\_ e. Minimize the administrative console.  
 \_\_\_ f. Return to the PlantsByWebSphere browser window and click **Home**.

- \_\_\_ g. Click **Shopping Cart**. Notice that the shopping cart is empty.

## PLANTS BY WEBSPHERE

ITEM #	ITEM DESCRIPTION	PACKAGING	QUANTITY	PRICE	SUBTOTAL
<b>Order Subtotal:\$0.00</b>					

- \_\_\_ h. Close the PlantsByWebSphere browser window.



### Information

The PlantsByWebSphere application is not coded to store the shopping cart in the session information. Therefore, a server failure causes the loss of the shopping cart contents.

If the failover of your session information is important, it is necessary to design your application with session failover in mind. There are a number of different possible designs, but that discussion is outside of the scope of this course.

One possible approach is to store the contents of the shopping cart in the HTTP session object. This approach is demonstrated in the next part of this exercise.

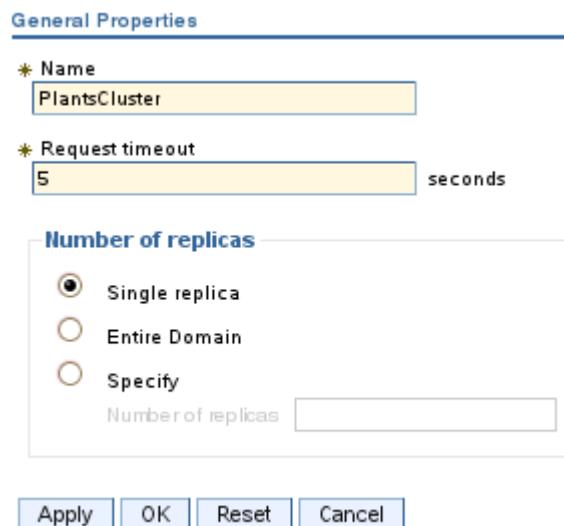
## Section 7: Configure session replication settings

In order for members of a cluster to share session information, a strategy to share session data must be put in place. WebSphere Application Server provides various mechanisms to achieve this goal. The main strategies are database and memory-to-memory replication. Setting up either of these mechanisms is straightforward. In this exercise, memory-to-memory replication is set up to handle session data replication.

Session management can be configured on each of the servers in the cluster. This action can be completed when you create the cluster or at a later point.

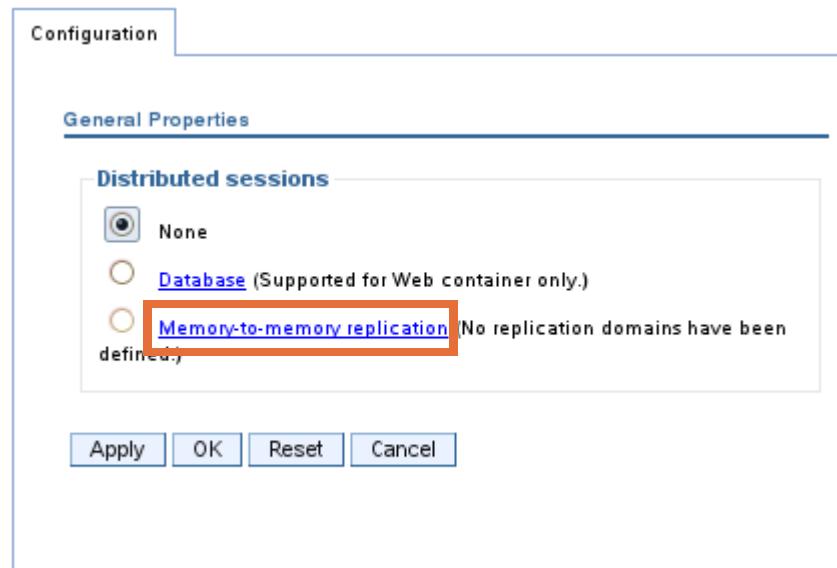
In this section, you configure session replication settings.

- 1. Stop the remaining server in the PlantsCluster.
  - a. Click **Servers > Server types > WebSphere application servers**.
  - b. Select the server that is still running, and click **Stop**. Click **OK** if you are prompted to confirm the stop in the **Stop server** panel.
- 2. Configure a replication domain.
  - a. Click **Environment > Replication domains**.
  - b. Click **New** to create a Replication domain.
  - c. Enter `PlantsCluster` for the Name. Keep all remaining defaults.

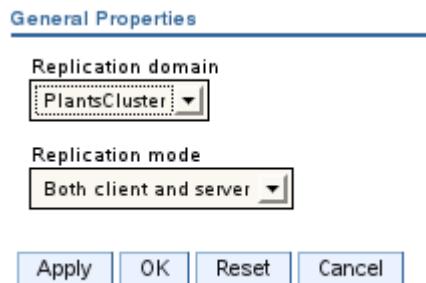


- d. Click **OK**.
- e. **Save** the changes.
- 3. Configure the memory-to-memory session replication settings for the cluster members.
  - a. Click **Servers > Server Types > WebSphere application servers**.
  - b. Click the hyperlink for either one of the servers.
  - c. Under Container Settings, click **Session management**.
  - d. Under Additional Properties, click **Distributed environment settings**.

- \_\_ e. Click the **Memory-to-memory replication** hyperlink.



- \_\_ f. In the General Properties area, select **PlantsCluster** for the replication domain. Verify that the replication mode is **Both client and server**.



- \_\_ g. Click **OK**.  
 \_\_ h. On the Distributed environment settings page, click **OK**.  
 \_\_ i. On the breadcrumb trail, click the server name.  
 \_\_ j. Under Container Settings, click **EJB Container Settings > EJB container**.

- \_\_ k. Click the **memory-to-memory replication** hyperlink.

**General Properties**

\* Passivation directory  
\$(USER\_INSTALL\_ROOT)/temp

Inactive pool cleanup interval  
30000 milliseconds

Default data source JNDI name  
(none)

Enable stateful session bean failover using [memory-to-memory replication](#)

Apply OK Reset Cancel

- \_\_ l. In the General Properties area, select **PlantsCluster** for the replication domain. Verify that the replication mode is **Both client and server**.

**General Properties**

Replication domain  
PlantsCluster

Replication mode  
Both client and server

Apply OK Reset Cancel

- \_\_ m. Click **OK**.

- \_\_ n. Now that you selected the replication domain, you can see that the box for **Enable stateful session bean failover** is selected. Click **OK**.

**General Properties**

\* Passivation directory  
\$(USER\_INSTALL\_ROOT)/temp

Inactive pool cleanup interval  
30000 milliseconds

Default data source JNDI name  
(none)

Enable stateful session bean failover using [memory-to-memory replication](#)

Apply OK Reset Cancel

- \_\_\_ o. **Save** the changes.
- \_\_\_ p. Repeat the previous steps for the other server.

## Section 8: Test the application for session failover

In this section, you test the failover of the session information. Although the PlantsByWebSphere application was not designed to fail over to the shopping cart, you can store content in the session object. After that is done, you stop the application server that is holding session information to demonstrate that the information does indeed fail over to the other cluster member.

- \_\_\_ 1. Start the cluster.
  - \_\_\_ a. Click **Servers > Clusters > WebSphere application server clusters**.
  - \_\_\_ b. Select the **PlantsCluster** and click **Start**. Wait for all cluster members to start.
  - \_\_\_ c. Log out of the administrative console and close the browser window.
- \_\_\_ 2. Continue shopping in the PlantsByWebSphere application.
  - \_\_\_ a. Open a new browser window and access the PlantsByWebSphere application by entering the following URL:  
`http://was85host/PlantsByWebSphere`
  - \_\_\_ b. Click **Home**.
  - \_\_\_ c. Click **Help** and proceed to the **View Server Info** page.
  - \_\_\_ d. Take note of the server name in the **Process** field: \_\_\_\_\_
  - \_\_\_ e. Click **Home** and browse through the store. Add a couple of items to your shopping cart.
  - \_\_\_ f. Return to the View Server Info page and confirm that the server name is the same.
- \_\_\_ 3. Add some content to the HTTP session object. Now, you have objects in the shopping cart and know to which server you have affinity.
  - \_\_\_ a. From the View Server Info page, enter **your name** into the **Session Data** field and click **Update**.

- \_\_\_ b. Notice that not only does the data show in the Session Data field, but the time that the session object was created is shown in the Session Created field.

## PLANTS BY WEBSPHERE

The screenshot shows a web application interface. At the top, there's a green header bar. Below it, a table titled "Runtime server information" displays the following data:

Cell	Node	Process	Session Data	Session Created
was85hostCell01	was85hostNode01	server1	Lily	Fri Sep 07 14:56:29 EDT 2013

Below the table, there are buttons for "Session Data" (containing "Lily"), "Update", and "Refresh". Further down, there's a button labeled "Show cookies".

- \_\_\_ c. Take note of the time that is shown in the **Session Created** field:  
 \_\_\_\_\_
- \_\_\_ d. Click **HOME** and add several items to your shopping cart.
- \_\_\_ e. Return to the **View Server Info** page (through the **Help** page) and notice that the session data and time for session that are created are not changed.
- \_\_\_ 4. Simulate a server failure.
- \_\_\_ a. Open a terminal window, and **Stop** the server to which you have affinity.  
 Navigate to the `<profile_root>/<profile_name>/bin` directory for the server to which you have affinity.
- \_\_\_ b. Enter the following command to stop the server:  
`./stopServer.sh <server_name> -user wasadmin -password websphere`
- \_\_\_ c. Wait for the server to stop completely.
- \_\_\_ 5. Verify the session data failover.
- \_\_\_ a. Return to the browser window that you used for accessing the PlantsByWebSphere application.
- \_\_\_ b. Click **Home > Shopping Cart**. Notice that the shopping cart is empty. This condition is expected.

- \_\_\_ c. Return to the **View Server Info** page. Notice that the server name changed to the other cluster member. Also, notice that the **Session Data** and **Session Created** fields stayed the same.

## PLANTS BY WEBSPHERE

Runtime server information				
Cell	Node	Process	Session Data	Session Created
was85hostCell01	was85hostNode02	server2	Lily	Fri Sep 07 15:13:40 EDT 2

Session Data



### Information

You can see that the shopping cart for PlantsByWebSphere does not fail over, and the shopping cart is empty after a failure. However, this last section demonstrates the use of an HTTP session object within the code, and the fact that the cluster members can fail that data over between the cluster members.

The other thing to note is that your browser now has affinity to the new server.

## End of exercise

## Exercise review and wrap-up

The first part of the exercise looked at creating a cluster of two servers, each in its own node.

Next, the applications are configured to run on the cluster by assigning the modules of the applications to the web server and the cluster.

Finally, the application was thoroughly tested in the clustered environment, and failover scenarios were created by stopping one of the servers.

To make failover work when session data is involved, the Data Replication Service used memory-to-memory replication.

# Exercise 11.Configuring the service integration bus

## What this exercise is about

This exercise shows you how to set up a WebSphere Application Server environment to support asynchronous communication by using the JMS protocol with the WebSphere default messaging provider. The exercise presents some of the design decisions that must be made to define how the service integration bus (SIBus) is configured. You learn how to configure the required SIBus artifacts and the required JMS artifacts. You explore the created configuration to learn how the artifacts are connected to each other.

You install two applications. One application acts as a client application that uses a web interface to accept requests from clients. The client application puts the requests onto a request queue. The second application acts as a request processing application. The second application retrieves client requests from the queue, and acts on them according to some business logic. You use these applications to explore messaging behavior under different messaging engine policy configurations.

The first part of this exercise uses scripts to configure the required SIBus, JMS, and application artifacts. Using the administrative console takes longer, and provides more opportunities for typing errors. Using scripting is less likely to result in a non-working configuration. Examine the contents of the scripts to understand the configuration steps.

## What you should be able to do

At the end of this exercise, you should be able to:

- Explain some of the design decisions that are required to set up a messaging environment
- Explain how to configure the service integration bus, messaging engines, and bus destinations in WebSphere Application Server
- Explain how to set up basic SIBus security
- Explain how to configure JMS queues, connection factories, and activation specifications for message-driven beans
- Explain how to install and test the messaging features of the two example programs

## Introduction

Many business problems can best be solved by using asynchronous messaging. Typically, a client application accepts requests from clients and puts those requests on a JMS destination. A processing application also exists that retrieves the requests from the destination and completes some business logic that is based on the request.

When you receive applications that use JMS for communication, you must configure the WebSphere environment to support these applications.

The configuration that is required for using the default messaging provider in WebSphere Application Server can be broken down into the following three main parts:

1. Configuring WebSphere Application Server specific SIBus resources
2. Configuring application-specific JMS resources
3. Installing the applications

WebSphere Application Server provides many configuration options that affect the behavior of message processing. In most cases, you use clusters to host the JMS applications, and to host the SIBus resources. The clusters can be configured to provide high availability for the hosted resources. The clusters can also be configured to provide scalability by using multiple instances of the hosted resources. Think carefully about the required behavior of your application when you configure WebSphere Application Server default messaging.

In many cases, you want to set up a secure messaging environment that requires a user ID and password to connect. You might also want to set up user IDs that use different roles to restrict access to the environment. WebSphere Application Server makes it easy to assign users and groups to the different roles necessary to send, receive, and connect to a messaging bus.

In this exercise, you configure both JMS and service integration bus resources to support two applications. The first application creates simulated transactions that represent buying and selling stocks. On a web page in the application, you can select the number of transactions of each type (buy or sell). Each transaction places messages on a JMS queue.

The second application uses an MDB to listen to the same queue and receives the messages that the simulator sends. The transactions are

displayed on a table. Every 30 seconds the oldest transaction is removed from the table.

## Requirements

For this exercise, you must first complete **Exercise 10: Clustering and workload management**. Additionally, this exercise assumes that HPEL is enabled for the application servers. HPEL is enabled in **Exercise 7: Problem determination**.

## Exercise instructions

### Section 1: Resetting the WebSphere environment



#### Note

To reset your WebSphere environment, read **Appendix A** for instructions on how to complete this procedure.

### Section 2: Making the configuration decisions

A typical messaging topology consists of an application that produces messages, and an application that consumes messages. Usually the producer application has a client-facing front end. The client provides some input that specifies the content of the message. The consuming application typically does not have a web interface. The consuming application usually interacts with a back-end service. In this exercise, the consuming application also has a web interface, so you can see a status when messages are consumed.

One of the basic decisions that you must make is whether to install your applications onto a single server, or into a cluster. In most cases, you would want to use clusters, at least for availability reasons. There might also be performance requirements that indicate the need for clustering.

Your first decision is to use clustering to host the two applications. Your second decision, then, is which clusters are used to host the two applications. In this scenario, you use the existing cluster that was created earlier for both applications.

- \_\_\_ 1. Explore the environment.
- \_\_\_ a. Examine the table that shows the application placement decisions that are used for the scenario in this exercise.
- \_\_\_ b. Write the cluster name `PlantsCluster` into the table you are using for your application placement design decisions. You use the same cluster for both applications.

**Table 7: Application placement design decisions**

Question	Decision
Single server or cluster for the client-facing EAR ( <code>MSGSenderSimulator.ear</code> )?	CLUSTER
<code>MSGSenderSimulator.ear</code>	
Cluster name?	
Single server or cluster for the business logic EAR ( <code>TPPProcessorApplication.ear</code> )?	CLUSTER
<code>TPPProcessorApplication.ear</code>	
Cluster name?	

The next basic decision that you must make is whether to host the SIBus messaging engine in a single server or in a cluster. In most cases, you would want to use a cluster, at least for availability reasons. If more than one messaging engine is needed to handle the expected amount of client traffic, you might also decide to use clustering for scalability reasons. In the initial scenario for this lab, you are using the existing cluster in high availability mode.

You must also decide whether the messaging engines use a file-based message store or a database message store. Since you use a cluster to host the messaging engines, the message store must be accessible from all cluster members. The scenario in this lab exercise uses a file-based message store. Since only one computer is used in this lab exercise, the local file system is accessible to all cluster members. In a scenario with multiple computers, a shared file system would be required.

- c. Examine the table that shows the messaging engine placement decisions that are used for the scenario in this exercise.
- d. Write the cluster name `PlantsCluster` into the table you are using to track your messaging engine placement design decisions.
- e. Write the policy type `High Availability` into the table you are using to track your messaging engine placement design decisions.
- f. You use a file store as the message store. Since you are using a high availability policy, the message store location must be visible from all cluster members in the cluster. In this case, since both cluster members are actually running on the same computer, you can use a common directory in the file system. In the typical case where cluster members are running on different computers, you would be required to use a shared file system.

**Table 8: Messaging engine placement design decisions**

Question	Decision
Messaging engine on single server or cluster?	CLUSTER
Messaging engine cluster name	_____
Messaging engine policy type:	_____
High availability or scalability or both?	_____
Database or file system message store?	File store
File system location?	/opt/IBM/WebSphere/AppServer/ MessagingFileStores/<ME_Name>

Another basic decision is whether to use security for the SIBus environment you create. In most cases, you want to use a secure messaging environment. After you decide to enable

bus security, you must determine the roles that the application uses. You must also determine the users and groups that are assigned to those roles.

- \_\_\_ g. Examine the table that shows the security decisions that are made for the scenario that is used in this exercise.

**Table 9: Security decisions for the SIBus environment**

Question	Decision
Use security or no security	SECURITY
Roles the applications need	Bus connector role
Bus connector user name	busUser
Bus connector password	websphere

After making the design decisions, more information must be gathered to configure the environment correctly. The applications require a JMS destination to communicate. The type of destination (queue or topic), and the JNDI name of the destination must be obtained from the application provider. The client-facing application requires a JMS connection factory to put messages on the destination. The business logic application requires a JMS ActivationSpec to get messages from the destination. The JNDI names of the JMS ConnectionFactory and of the JMS ActivationSpec must be obtained from the application provider.

- \_\_\_ h. Examine the table that shows the additional JMS configuration information that is used in this scenario.
- \_\_\_ i. Write the JNDI name `jms/tradeprocq` for the destination JNDI name.
- \_\_\_ j. Write the JNDI name `jms/tradeCF` for the connection factory JNDI name.
- \_\_\_ k. Write the JNDI name `jms/tradeAS` for the activation spec JNDI name.

**Table 10: JMS configuration information**

Type	Value
Destination type	QUEUE
Destination JNDI name	
Connection factory JNDI name	
Activation spec JNDI name	

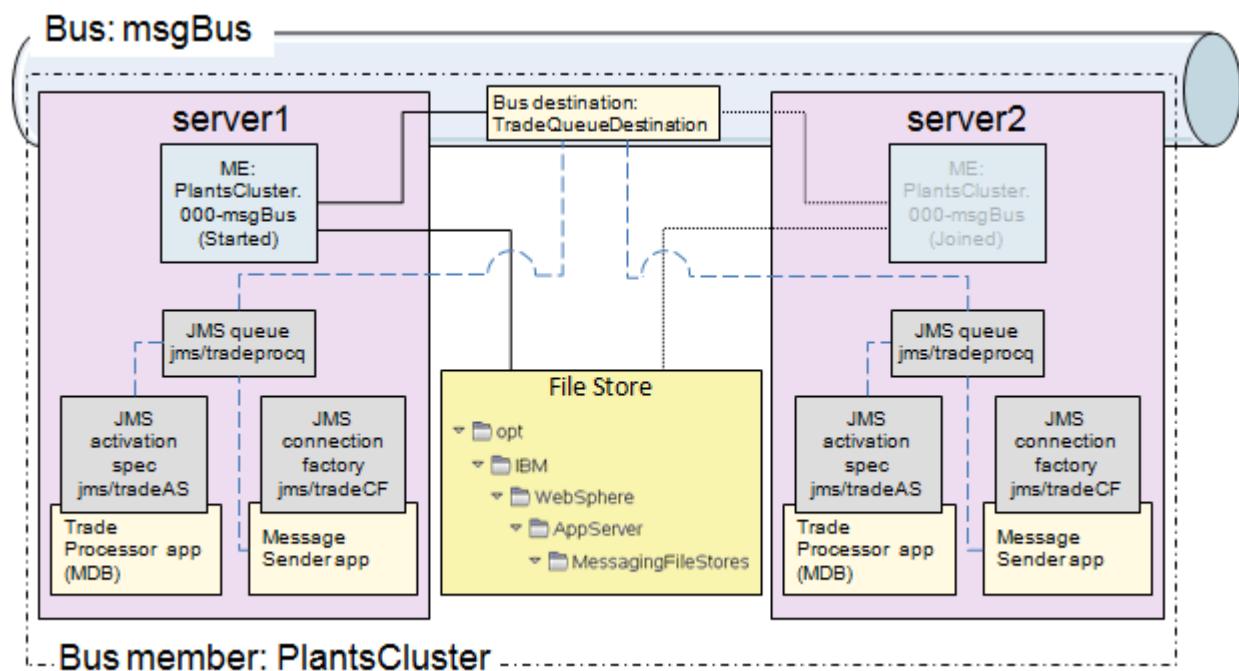
You must also decide on the configuration of the SIBus artifacts. These decisions do not require information from the application provider. You must decide on the name of the bus that is created. You must also decide on the name of the SIBus queue destination that is created.

- \_\_ l. Examine the following table that shows the additional SIBus configuration information that this scenario uses.
- \_\_ m. Write the name `TradeQueueDestination` for the SIBus queue name.

**Table 11:**

Type	Value
SIBus bus name	msgBus
SIBus queue destination name	

The messaging environment that is created in the exercise is shown in the graphic.



You create the SIBus objects and the JMS objects that are required to support the two messaging applications. You complete the following tasks in this exercise:

1. Create and secure an SIBus named: `msgBus`
2. Create a bus user and authentication alias, and add the user to the bus connector role.
3. Add the **PlantsCluster** as a member of the bus. The messaging engine (`PlantsCluster.000-msgBus`) becomes active in one of the cluster members, `server1`, or `server2`.

4. Create the bus destination named: TradeQueueDestination
5. Create the JMS objects that the messaging applications require: **Connection Factory, Destination Queue, and Activation Specification**. These objects are Java objects that are scoped to the cluster and are given JNDI names.
6. Install the messaging applications to the PlantsCluster.
7. The following is a table with all of the design decisions that you made.

**Table 12:**

<b>Question</b>	<b>Decision</b>
Client facing EAR (MSGSenderSimulator.ear) single server or cluster?	CLUSTER
MSGSenderSimulator.ear cluster name?	PlantsCluster
Business logic EAR (TPProcessorApplication.ear) single server or cluster?	CLUSTER
TPProcessorApplication.ear cluster name?	PlantsCluster
Messaging engine on single server or cluster?	CLUSTER
Messaging engine cluster name	PlantsCluster
Messaging engine policy type	High availability (HA)
Database or file system message store?	File store
File system location?	/opt/IBM/WebSphere/AppServer/MessagingFileStores/<ME_Name>
Use security or no security	SECURITY
Roles that applications need	Bus connector role
Bus connector user name	busUser
Bus connector password	web1sphere
Destination type	QUEUE
Destination JNDI name	jms/tradeprocq
Connection factory JNDI name	jms/tradeCF
Activation spec JNDI name	jms/tradeAS
SIBus bus name	msgBus
SIBus queue destination name	TradeQueueDestination

In addition to creating this environment, you configure and test two messaging policies:

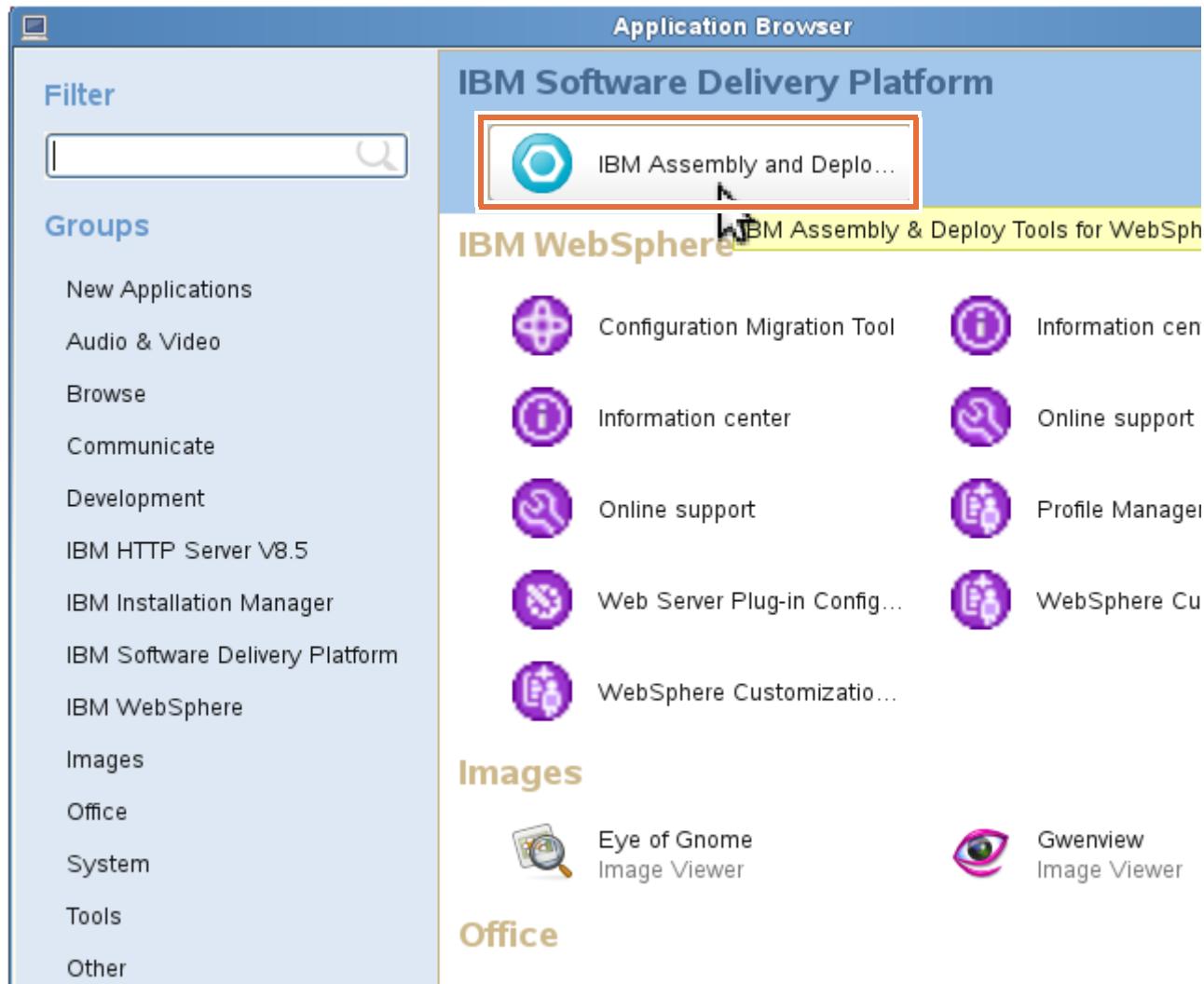
- High availability
- High availability with the option **Always activate MDBs in all servers enabled**

### **Section 3: Examine the scripting environment**

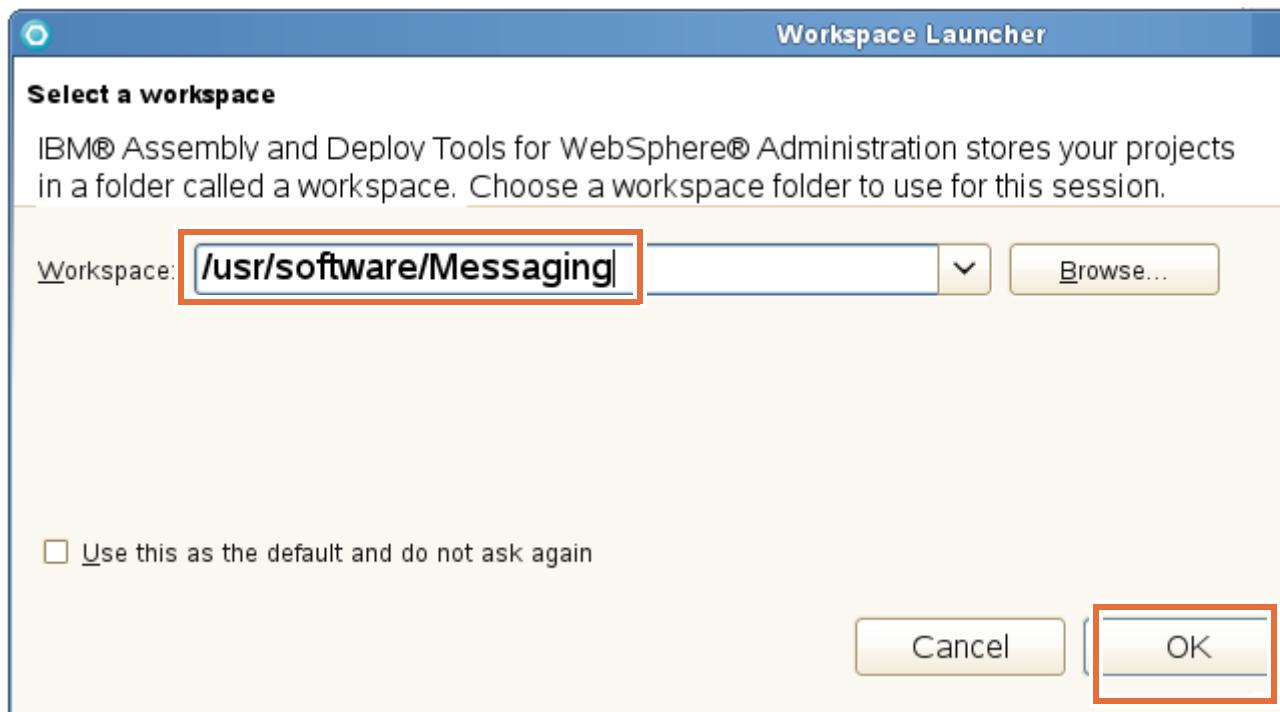
Examine the scripting artifacts that are created to complete the default messaging configuration for this lab exercise. In this exercise, you examine the properties file that

contains the values that the scripts use, the `buildf.xml` file that is used to call the scripts, and the scripts that complete the configuration of the environment.

- \_\_\_ 1. Start IBM Assembly and Deploy Tools for WebSphere Administration.
  - \_\_\_ a. Click **Computer** in the lower left of the screen, then click **More Applications**.
  - \_\_\_ b. Click **IBM Software Delivery Platform** on the left side.
  - \_\_\_ c. Click **IBM Assembly and Deploy**.

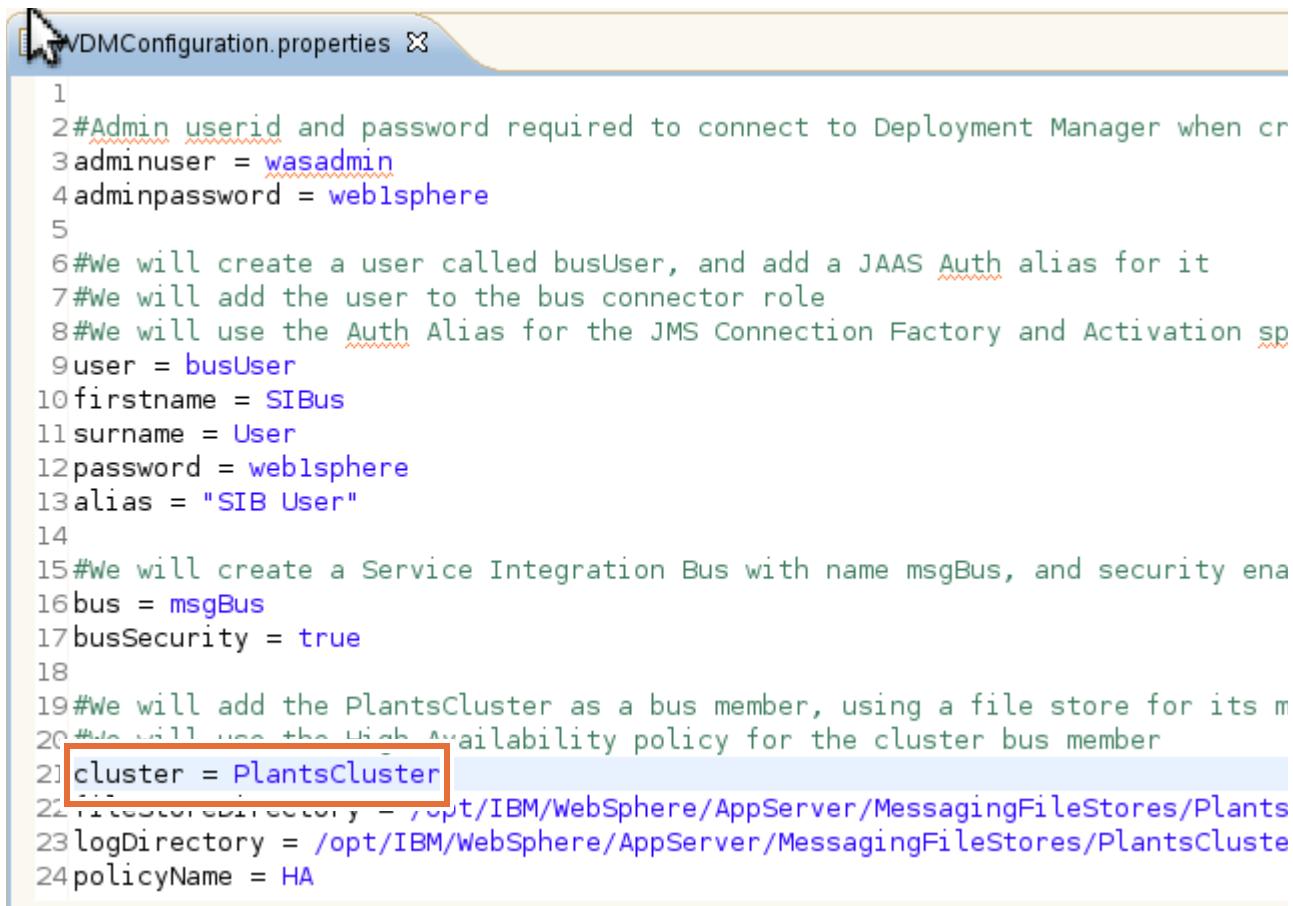


- \_\_ d. On the Workspace Launcher screen enter /usr/software/Messaging for the workspace, and click OK.



- \_\_ e. When the workspace opens, expand the Scripts subdirectory in the Enterprise Explorer.
- \_\_ 2. Examine the properties file.
- \_\_ a. In the left pane, double-click **WDMConfiguration.properties**.
- \_\_ b. When the properties file is displayed in the editor, double-click the tab for the file to expand it to a larger size.

- \_\_\_ c. Notice that the values of the properties correspond to the design and configuration decisions you made earlier.



```

1
2 #Admin userid and password required to connect to Deployment Manager when cr
3 adminuserid = wasadmin
4 adminpassword = websphere
5
6 #We will create a user called busUser, and add a JAAS Auth alias for it
7 #We will add the user to the bus connector role
8 #We will use the Auth Alias for the JMS Connection Factory and Activation sp
9 user = busUser
10 firstname = SIBus
11 surname = User
12 password = websphere
13 alias = "SIB User"
14
15 #We will create a Service Integration Bus with name msgBus, and security ena
16 bus = msgBus
17 busSecurity = true
18
19 #We will add the PlantsCluster as a bus member, using a file store for its m
20 #We will use the High Availability policy for the cluster bus member
21 cluster = PlantsCluster
22 fileStoreDirectory = /opt/IBM/WebSphere/AppServer/MessagingFileStores/Plants
23 logDirectory = /opt/IBM/WebSphere/AppServer/MessagingFileStores/PlantsCluste
24 policyName = HA

```

- \_\_\_ d. If any of the values in the properties file differ from your design decisions, modify the value of the property in the file. In particular, make sure that the cluster property has the same value as the name of the cluster that was created earlier.
- \_\_\_ e. Double-click the tab again to reduce its size.
- \_\_\_ 3. Examine the build.xml file. This file is an Ant build file. It contains one target for each of the operations that are needed to configure the environment. Each target contains a wsadmin task that starts a wsadmin script.
- \_\_\_ a. In the left pane, double-click **build.xml**.
- \_\_\_ b. When the build file is displayed in the editor, double-click the tab for the file to expand it to a larger size.
- \_\_\_ c. Note the various targets that are used to create the configuration. Each target calls a wsadmin script, passing property values as arguments.

- \_\_\_ d. Notice that the property values used by the tasks are obtained from the WDMConfiguration.properties file.

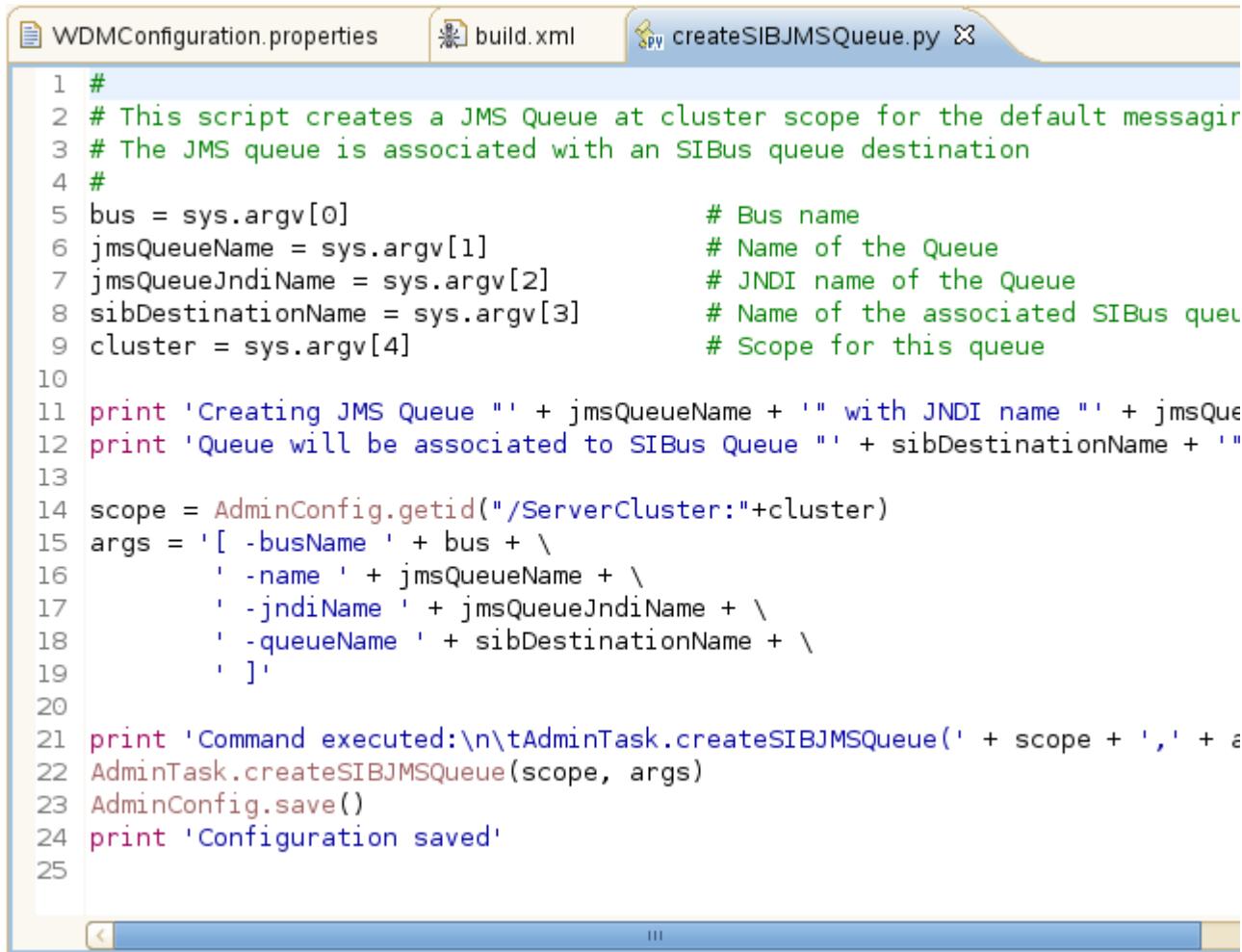
The screenshot shows a code editor with two tabs: "WDMConfiguration.properties" and "build.xml". The "build.xml" tab is active, displaying an Ant script. A red box highlights the line: <property file="/usr/software/Messaging/WDMConfiguration.properties" />. The script defines several targets, including "info" which echoes possible targets like "createUserAndAuthAlias" and "createSIBus". It also defines a target "createUserAndAuthAlias" which uses the wsadmin task with specific arguments for user and password.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project name="Messaging" default="info">
3 <taskdef name="wsadmin" classname="com.ibm.websphere.ant.tasks.WsAdmin"
4
5 <!--
6 All of the targets run scripts that take their arguments from the comma
7 line. The arguments to pass to the scripts are retrieved from a proper
8 file.
9 -->
10 <property file="/usr/software/Messaging/WDMConfiguration.properties" />
11
12 <target name="info">
13 <echo>
14 Possible targets:
15 createUserAndAuthAlias
16 createSIBus
17 addUserToBusConnectorRole
18 addClusterBusMember
19 createSIBQueueDestination
20 createSIBJMSConnectionFactory
21 createSIBJMSQueue
22 createSIBJMSActivationSpec
23 installProducerApp
24 installConsumerApp
25 </echo>
26 </target>
27
28 <!--
29 The createUserAndAuthAlias target is the only script that requires a
30 connection to the deployment manager. We must pass the username and
31 password to the wsadmin command. The script creates a new user in
32 the file based repository, and creates a JAAS authentication alias for
33 the new user. The AuthAlias has the deployment manager's node name
34 prepended to it.
35 -->
36 <target name="createUserAndAuthAlias">
37 <wsadmin conntype="SOAP" lang="jython" failonerror="true"
38 user="${adminuser}" password="${adminpassword}"
39 script="/usr/software/Messaging/createUserAndAuthAlias.py">
40 <arg value="${user}" />
41 <arg value="${password}" />

```

- \_\_\_ e. Double-click the tab again to reduce its size.  
\_\_\_ 4. Examine the scripts that are used.  
\_\_\_ a. In the left pane, double-click any of the .py files.

- \_\_\_ b. When the script file is displayed in the editor, double-click the tab for the file to expand it to a larger size.
- \_\_\_ c. Note the arguments read by the script correspond to the arguments passed in by the build file.
- \_\_\_ d. Examine the contents of the script files. Examine how the SIBus and JMS artifacts are created. Examine how the producer and consumer applications are installed.



```

1 #
2 # This script creates a JMS Queue at cluster scope for the default messaging bus
3 # The JMS queue is associated with an SIBus queue destination
4 #
5 bus = sys.argv[0] # Bus name
6 jmsQueueName = sys.argv[1] # Name of the Queue
7 jmsQueueJndiName = sys.argv[2] # JNDI name of the Queue
8 sibDestinationName = sys.argv[3] # Name of the associated SIBus queue
9 cluster = sys.argv[4] # Scope for this queue
10
11 print 'Creating JMS Queue "' + jmsQueueName + '" with JNDI name "' + jmsQueueJndiName + '"'
12 print 'Queue will be associated to SIBus Queue "' + sibDestinationName + '"'
13
14 scope = AdminConfig.getid("/ServerCluster:" + cluster)
15 args = '[-busName ' + bus + \
16 ' -name ' + jmsQueueName + \
17 ' -jndiName ' + jmsQueueJndiName + \
18 ' -queueName ' + sibDestinationName + \
19 ']'
20
21 print 'Command executed:\n\tAdminTask.createSIBJMSQueue(' + scope + ', ' + args + ')'
22 AdminTask.createSIBJMSQueue(scope, args)
23 AdminConfig.save()
24 print 'Configuration saved'
25

```

- \_\_\_ 5. Exit from the IBM Assembly and Deploy Tool.

## Section 4: Setting up the service integration bus

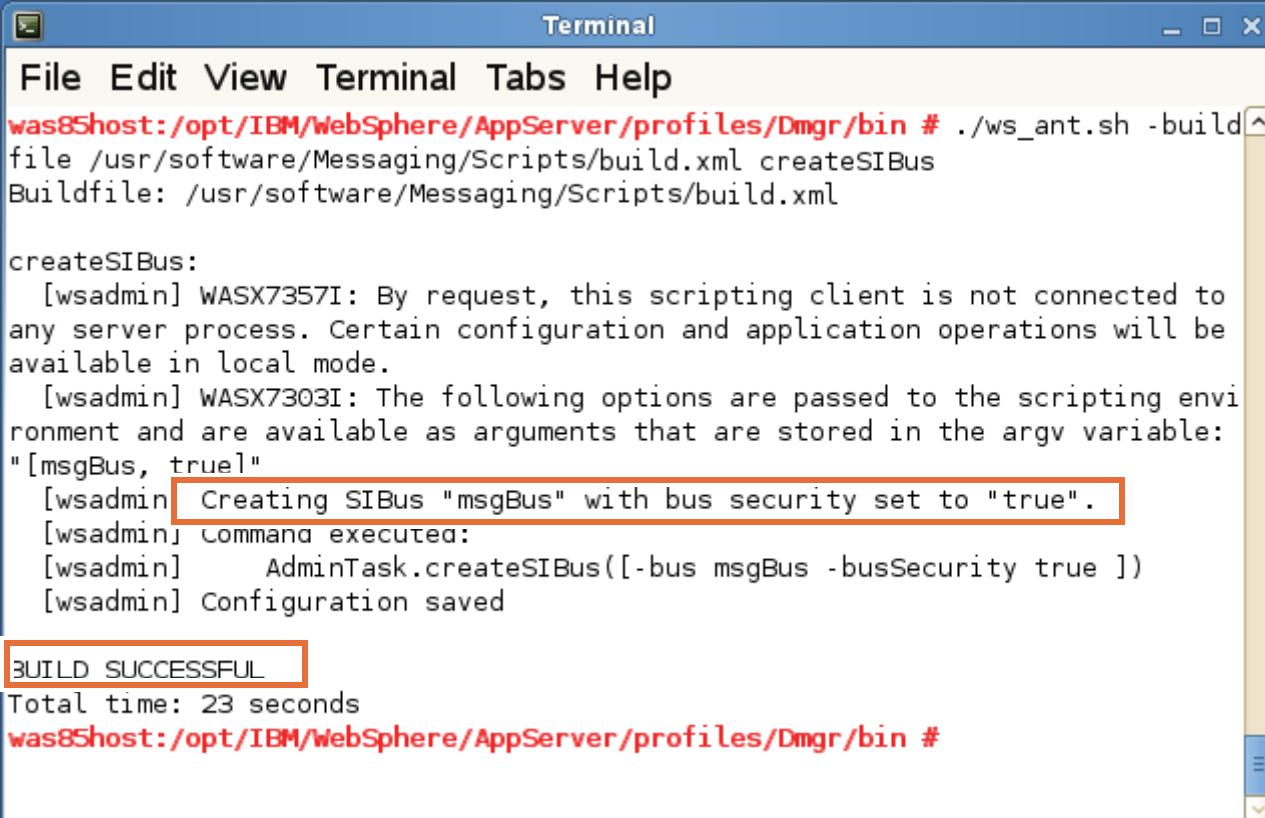
In these steps you create a secured service integration bus, add the PlantsCluster as a bus member, configure an SIBus destination, and authorize specific users and groups for the different bus roles.

- \_\_\_ 1. Shut down all servers in the cell except the deployment manager. Use the stopCell script to stop everything. After all of the servers are stopped, restart the deployment manager.
  - \_\_\_ a. Open a terminal window, and change directory to:  
`/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin`
  - \_\_\_ b. Make sure that the deployment manager is running. If not, enter the  
`./startManager.sh` command.
  - \_\_\_ c. Enter the `./wsadmin.sh -f /usr/software/scripts/stopCell.py` command.
  - \_\_\_ d. Restart the deployment manager with the command: `./startManager.sh`
- \_\_\_ 2. Create a secured service integration bus named: `msgBus`

A service integration bus is required to use the WebSphere default messaging provider. The bus is created with security enabled (by setting the busSecurity property to true as required by the design decisions). Since security is enabled, a user with bus connector authority is created, and an authentication alias for that user is created. The applications use this authentication alias to connect to the bus.

- \_\_\_ a. Make sure that your command prompt is open in the  
`/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin` directory.
- \_\_\_ b. Enter the following command:  
`./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml  
createSIBus`

- \_\_\_ c. The output of the script describes the operations run by the script. Make sure that the build completes successfully by looking for the BUILD SUCCESSFUL message.



```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin # ./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml createSIBus
Buildfile: /usr/software/Messaging/Scripts/build.xml

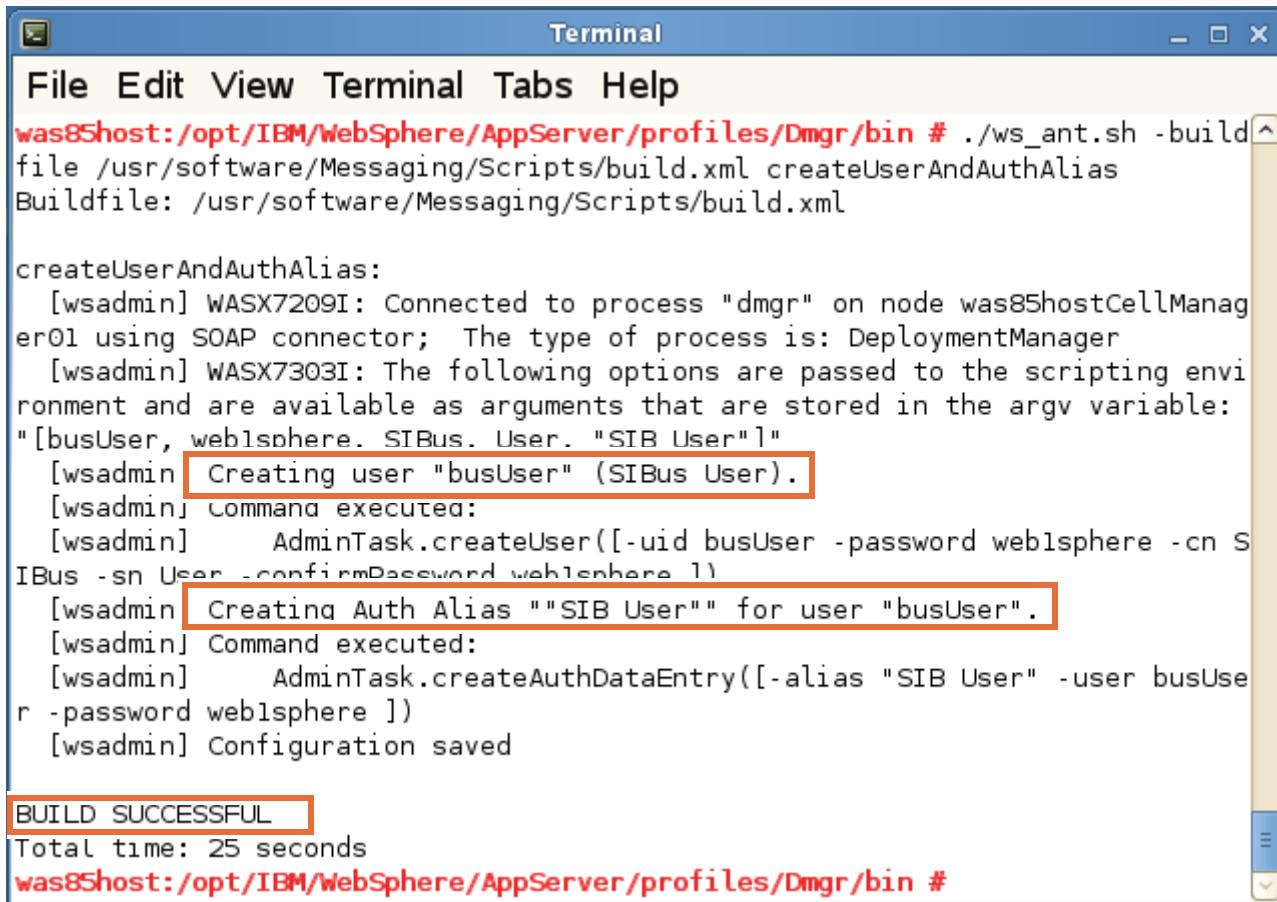
createSIBus:
[wsadmin] WASX7357I: By request, this scripting client is not connected to any server process. Certain configuration and application operations will be available in local mode.
[wsadmin] WASX7303I: The following options are passed to the scripting environment and are available as arguments that are stored in the argv variable:
"[msgBus, true]"
[wsadmin] Creating SIBus "msgBus" with bus security set to "true".
[wsadmin] Command executed:
[wsadmin] AdminTask.createSIBus([-bus msgBus -busSecurity true])
[wsadmin] Configuration saved

BUILD SUCCESSFUL
Total time: 23 seconds
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin #
```

- \_\_\_ d. Enter the following command:

```
./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml
createUserAndAuthAlias
```

- \_\_ e. The output of the script describes the operations run by the script. Make sure that the build completes successfully by looking for the BUILD SUCCESSFUL message.



```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin # ./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml createUserAndAuthAlias
Buildfile: /usr/software/Messaging/Scripts/build.xml

createUserAndAuthAlias:
[wsadmin] WASX7209I: Connected to process "dmgr" on node was85hostCellManager01 using SOAP connector; The type of process is: DeploymentManager
[wsadmin] WASX7303I: The following options are passed to the scripting environment and are available as arguments that are stored in the argv variable:
"[busUser, websphere. SIBus. User. "SIB User"]"
[wsadmin] Creating user "busUser" (SIB User).
[wsadmin] Command executed:
[wsadmin] AdminTask.createUser([-uid busUser -password websphere -cn SIBus -sn User -confirmPassword websphere])
[wsadmin] Creating Auth Alias ""SIB User"" for user "busUser".
[wsadmin] Command executed:
[wsadmin] AdminTask.createAuthDataEntry([-alias "SIB User" -user busUser -password websphere])
[wsadmin] Configuration saved

BUILD SUCCESSFUL
Total time: 25 seconds
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin #
```

- \_\_ f. Enter the following command:

```
./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml
addUserToBusConnectorRole
```

- \_\_\_ g. The output of the script describes the operations run by the script. Make sure that the build completes successfully by looking for the BUILD SUCCESSFUL message.

```

Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin # ./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml addUserToBusConnectorRole
Buildfile: /usr/software/Messaging/Scripts/build.xml

addUserToBusConnectorRole:
[wsadmin] WASX7357I: By request, this scripting client is not connected to any server process. Certain configuration and application operations will be available in local mode.
[wsadmin] WASX7303I: The following options are passed to the scripting environment and are available as arguments that are stored in the argv variable:
" [msgBus, busUser1"
[wsadmin] Adding user "busUser" to bus "msgBus" in bus connector role.
[wsadmin] Command executed:
[wsadmin] AdminTask.addUserToBusConnectorRole([-bus msgBus -user busUser1])
[wsadmin] Configuration saved

BUILD SUCCESSFUL
TOTAL time: 11 seconds
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin #

```

- \_\_\_ 3. Add the cluster as a bus member. A bus member must be added to the bus to provide the runtime services that are needed to handle the JMS protocol. This bus member is added by using the high availability policy as specified in the design decisions. When the bus member is added with the HA policy, a single messaging engine is created. The messaging engine must be told where its message store is located. The message store is at /opt/IBM/WebSphere/AppServer/MessagingFileStores/ as specified in the design decisions.

- \_\_\_ a. Enter the following command:

```

./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml
addClusterBusMember

```

- \_\_\_ b. The output of the script describes the operations run by the script. Make sure that the build completes successfully by looking for the BUILD SUCCESSFUL message.

```

Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin # ./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml addClusterBusMember
Buildfile: /usr/software/Messaging/Scripts/build.xml

addClusterBusMember:
[wsadmin] WASX7357I: By request, this scripting client is not connected to any server process. Certain configuration and application operations will be available in local mode.
[wsadmin] WASX7303I: The following options are passed to the scripting environment and are available as arguments that are stored in the argv variable:
"[msgBus, PlantsCluster, /opt/IBM/WebSphere/AppServer/MessagingFileStores/PlantsCluster.000-msgBus/ME_store, /opt/IBM/WebSphere/AppServer/MessagingFileStores/PlantsCluster.000-msgBus/ME_logs, null]"
[wsadmin] Adding cluster "PlantsCluster" as bus member to bus "msgBus" with "HA" policy.
[wsadmin] Using "/opt/IBM/WebSphere/AppServer/MessagingFileStores/PlantsCluster.000-msgBus/ME_store" for permanent and temporary store.
[wsadmin] Using "/opt/IBM/WebSphere/AppServer/MessagingFileStores/PlantsCluster.000-msgBus/ME_logs" for logs.

[wsadmin] Command executed:
[wsadmin] AdminTask.addSIBusMember([-bus msgBus -cluster PlantsCluster -fileStore -permanentStoreDirectory /opt/IBM/WebSphere/AppServer/MessagingFileStores/PlantsCluster.000-msgBus/ME_store -logDirectory /opt/IBM/WebSphere/AppServer/MessagingFileStores/PlantsCluster.000-msgBus/ME_logs -temporaryStoreDirectory /opt/IBM/WebSphere/AppServer/MessagingFileStores/PlantsCluster.000-msgBus/ME_store -enableAssistance true -policyName HA])
[wsadmin] Configuration saved

BUILD SUCCESSFUL
TOTAL time: 25 seconds
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin #

```

- \_\_\_ 4. Create the SIBus queue destination. The two applications that are installed use a JMS queue to communicate. To support the JMS queue, an SIBus queue destination must be created.

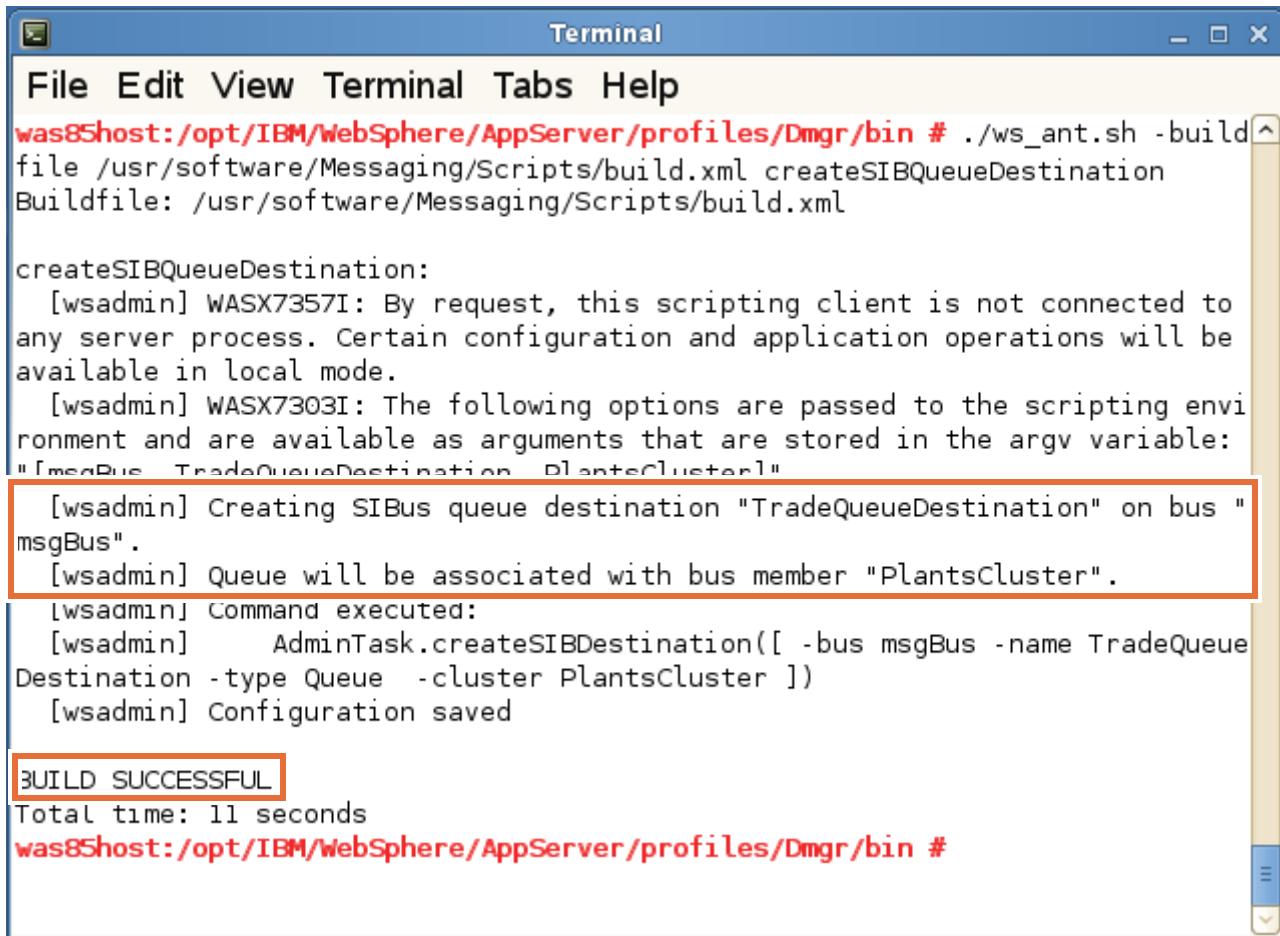
- \_\_\_ a. Enter the following command:

```

./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml
createSIBQueueDestination

```

- \_\_\_ b. The output of the script describes the operations run by the script. Make sure that the build completes successfully by looking for the BUILD SUCCESSFUL message.



```

Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin # ./ws_ant.sh -build
file /usr/software/Messaging/Scripts/build.xml createSIBQueueDestination
Buildfile: /usr/software/Messaging/Scripts/build.xml

createSIBQueueDestination:
[wsadmin] WASX7357I: By request, this scripting client is not connected to
any server process. Certain configuration and application operations will be
available in local mode.
[wsadmin] WASX7303I: The following options are passed to the scripting envi
ronment and are available as arguments that are stored in the argv variable:
["msgBus", "TradeQueueDestination", "PlantsCluster"]
[wsadmin] Creating SIBus queue destination "TradeQueueDestination" on bus "
msgBus".
[wsadmin] Queue will be associated with bus member "PlantsCluster".
[wsadmin] Command executed:
[wsadmin] AdminTask.createSIBDestination([-bus msgBus -name TradeQueue
Destination -type Queue -cluster PlantsCluster])
[wsadmin] Configuration saved

BUILD SUCCESSFUL
Total time: 11 seconds
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin #

```

## Section 5: Configuring the JMS resources

Now that the service integration bus is configured, it is necessary to configure the JMS resources so that the applications can produce and consume messages.

Three JMS resources must be configured for the applications to work; they are:

- JMS connection factory
- JMS queue
- JMS activation specification

Remember that applications have no knowledge of the service integration bus itself. The applications use JMS to place and retrieve messages that use, in this case, a JMS queue.

- \_\_\_ 1. Create a connection factory called: Trade Connection Factory

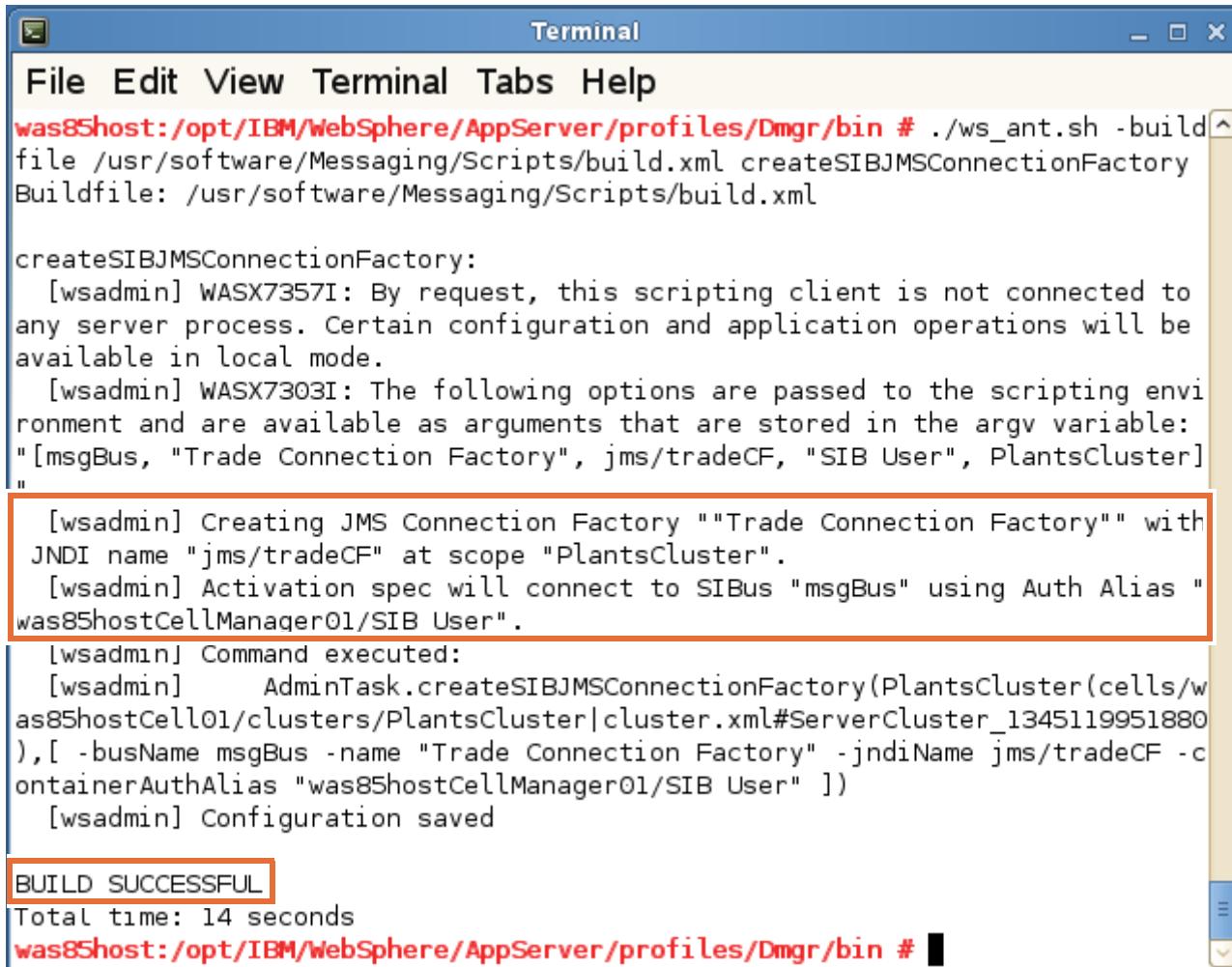
The producer application uses a JMS connection factory to connect to the JMS messaging provider. In this case, the messaging provider is WebSphere default messaging, so the JMS connection factory connects to a messaging engine. The

application uses a JNDI name to look up and retrieve the connection factory object from the naming service. Since the application runs in a cluster, the connection factory is created at cluster scope.

- \_\_ a. Enter the following command:

```
./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml
createSIBJMSConnectionFactory
```

- \_\_ b. The output of the script describes the operations run by the script. Make sure that the build completes successfully by looking for the BUILD SUCCESSFUL message.



```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin # ./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml createSIBJMSConnectionFactory
Buildfile: /usr/software/Messaging/Scripts/build.xml

createSIBJMSConnectionFactory:
[wsadmin] WASX7357I: By request, this scripting client is not connected to any server process. Certain configuration and application operations will be available in local mode.
[wsadmin] WASX7303I: The following options are passed to the scripting environment and are available as arguments that are stored in the argv variable:
"[msgBus, "Trade Connection Factory", jms/tradeCF, "SIB User", PlantsCluster]
"
[wsadmin] Creating JMS Connection Factory "Trade Connection Factory" with JNDI name "jms/tradeCF" at scope "PlantsCluster".
[wsadmin] Activation spec will connect to SIBus "msgBus" using Auth Alias "was85hostCellManager01/SIB User".
[wsadmin] Command executed:
[wsadmin] AdminTask.createSIBJMSConnectionFactory(PlantsCluster(cells/was85hostCell01/clusters/PlantsCluster|cluster.xml#ServerCluster_1345119951880),[-busName msgBus -name "Trade Connection Factory" -jndiName jms/tradeCF -containerAuthAlias "was85hostCellManager01/SIB User"])
[wsadmin] Configuration saved

BUILD SUCCESSFUL
Total time: 14 seconds
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin #
```

- \_\_ 2. Create a queue named: Trade Processor Queue

This object is a JMS queue. The configuration gives it a JNDI name. The JMS queue is mapped to the SIBus queue destination that was created earlier in this exercise.

- \_\_ a. Enter the following command:

```
./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml
createSIBJMSQueue
```

- \_\_\_ b. The output of the script describes the operations run by the script. Make sure that the build completes successfully by looking for the BUILD SUCCESSFUL message.

```

Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin # ./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml createsSIBJMSQueue
Buildfile: /usr/software/Messaging/Scripts/build.xml

createSIBJMSQueue:
[wsadmin] WASX7357I: By request, this scripting client is not connected to any server process. Certain configuration and application operations will be available in local mode.
[wsadmin] WASX7303I: The following options are passed to the scripting environment and are available as arguments that are stored in the argv variable:
"[msgBus, "Trade Processor Queue", jms/tradeprocq, TradeQueueDestination, PlantsCluster1"]

[wsadmin] Creating JMS Queue ""Trade Processor Queue"" with JNDI name "jms/tradeprocq" at scope "PlantsCluster".
[wsadmin] Queue will be associated to SIBus Queue "TradeQueueDestination" on bus "msgBus".
[wsadmin] Command executed:
[wsadmin] AdminTask.createSIBJMSQueue(PlantsCluster(cells/was85hostCell01/clusters/PlantsCluster|cluster.xml#ServerCluster_1345119951880), [-busName msgBus -name "Trade Processor Queue" -jndiName jms/tradeprocq -queueName TradeQueueDestination])
[wsadmin] Configuration saved

BUILD SUCCESSFUL
Total time: 11 seconds
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin #

```

- \_\_\_ 3. Create the activation specification.

The consumer application uses a JMS activation spec to connect to the JMS messaging provider. In this case, the messaging provider is WebSphere default messaging, so the JMS activation spec connects to a messaging engine. The application uses a JNDI name to look up and retrieve the activation spec object from the naming service. The activation spec itself is configured with the JNDI name of the JMS destination it is to consume from. Since the application is running in a cluster, the activation spec is created at cluster scope.

- \_\_\_ a. Enter the following command:

```

./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml
createsSIBJMSActivationSpec

```

- \_\_\_ b. The output of the script describes the operations run by the script. Make sure that the build completes successfully by looking for the BUILD SUCCESSFUL message.

```
File Edit View Terminal Tabs Help
was8host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin # ./ws_ant.sh -build
file /usr/software/Messaging/Scripts/build.xml createSIBJMSActivationSpec
Buildfile: /usr/software/Messaging/Scripts/build.xml

createSIBJMSActivationSpec:
[wsadmin] WASX7357I: By request, this scripting client is not connected to
any server process. Certain configuration and application operations will be
available in local mode.
[wsadmin] WASX7303I: The following options are passed to the scripting envi
ronment and are available as arguments that are stored in the argv variable:
"[msgBus, "Trade Processor Activation Spec", jms/tradeAS, jms/tradeprocq, "SI
B User". PlantsCluster1"
[wsadmin] Creating JMS Activation Spec ""Trade Processor Activation Spec""
with JNDI name "jms/tradeAS" at scope "PlantsCluster".
[wsadmin] Activation spec will be associated with destination "jms/tradepro
cq".
[wsadmin] Activation spec will connect to SIBus "msgBus" using Auth Alias "
was8hostCellManager01/SIB User".
[wsadmin] Command executed:
[wsadmin] AdminTask.createSIBJMSActivationSpec(PlantsCluster(cells/was8
5hostCell01/clusters/PlantsCluster|cluster.xml#ServerCluster_1345119951880), [
-busName msgBus -name "Trade Processor Activation Spec" -jndiName jms/tradeA
S -destinationJndiName jms/tradeprocq -destinationType Queue -authenticationA
lias "was85hostCellManager01/SIB User"]
[wsadmin] Configuration saved

BUILD SUCCESSFUL
Total time: 13 seconds
was8host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin #
```

## Section 6: Installing the messaging applications

As mentioned at the beginning of this exercise, two applications are used to demonstrate messaging on the service integration bus. The first application produces messages that are placed on a queue, and the second application consumes those messages. Now that the message bus and the JMS resources are configured, you must install the applications.

- \_\_\_ 1. Install the MSGSenderSimulator application.

Since this application is writing messages to the bus, it must know how to find the connection factory that was previously defined. It also must know how to find the JMS queue. The application uses references to find the JNDI name for the

connection factory and queue. These references must be assigned when the application is installed.

- \_\_ a. Enter the following command:

```
./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml
installProducerApp
```

- \_\_ b. The output of the script describes the operations run by the script. Make sure that the build completes successfully by looking for the BUILD SUCCESSFUL message.

```
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin # ./ws_ant.sh -build
file /usr/software/Messaging/Scripts/build.xml installProducerApp
Buildfile: /usr/software/Messaging/Scripts/build.xml

installProducerApp:
[wsadmin] WASX7357I: By request, this scripting client is not connected to
any server process. Certain configuration and application operations will be
available in local mode.
[wsadmin] WASX7303I: The following options are passed to the scripting envi
ronment and are available as arguments that are stored in the argv variable:
"/[usr/software/Messaging/MSGSenderSimulator.ear, jms/tradeprocq, jms/tradeCF
"STR User". PlantsCluster"
[wsadmin] Installing app "/usr/software/Messaging/MSGSenderSimulator.ear" t
o cluster "PlantsCluster".
[wsadmin] Setting destination ref for app to "jms/tradeprocq".
[wsadmin] Setting resource ref for app to "jms/tradeCF" using Auth Alias "w
as85hostCellManager01/SIB User".
[wsadmin] Command executed.
[wsadmin] AdminApp.install(/usr/software/Messaging/MSGSenderSimulator.e
ar, [-cluster PlantsCluster -MapMessageDestinationRefToEJB [[".** ".** ".*"
".** jms/tradeprocq]] -MapResRefToEJB [[".** ".** ".** ".** jms/tradeC
F DefaultPrincipalMapping [was85hostCellManager01/SIB User]]]])
[wsadmin] ADMA5016I: Installation of MSGSenderSimulator started.

[wsadmin] ADMA5013I: Application MSGSenderSimulator installed successfully.
[wsadmin] Configuration saved

BUILD SUCCESSFUL
Total time: 21 seconds
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin #
```

- \_\_ 2. Install the Trade processor application.

This application consumes the messages, so it must know how to find the activation specification (already coded into the application).

- \_\_ a. Enter the following command:

```
./ws_ant.sh -buildfile /usr/software/Messaging/Scripts/build.xml
installConsumerApp
```

- \_\_\_ b. The output of the script describes the operations run by the script. Make sure that the build completes successfully by looking for the BUILD SUCCESSFUL message.

```

Terminal
File Edit View Terminal Tabs Help
was8$ host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin # ./ws_ant.sh -build
file /usr/software/Messaging/Scripts/build.xml installConsumerApp
Buildfile: /usr/software/Messaging/Scripts/build.xml

installConsumerApp:
[wsadmin] WASX7357I: By request, this scripting client is not connected to
any server process. Certain configuration and application operations will be
available in local mode.
[wsadmin] WASX7303I: The following options are passed to the scripting envi
ronment and are available as arguments that are stored in the argv variable:
"/usr/software/Messaging/TPApplication.ear PlantsCluster"
[wsadmin] Installing app "/usr/software/Messaging/TPApplication.ear" to clu
ster "PlantsCluster".
[wsadmin] Command executed:
[wsadmin] AdminApp.install(/usr/software/Messaging/TPApplication.ear,[
-cluster PlantsCluster])
[wsadmin] ADMA5016I: Installation of TradeProcessorApplication started.

[wsadmin] ADMA5013I: Application TradeProcessorApplication installed succe
ssfully.
[wsadmin] Configuration saved

BUILD SUCCESSFUL
Total time: 14 seconds
was8$ host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin #

```

## Section 7: Examine the configured environment

In this part of the exercise, you use the administrative console to examine the configuration that was created by using scripting.

- \_\_\_ 1. Examine the producer application.

The producer application is called MSGSenderSimulator. The application is installed to the PlantsCluster. The producer application has a resource reference to a ConnectionFactory, which uses the JNDI name `jms/tradeCF`. It also has a resource reference to a destination, which uses the JNDI name `jms/tradeprocq`.

- \_\_\_ a. From the administrative console, click **Applications > Application Types > WebSphere enterprise applications**.
- \_\_\_ b. In the Enterprise Applications list, click **MSGSenderSimulator**.

- \_\_\_ c. Under Modules, click **Manage Modules**. Note the MSGSenderWeb module is mapped to the PlantsCluster.

A screenshot of a software interface titled 'Manage Modules'. At the top, there are four buttons: 'Remove', 'Update', 'Remove File', and 'Export File'. Below these are two icons: a checkmark and a trash can. The main area is a table with columns: 'Select', 'Module', 'URI', 'Module Type', and 'Server'. A single row is selected, showing 'MSGSenderWeb' in the Module column, 'MSGSenderWeb.war,WEB-INF/web.xml' in the URI column, 'Web Module' in the Module Type column, and 'WebSphere:cell=was85hostCell01,cluster=PlantsCluster' in the Server column. This last column is highlighted with a red box.

Select	Module	URI	Module Type	Server
<input type="checkbox"/>	MSGSenderWeb	MSGSenderWeb.war,WEB-INF/web.xml	Web Module	WebSphere:cell=was85hostCell01,cluster=PlantsCluster

- \_\_\_ d. Click **MSGSenderSimulator** in the breadcrumb trail.  
 \_\_\_ e. Under References, click **Resource references**.  
 \_\_\_ f. Note the Target Resource JNDI Name for the qConnectionFactory is jms/tradeCF. The connection factory uses the "SIB User" authentication alias to connect to the bus.

A screenshot of a software interface titled 'Resource references'. At the top, there are two buttons: 'Application Method...' and 'Extended Properties...'. The main area is a table with columns: 'Resource Reference', 'Target Resource JNDI Name', and 'Login configuration'. A single row is selected, showing 'com.ibm.msgsender.servlet.MSGSenderServlet/qConnectionFactory' in the Resource Reference column, 'jms/tradeCF' in the Target Resource JNDI Name column, and a detailed configuration section in the Login configuration column. The 'Target Resource JNDI Name' cell is highlighted with a red box. The 'Login configuration' section is also highlighted with a red box.

Resource Reference	Target Resource JNDI Name	Login configuration
com.ibm.msgsender.servlet.MSGSenderServlet/qConnectionFactory	jms/tradeCF	Resource authorization: Container Authentication method: DefaultPrincipalMapping  Authentication data entry: was85hostCellManager01/SIB User

- \_\_\_ g. Click **MSGSenderSimulator** in the breadcrumb trail.  
 \_\_\_ h. Under References, click **Message destination references**. Notice that the Target Resource JNDI Name for the applications queue is jms/tradeprocq.

A screenshot of a software interface titled 'Message destination references'. At the top, there are two buttons: 'Application Method...' and 'Extended Properties...'. The main area is a table with columns: 'RI', 'Message destination object', 'Type', and 'Target Resource JNDI Name'. A single row is selected, showing 'MSGSenderWeb.war,WEB-INF/web.xml' in the RI column, 'com.ibm.msgsender.servlet.MSGSenderServlet/queue' in the Message destination object column, 'Reference' in the Type column, and 'jms/tradeprocq' in the Target Resource JNDI Name column. The 'Target Resource JNDI Name' cell is highlighted with a red box.

RI	Message destination object	Type	Target Resource JNDI Name
MSGSenderWeb.war,WEB-INF/web.xml	com.ibm.msgsender.servlet.MSGSenderServlet/queue	Reference	jms/tradeprocq

## 2. Examine the consumer application.

The consumer application is called TradeProcessorApplication. The application is installed to the PlantsCluster. The consumer application does not use a resource reference to access the JNDI name of its activation spec. The JNDI name is hardcoded into the application.

- \_\_ a. Click Applications > Application Types > WebSphere enterprise applications.
- \_\_ b. In the Enterprise Applications list, click **TradeProcessorApplication**.
- \_\_ c. Under Modules, click **Manage Modules**. Notice that all modules are mapped to the PlantsCluster.

	URI	Module Type	Server
	TPEJB.jar,META-INF/ejb-jar.xml	EJB Module	WebSphere:cell=was85hostCell01,cluster=PlantsCluster
TradeProcessorWeb	Trade ProcessorWeb.war,WEB-INF/web.xml	Web Module	WebSphere:cell=was85hostCell01,cluster=PlantsCluster

- \_\_ 3. Examine the JMS queue destination.
- \_\_ a. In the navigation tree, click **Resources > JMS > Queues**.
- \_\_ b. In the queues list, click **Trade Processor Queue**. Notice that the JNDI name for the queue is `jms/tradeprocq`, and that it is associated with the `TradeQueueDestination` on the bus called `msgBus`.

\* Name  
Trade Processor Queue

\* JNDI name  
jms/tradeprocq

Description

**Connection**

Bus name  
msgBus

\* Queue name  
TradeQueueDestination

- \_\_ 4. Examine the JMS connection factory.
- \_\_ a. Click **Resources > JMS > Connection factories**.
- \_\_ b. In the connection factories list, click **Trade Connection Factory**. Notice that the JNDI name for the connection factory is `jms/tradeCF`. It uses the authentication alias "SIB User" to connect to the bus `msgBus`.

[Connection factories > Trade Connection Factory](#)

A JMS connection factory is used to create connections to the associated JMS provider of JMS destinations, for publish/subscribe messaging. Use connection factory administrative objects to manage JMS connection factories for messaging provider.

The screenshot shows the 'Configuration' tab selected in the top navigation bar. Below it, the 'General Properties' section is active. The configuration details are as follows:

- Scope:** Cluster=PlantsCluster
- Provider:** Default messaging provider
- Name:** Trade Connection Factory
- JNDI name:** jms/tradeCF (highlighted with a red box)
- Description:** (empty)
- Category:** (empty)

Below the General Properties, the 'Connection' section is expanded, showing:

- Bus name:** msgBus (highlighted with a red box)

Under the 'Security settings' section, the following fields are present:

- Authentication alias for XA recovery:** (none)
- Mapping-configuration alias:** DefaultPrincipalMapping
- Container-managed authentication alias:** was85hostCellManager01/SIB User (highlighted with a red box)

- \_\_\_ 5. Examine the JMS activation spec.
  - \_\_\_ a. Click **Resources > JMS > Activation specifications.**
  - \_\_\_ b. In the activation specifications list, click **Trade Processor Activation Spec.**  
Notice that the JNDI name for the activation specification is `jms/tradeAS`. It is associated with the JMS queue destination with JNDI name `jms/tradeprocq`. The activation spec connects to the bus with the authentication alias “SIB User.”

**Activation specifications**

**Activation specifications > Trade Processor Activation Spec**

A JMS activation specification is associated with one or more message-driven beans and provides the configuration for them to receive messages.

**Configuration**

**General Properties**

**Administration**

- Scope**: Cluster=PlantsCluster
- Provider**: Default messaging provider
- \* Name**: Trade Processor Activation Spec
- \* JNDI name**: jms/tradeAS

**Description**

**Destination**

- \* Destination type**: Queue
- \* Destination JNDI name**: jms/tradeprocq
- Message selector**:
- \* Bus name**: msgBus

**Security settings**

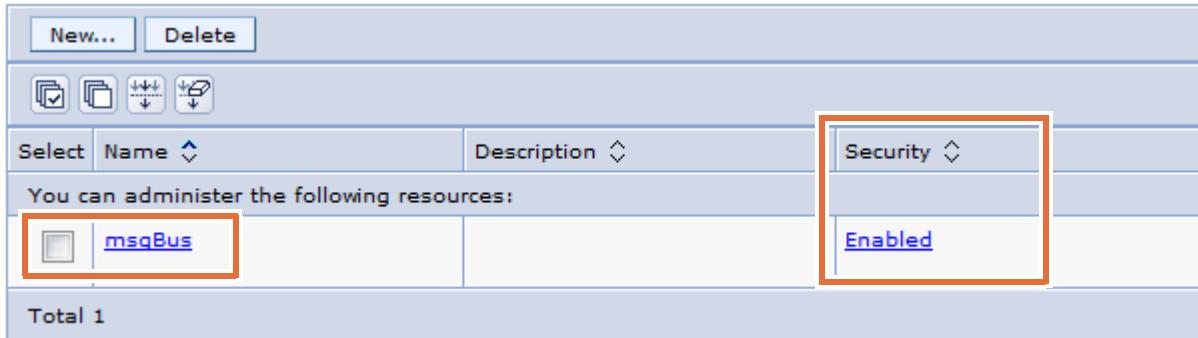
Select the authentication values for this resource.

- Authentication alias**: was85hostCellManager01/SIB User

**Related Item**

- JAAS - authen data
- Buses

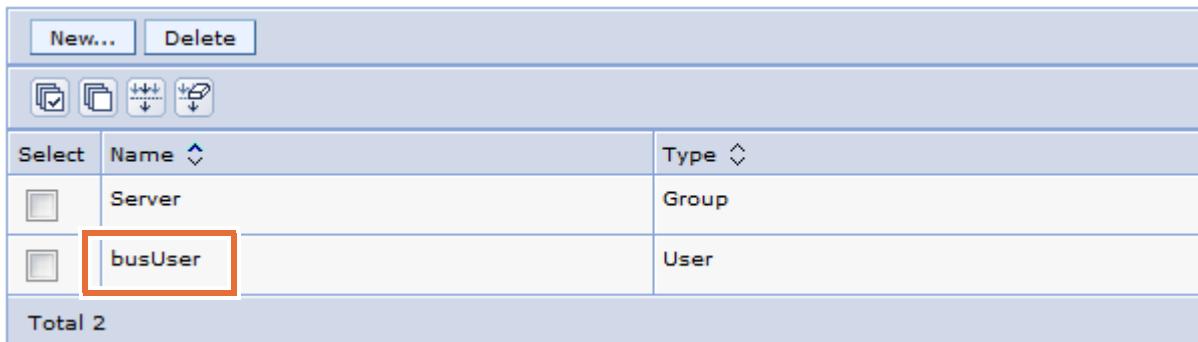
- \_\_\_ 6. Examine the SIBus configuration.
- \_\_\_ a. In the navigation tree, click **Service integration > Buses**.
- \_\_\_ b. In the list of buses, notice that security for the bus named `msgBus` is enabled.



The screenshot shows a table with columns: Select, Name, Description, and Security. A row for 'msgBus' is selected, indicated by a red box around the entire row. The 'Security' column for this row also has a red box around it, highlighting the 'Enabled' link.

Select	Name	Description	Security
<i>You can administer the following resources:</i>			
<input type="checkbox"/>	<a href="#">msgBus</a>		<a href="#">Enabled</a>
Total 1			

- \_\_\_ c. In the list of buses, for the bus named `msgBus`, click the link that says **Enabled**.
- \_\_\_ d. Under Authorization Policy, click **Users and groups in the bus connector role**.
- \_\_\_ e. Notice a group that is called `Server`. This group gives application servers authority to connect to the bus. This group does not give applications authority to connect to the bus. Notice that there is a user called `busUser`. This user gives the producer and consumer applications authority to connect to the bus.



The screenshot shows a table with columns: Select, Name, Type. Two rows are present: 'Server' (Group) and 'busUser' (User). The 'busUser' row is selected, indicated by a red box around the entire row.

Select	Name	Type
<input type="checkbox"/>	Server	Group
<input type="checkbox"/>	<a href="#">busUser</a>	User
Total 2		

- \_\_\_ f. Click **Service integration > Buses**.
- \_\_\_ g. In the list of buses, click **msgBus**.
- \_\_\_ h. Under Topology, click **Bus members**.

- \_\_\_ i. Note: one bus member exists that is called PlantsCluster, which is of cluster type, and is enabled with the high availability policy.

The screenshot shows a user interface for managing bus members. At the top, there are 'Add' and 'Remove' buttons. Below them is a toolbar with icons for selecting, adding, removing, and other operations. A search bar allows filtering by 'Name' and 'Type'. A message says 'You can administer the following resources:'. A table lists one resource: 'PlantsCluster' (Type: Cluster, Status: Enabled (High availability)). A red box highlights this row. At the bottom, it says 'Total 1'.

- \_\_\_ 7. Examine the messaging engine configuration.
- \_\_\_ a. From the breadcrumb trail, click **msgBus**.
  - \_\_\_ b. Under Topology, click **Messaging engines**.
  - \_\_\_ c. Notice that there is only one messaging engine for the cluster. This topology is used because the policy for the cluster bus member is set to high availability.

The screenshot shows a user interface for managing messaging engines. At the top, there are 'Start' and 'Stop' buttons. Below them is a toolbar with icons for selecting, adding, removing, and other operations. A search bar allows filtering by 'Name' and 'Description'. A message says 'You can administer the following resources:'. A table lists one resource: 'PlantsCluster.000-msgBus' (Status: Running). A red box highlights this row. At the bottom, it says 'Total 1'.

- \_\_\_ d. From the list of messaging engines, click **PlantsCluster.000-msgBus**.

- \_\_\_ e. Notice that the Message store type is a File store.

[Buses > msgBus > Messaging engines > PlantsCluster.000-msgBus](#)

A messaging engine is a component, running inside a server, that manages messaging requests. Applications are connected to a messaging engine when they access a service integration point.

The screenshot shows the 'Runtime' tab selected in the navigation bar. Below it, the 'General Properties' section is displayed. The 'Message store type' dropdown is highlighted with a red box. Other visible fields include 'Name' (PlantsCluster.000-msgBus), 'UUID' (D30F15DDBADF8D04), 'Description' (empty), 'Initial state' (Started), and 'Message store type' (File store).

- \_\_\_ f. Under Additional Properties, click **Message store**.  
\_\_\_ g. Notice that the directory for logs is called `ME_logs`, and the directory for the temporary store is called `ME_store`.

**Buses**

[Buses](#) > [msgBus](#) > [Messaging engines](#) > [PlantsCluster.000-msgBus](#) > File store

The persistent store for messages and other state managed by the messaging engine.

Configuration

**General Properties**

UUID  
12D0D710B034D7F7

**Log**

\* Log size  
100 MB

Log directory path  
/opt/IBM/WebSphere/AppServer/MessagingFileStores  
/PlantsCluster.000-msgBus/ME\_logs

**Permanent store**

\* Minimum permanent store size  
200 MB

Unlimited permanent store size

\* Maximum permanent store size  
500 MB

Permanent store directory path  
/opt/IBM/WebSphere/AppServer/MessagingFileStores  
/PlantsCluster.000-msgBus/ME\_store

**Temporary store**

\* Minimum temporary store size  
200 MB

Unlimited temporary store size

\* Maximum temporary store size  
500 MB

Temporary store directory path  
/opt/IBM/WebSphere/AppServer/MessagingFileStores  
/PlantsCluster.000-msgBus/ME\_store

Apply OK Reset Cancel

## Section 8: Applications that are used in this exercise

The two applications that you installed for this exercise provide the capability to produce messages to a JMS queue, and to consume messages from a JMS queue.

**Table 13: Applications and their function**

Application	Function
Message Sender Simulator MSGSenderSimulator.ear	This application simulates the function of buying and selling stocks. You can select how many shares to buy or sell. For each operation, a message is placed on a JMS queue. Each message has a transaction number that includes the name of the server that produced it and a sequence number. You can view the action of placing the message on the queue in the SystemOut log file.
TradeProcessorApplication TPApplication.ear	A message-driven bean EJB (MDB) drives this application. This bean listens on the same queue on which the simulator places messages. As the MDB receives messages, the messages are listed in a table on a web page and displayed to the user. Every 30 seconds the oldest message is assumed to be processed and is removed from the table. As messages are received and discarded, trace entries are written to the SystemOut log file.

Below you see the user interface for both applications:

### Sending messages from server: server1

Select number of messages to be sent for each of the **Buy** and **Sell** categories, then click **Send messages**.

Buy messages	<input type="text" value="1"/>
Sell messages	<input type="text" value="2"/>
<b>Send messages</b>	

From the `Monitor.html` web page, you can create Buy messages and Sell messages and send them to a server for processing.

Trade Requests To Be Processed On Server: Server1					
Account	Buy/Sell	Symbol	Qty	Total cost	Transaction
23423234 (John Doe)	BUY	PG	10.0	\$657.90	server1 - 1
87652289 (Elaine Moose)	SELL	DELL	1.0	\$19.89	server1 - 3
87652289 (Elaine Moose)	SELL	DELL	1.0	\$19.89	server1 - 2

From the same `Monitor.html` web page, you can see which server received the messages for processing. The Transaction column shows what server sent the message and the order in which the message was received.

## ***Section 9: Testing the applications and exploring messaging engine policies***

In this part of the exercise, you test the applications under different messaging engine policies. Also, you explore the effects of configuring different activation specification options and queue options. The first configuration you test is a single messaging engine that uses the high availability policy and all other default options.

Before you can test the applications, both node agents and both cluster members must be started.

- \_\_\_ 1. Stop the deployment manager. Restart the cell. Restarting the deployment manager ensures that the configuration it has in memory matches the configuration on disk.
  - \_\_\_ a. In a command window, navigate to `<profile_root>/Dmgr/bin` and enter the `./stopManager.sh` command.
  - \_\_\_ b. Wait until the deployment manager is stopped.
  - \_\_\_ c. Enter the `./startManager.sh` command.
  - \_\_\_ d. Wait until the deployment manager is running.
  - \_\_\_ e. In a command window, navigate to `<profile_root>/profile1/bin` and enter the `./startNode.sh` command.
  - \_\_\_ f. Wait until the node agent is running.

- \_\_\_ g. To start the second node agent, navigate to <profile\_root>/profile2/bin and enter the ./startNode.sh command.
- \_\_\_ h. Wait until the node agent is running.
- \_\_\_ i. Using the administrative console, verify that both node agents are started by clicking **System administration > Node agents**.

The screenshot shows the 'Node agents' list in the administrative console. The columns are: Select, Name, Node, Host Name, Version, and Status. Two rows are present:

Select	Name	Node	Host Name	Version	Status
<input type="checkbox"/>	<a href="#">nodeagent</a>	was85hostNode01	was85host	ND 8.5.0.0	
<input type="checkbox"/>	<a href="#">nodeagent</a>	was85hostNode02	was85host	ND 8.5.0.0	

- \_\_\_ j. Click **System administration > Nodes**.
  - \_\_\_ k. Synchronize both nodes.
2. Start server1 and then start server2.
- \_\_\_ a. Click **Servers > Server Types > WebSphere application servers**.
  - \_\_\_ b. Check the box for **server1** and click **Start**.
  - \_\_\_ c. Wait for the Status indicator to turn to a solid green arrow (it can take several minutes).
  - \_\_\_ d. Check the box for **server2** and click **Start**.
  - \_\_\_ e. Wait for the Status indicator to turn to a solid green arrow (it can take several minutes).

The screenshot shows the 'Servers' list in the administrative console. The columns are: New..., Delete, Templates..., Start, Stop, Restart, Immediate Stop, Terminate, Select, Name, Node, Host Name, Version, Cluster Name, and Status. Two rows are present:

New...	Delete	Templates...	Start	Stop	Restart	Immediate Stop	Terminate	Select	Name	Node	Host Name	Version	Cluster Name	Status
								<input type="checkbox"/>	<a href="#">server1</a>	was85hostNode01	was85host	ND 8.5.0.0	PlantsCluster	
								<input type="checkbox"/>	<a href="#">server2</a>	was85hostNode02	was85host	ND 8.5.0.0	PlantsCluster	

Total 2

**Note**

You specified the high availability policy with no additional customization for the PlantsCluster bus member. The messaging engine starts in the first cluster member that was started. In this case, the messaging engine is started on server1, and the messaging engine is in joined state on server2.

3. Ensure that both the applications you installed earlier are running.

- \_\_ a. Click **Applications > Application Types > WebSphere enterprise applications.**
- \_\_ b. Verify that both applications are in a started state.

<input type="checkbox"/>	<a href="#">MSGSEnderSimulator</a>	
<input type="checkbox"/>	<a href="#">TradeProcessorApplication</a>	

**Note**

If either of the applications is not running, select the application and click **Start**. If the application does not start, check the SystemOut logs to determine what is preventing the application from starting. Check the steps that you went through when you installed the applications. The problem is most likely related to the resolution of resource references or authentication aliases not being applied to the resource references. If after you check these settings you still have a problem, contact your instructor.



## Information

Two applications are installed, a message producer and a message consumer. The message simulator application is the message producer; the trade processor application is the consumer of messages. Since the applications are installed in a cluster, each server in the cluster runs both applications.

When you use the **high availability messaging engine policy**, only one messaging engine is started in the cluster. The WebSphere high availability manager decides which server in the cluster gets to run the messaging engine. Usually the first server that is started in the cluster is chosen. If the server that runs the messaging engine fails, the high availability manager chooses another cluster member and starts the messaging engine there. In this configuration, the messaging engine can fail over to another cluster member, thus providing high availability.

Producers in any cluster member can generate and place messages on a queue. However, only the consumer that is on the same server as the running messaging engine gets to consume and process the messages. In the next few steps, you prove these points.

- 4. Start the `Monitor.html` page. This web page is provided as part of the Trade processor application to more easily demonstrate the interactions between the application servers and messaging engines.

- \_\_\_ a. Go to the home page of the browser and select the link to open the **Trade processor monitor**. Alternatively, enter the URL  
<http://was85host:9080/Trade/processor/Monitor.html> in the browser.

The page has four frames:

- **Upper left:** Message producer simulator that runs on server1
- **Upper right:** Trade processor that runs on server1
- **Lower left:** Message producer simulator that runs on server2
- **Lower right:** Trade processor that runs on server2

In the message sender applications you can select how many messages of each kind, buy or sell, are sent to the queue.

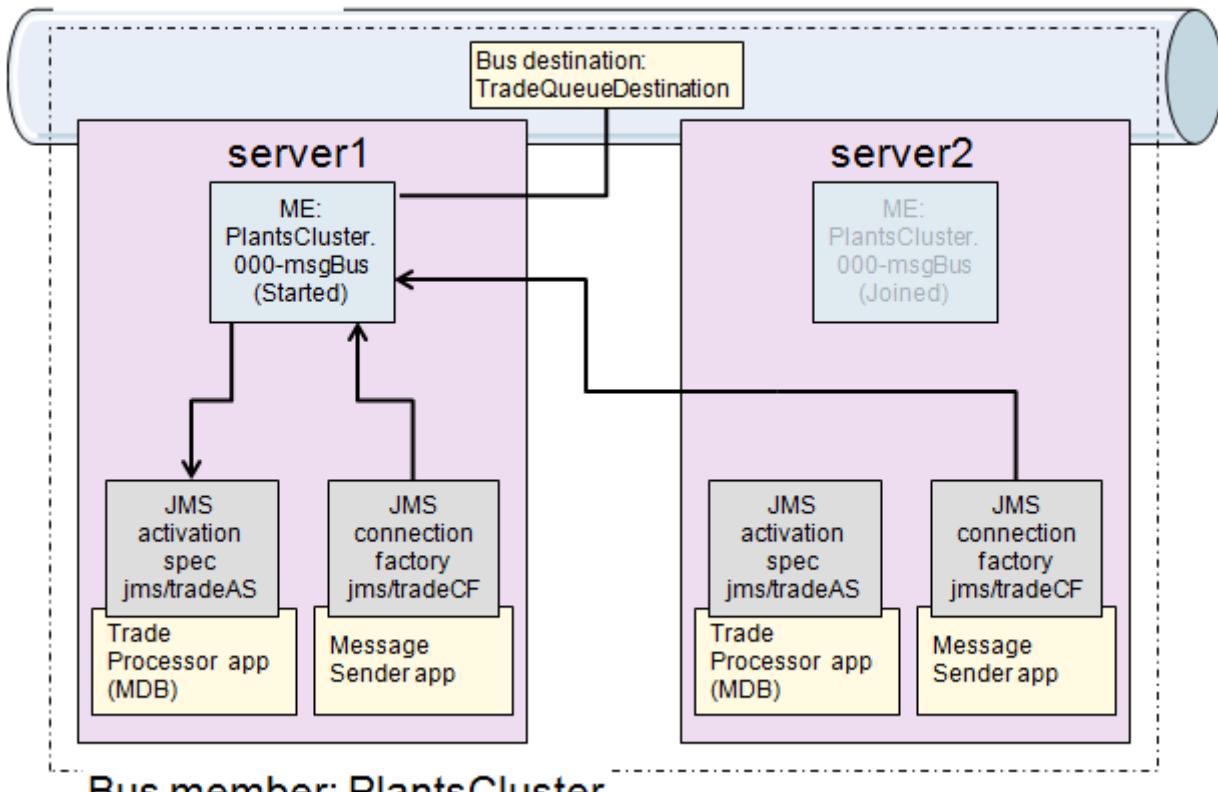


The bookmark on the home page sends the request to server1 (port 9080). You can modify the URL to use port 9081 for the web address of the monitor. Doing that would serve the page from server2 instead of server1.

- \_\_\_ 5. Validate the expected behavior of the application when the cluster bus member uses the high availability policy.

Because server1 was started first, it has the active messaging engine. By default, if a consumer application runs on the same cluster as the messaging engines, only the MDB with the active messaging engine receives work. As shown in the following diagram, no messages are sent to the MDB on server2, although the producer application instance that is running on server2 can still send messages.

### Bus: msgBus



- \_\_ a. From the upper left frame of the Monitor.html web page (sending messages from server: server1), leave the default values (one message) for Buy and Sell messages, and click **Send messages**.



- \_\_ b. Observe which server processed the messages: server1 or server2.
- \_\_ c. Record the server name here: \_\_\_\_\_  
In the high availability policy, only one server in the cluster is running the messaging engine.
- \_\_ d. From the lower left frame of the Monitor.html web page (sending messages from server: server2), leave the default values (one message) for Buy and Sell messages and click **Send messages**.
- \_\_ e. Record the server name here: \_\_\_\_\_  
The same server is processing messages that are sent from both server1 and server2.

**Note**

\_\_\_\_\_

If everything is configured as expected, then server1 consumes all of the messages that both server1 and server2 produce.

- \_\_\_ f. Examine the log files for server1. If you did not enable HPEL for that server, open its `SystemOut.log` file with a text editor. If HPEL is enabled, use the Log Viewer and go to the last page of the logs. Look for messages similar to the following.

8/25/11 15:34:11.826	00000042	SystemOut	DETAIL	MDB received message: server1 - 1
8/25/11 15:34:11.842	00000042	SystemOut	DETAIL	MDB received message: server1 - 2
8/25/11 15:34:41.170	0000003C	SystemOut	DETAIL	Removed message: server1 - 1
8/25/11 15:35:02.529	0000003A	SystemOut	DETAIL	Message: server2 - 1, was placed on the queue.
8/25/11 15:35:02.560	0000003A	SystemOut	DETAIL	Message: server2 - 2, was placed on the queue.
8/25/11 15:35:02.576	00000042	SystemOut	DETAIL	MDB received message: server2 - 1
8/25/11 15:35:02.576	00000042	SystemOut	DETAIL	MDB received message: server2 - 2
8/25/11 15:35:11.404	0000003C	SystemOut	DETAIL	Removed message: server1 - 2
8/25/11 15:35:41.889	0000003C	SystemOut	DETAIL	Removed message: server2 - 1
8/25/11 15:36:11.967	0000003B	SystemOut	DETAIL	Removed message: server2 - 2



### Note

The application that receives the messages is the one running on the server where the messaging engine is active (server1 in this case). You can verify this behavior by looking at the logs for both cluster members. Look for the message that indicates the messaging engine is in the Started state:

Messaging engine PlantsCluster.000-msgBus is in state Started.

The other server in the cluster (server2 in this case) has a message that indicates the messaging engine is in the joined state:

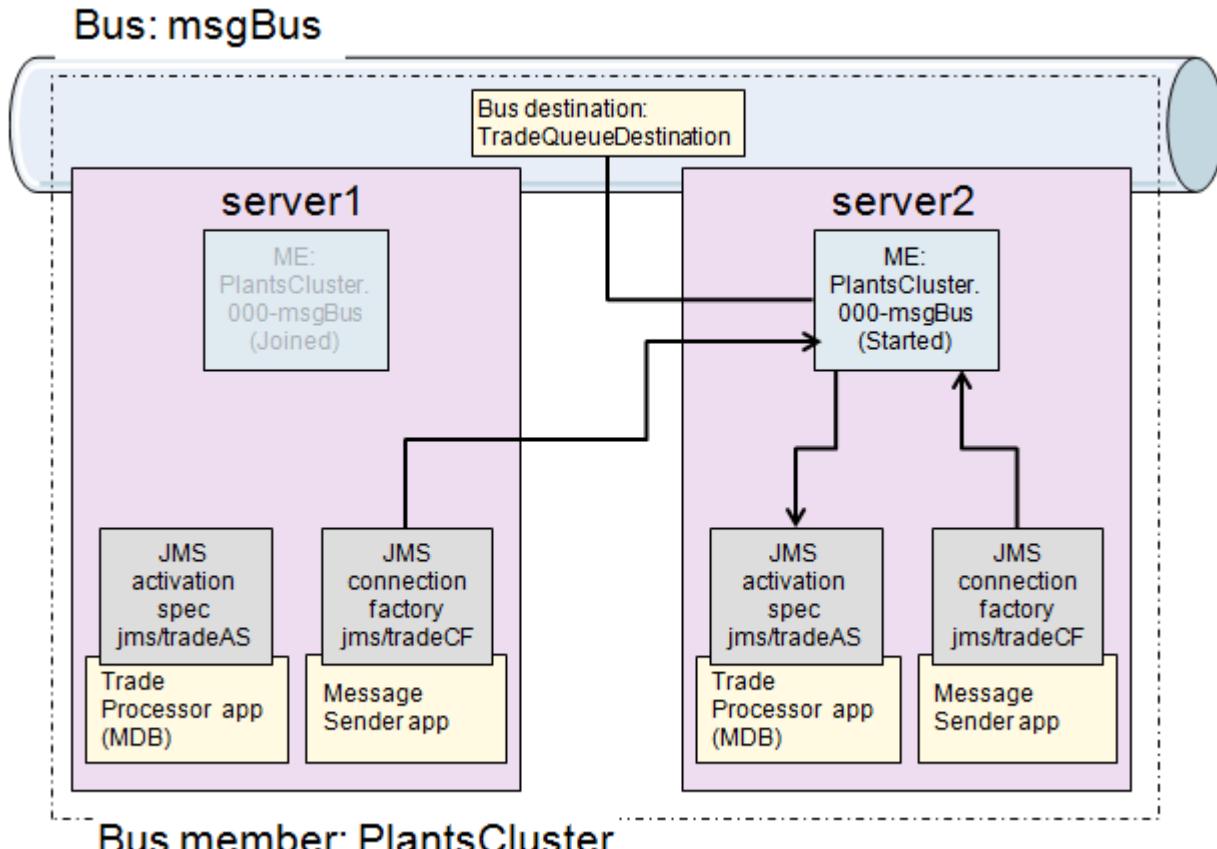
Messaging engine PlantsCluster.000-msgBus is in state Joined.

- \_\_\_ g. Close the browser in which the `Monitor.html` is running.
- \_\_\_ 6. Test messaging engine failover.
- \_\_\_ a. From the administrative console, stop server1.
- \_\_\_ b. Wait for the server to stop; then examine the logs for server2 and look for the following message near the end of the log.

8/25/11 15:51:06.795	00000036	SibMessage	INFO	[msgBus:PlantsCluster.000-msgBus] <a href="#">CWSID00161</a> : Messaging engine PlantsCluster.000-msgBus is in state Started.
-------------------------	----------	------------	------	-------------------------------------------------------------------------------------------------------------------------------

- \_\_\_ c. This message indicates that the messaging engine is now started on server2.
  - \_\_\_ d. From the administrative console, start server1.
  - \_\_\_ e. Wait for the server to start; then open a browser and from the home page, click the **Trade processor monitor** link.
- \_\_\_ 7. Now check which server is processing the messages as you did before.

As shown in the following diagram, no messages are sent to the MDB on server1, although the producer application instance that is running on server1 can still send messages.



- \_\_\_ a. From the upper left frame of the `Monitor.html` web page (sending messages from server: server1), leave the default values for Buy and Sell messages and click **Send messages**.
- \_\_\_ b. Observe which server processed the messages, server1 or server2.
- \_\_\_ c. Record the server name here: \_\_\_\_\_  
In the high availability policy, only one server in the cluster is running the messaging engine.
- \_\_\_ d. From the lower left frame of the `Monitor.html` web page (sending messages from server: server2), leave the default values (one message) for Buy and Sell messages and click **Send messages**.

- \_\_\_ e. Record the server name here: \_\_\_\_\_  
The same server is processing messages that are sent from both server1 and server2.
- \_\_\_ f. You can observe that after failover, when the original server is running again, the messaging engine continues to run on the server it failed over to. The high availability policy has failover but not “fail back”.
- \_\_\_ g. Close the browser in which Monitor.html is running.



### Information

#### High availability: Pros and cons

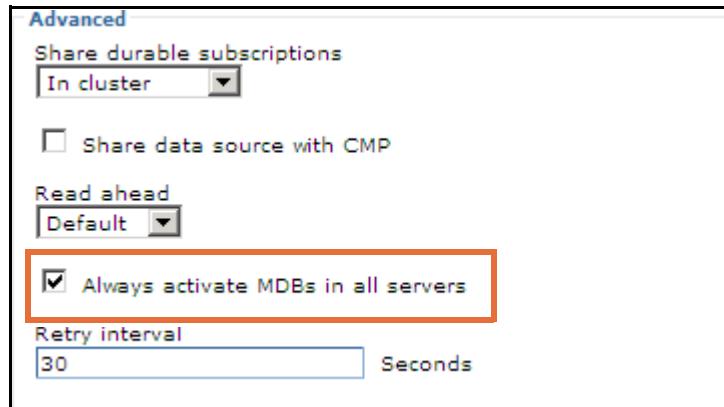
The fact that only one of the message consumers in a cluster gets to process messages might be just what you want. This configuration is called for when messages must be processed in the same sequence in which they were produced. (To ensure message order, more configuration might be required.)

If message order is not important, this configuration is not efficient in a clustered configuration. All servers in the cluster are running the message consumer application, but only one of those servers actually processes them.

There are at least two solutions to get more message consumers that process messages from the queue. In the next steps, you first configure the **Activation Specification** to allow all consumers in the cluster to get messages. Later, you change the messaging engine policy from high availability to scalability to provide workload management of the messaging engine.

- \_\_\_ 8. Configure the activation specification to start all MDBs in the cluster. There is an option to configure an activation specification to activate MDBs in all the servers, not just the server with the running messaging engine. The result of this configuring option is that all consumers in all cluster members receive messages.
  - \_\_\_ a. From the administrative console, click **Resources > JMS > Activation specifications**.
  - \_\_\_ b. Click the link for the **Trade Processor Activation Spec**.

- \_\_\_ c. Near the bottom of the page, in the Advanced section, select **Always activate MDBs in all servers**, and click **OK**.

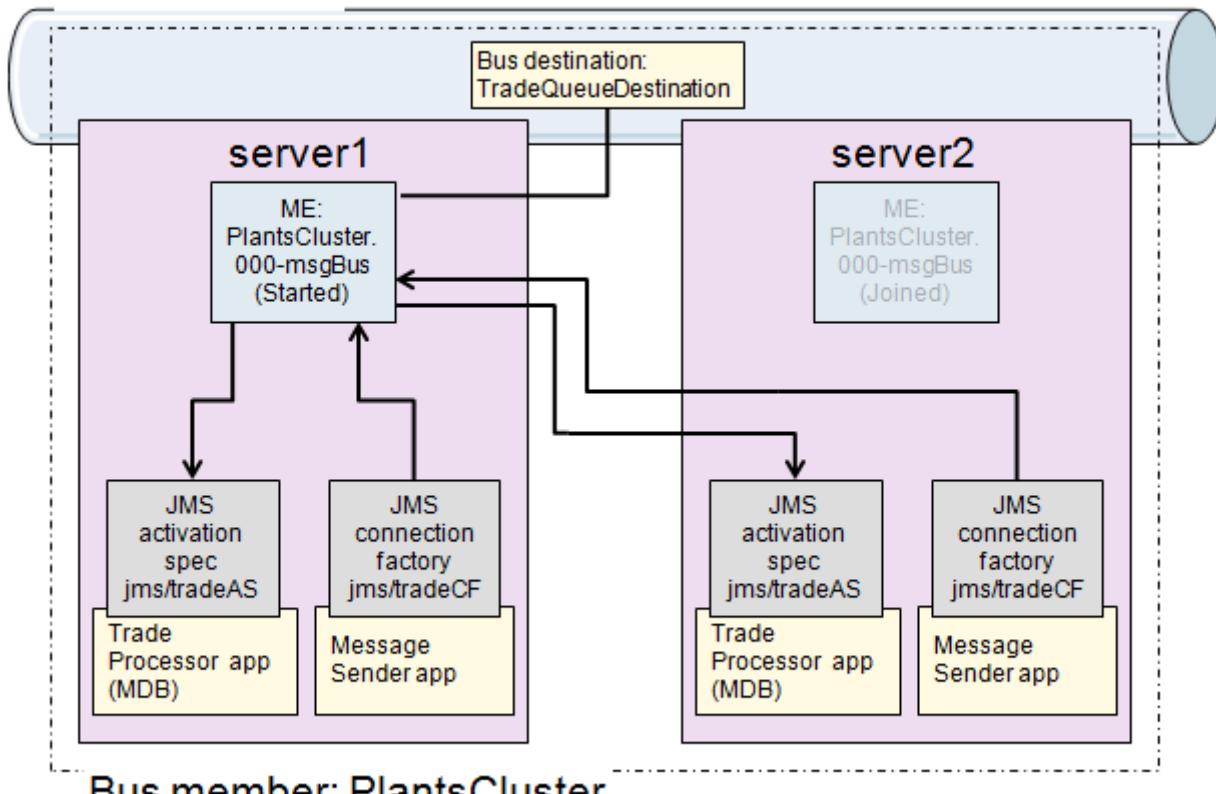


- \_\_\_ d. **Save** the changes to the master configuration.
- \_\_\_ 9. Ripplestart the PlantsCluster.
- \_\_\_ a. From the administrative console, click **Servers > Clusters > WebSphere application server clusters**.
- \_\_\_ b. Check the box for **PlantsCluster** and click **Ripplestart**.
- \_\_\_ c. Wait for the Status indicator to turn to a solid green arrow (it can take several minutes) signifying that all servers in the cluster are started. (It might be necessary to click the refresh icon several times.)
- \_\_\_ d. Verify that both servers are started by clicking **Servers > Server Types > WebSphere application servers**.
- \_\_\_ 10. Test whether MDBs are active in both servers that use the `Monitor.html` web page.

As shown in the following diagram, messages are now sent to the MDBs on both servers. The messaging engine is still running in only one server. The server where the messaging engine started depends on which server started first during the

Ripplestart. You can check by looking for the CWSID0016I message in the server logs. The diagram assumes that the messaging engine is started on server1.

## Bus: msgBus



## Bus member: PlantsCluster

- Start a new browser and enter the web address:  
<http://was85host:9080/Trade/processor/Monitor.html>

- \_\_\_ b. Make sure that both Trade processor applications have no pending transactions. You can quickly clear the table of pending transactions if necessary by selecting **Reset** from the Refresh menu.

Trade Requests To Be Processed On Server: Server1					
Account	Buy/Sell	Symbol	Qty	Total cost	Transaction

Refresh every:  seconds

- \_\_\_ c. Leave the default values of 1 Buy and 1 Sell messages, and click **Send messages** in the upper left frame.

### Sending messages from server: server1

Select number of messages to be sent for each of the **Buy** and **Sell** categories, then click **Send messages**.

Buy messages	<input type="button" value="1"/>
Sell messages	<input type="button" value="1"/>
<b>Send messages</b>	

### Trade Requests To Be Processed On Server: Server1

Refresh every:  seconds

Account	Buy/Sell	Symbol	Qty	Total cost	Transaction
23423234 (John Doe)	BUY	PG	10.0	\$657.90	server1 - 5

### Sending messages from server: server2

Select number of messages to be sent for each of the **Buy** and **Sell** categories, then click **Send messages**.

Buy messages	<input type="button" value="1"/>
Sell messages	<input type="button" value="1"/>

### Trade Requests To Be Processed On Server: Server2

Refresh every:  seconds

Account	Buy/Sell	Symbol	Qty	Total cost	Transaction
87652289 (Elaine Moose)	SELL	DELL	1.0	\$19.89	server1 - 6

- \_\_\_ d. Which processor application displayed the messages? You can see similar behavior in your own browser. Server 1 processed one message, the Buy message; and Server2 processed the other message, Sell. As a result of the new configuration of the activation specification, there is now an active MDB on each server that can consume messages.
- \_\_\_ e. Now from the lower left frame, send three **Buy** messages and two **Sell** messages.

**Sending messages from server: server2**

Select number of messages to be sent for each of the **Buy** and **Sell** categories, then click **Send messages**.

Buy messages	<input type="text" value="3"/>
Sell messages	<input type="text" value="2"/>
<b>Send messages</b>	

- \_\_\_ f. Notice the distribution of messages between the two processor applications. They each might, or might not, receive an equal number of messages.

## Trade Requests To Be Processed On Server: Server1

Refresh every:  seconds

Account	Buy/Sell	Symbol	Qty	Total cost	Transaction
23423234 (John Doe)	BUY	PG	10.0	\$657.90	server2 - 7
87652289 (Elaine Moose)	SELL	DELL	1.0	\$19.89	server2 - 9

## Trade Requests To Be Processed On Server: Server2

Refresh every:  seconds

Account	Buy/Sell	Symbol	Qty	Total cost	Transaction
23423234 (John Doe)	BUY	PG	10.0	\$657.90	server2 - 6
23423234 (John Doe)	BUY	PG	10.0	\$657.90	server2 - 8
87652289 (Elaine Moose)	SELL	DELL	1.0	\$19.89	server2 - 10

- \_\_\_ g. Try sending some more messages from the producers in both servers and see which consumer picks up the messages.  
 \_\_\_ h. Close the browser in which `Monitor.html` is running.



### Information

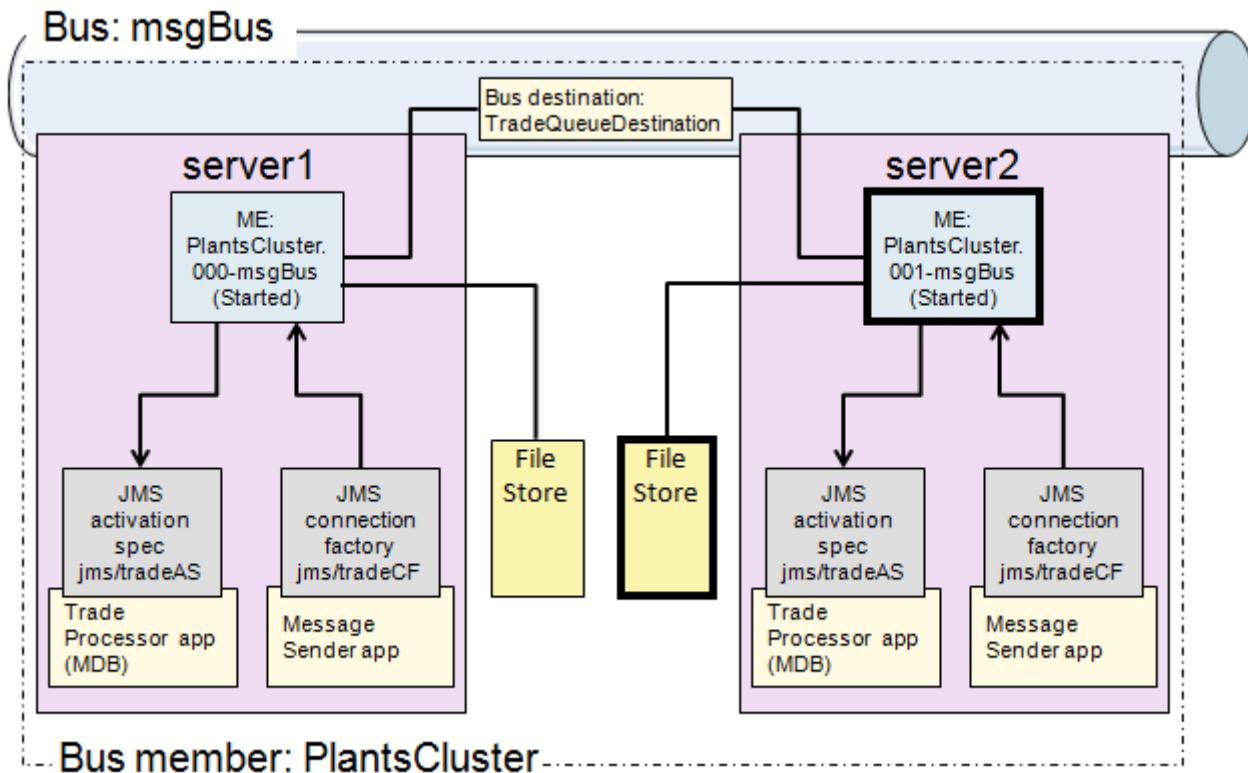
#### Possible performance issues

One of the problems with this activation specification configuration is that the single active messaging engine is now sending messages to queue points in multiple servers. In a local network with a small volume of messages, there might not be any performance issues. But in a widely distributed environment, with heavy volume of messages, it can cause performance problems because of the network traffic.

To provide better scalability, you can change the messaging engine policy to scalability, which requires a messaging engine in each cluster member.

## Section 10: Read only: Configure the scalability messaging engine policy

The steps for changing the messaging policy to scalability are described next. This configuration involves creating another messaging engine that runs on the second cluster member.

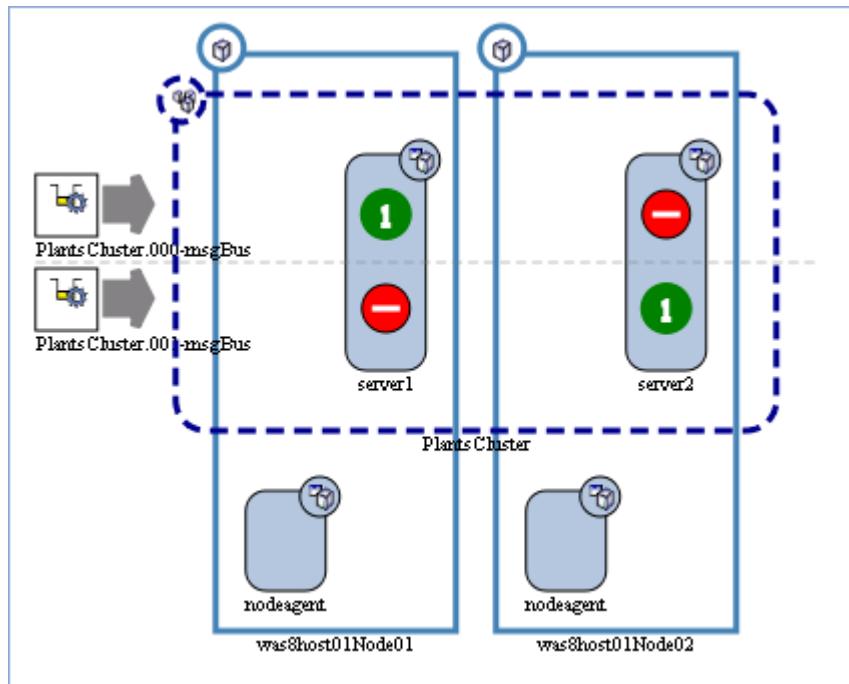


Two different messaging engines are run on their preferred server. The messaging engines do not fail over to another server if their preferred server is not running.

Use the messaging engine policy assistance wizard to configure the scalability policy.

General Properties		
Name	PlantsCluster	
Type	Cluster	
<input checked="" type="checkbox"/> Enable messaging engine policy assistance?		
Select	Policy type	Is further configuration required?
<input type="radio"/>	High availability	No
<input checked="" type="radio"/>	Scalability	<p><span style="color: yellow;">⚠</span> You need to add the following number of messaging engines: 1.</p> <p><span style="color: yellow;">⚠</span> You need to correct the following number of messaging engine policies: 1.</p>

The wizard guides you through adding the second messaging engine and correcting the policy for scalability. When the configuration is completed, the wizard displays the new scalability policy.



Interpreting the diagram, there are now two messaging engines, PlantsCluster.000-msgBus (000) and 001. Messaging engine 000 runs only on its preferred server, server1, and does not fail over to server2. Similarly, messaging engine 001 runs only on its preferred server, server2, and does not fail over to server1.

Recall that for the high availability policy, there was an option to enable **Always activate MDBs in all servers** on the activation specification. This option is no longer required for the scalability policy since this behavior is now automatic.

With this messaging engine policy, messages are consumed on the same server where they are produced. However, if you prefer to have messages that are consumed independently of where they are produced, you can configure settings on the queue definition. A feature in WebSphere Application Server is the capability to control the local queue point behavior. You might, or might not, prefer a local queue point. This choice

depends on the requirements of your application. This option is part of the JMS queue configuration.

**Message control across multiple queue points**

**Control across multiple queue points per MessageProducer**

Local queue point preference

Prefer to send messages to a local queue point  
 Do not prefer a local queue point over other queue points

Message affinity across queue points

Send all messages to the same queue point  
 Messages may be sent to different queue points

The default behavior is to prefer to send messages to a local queue point.

## **End of exercise**

## Exercise review and wrap-up

In this exercise, you configured the service integration bus and JMS resources as the Trade Processor and Message Sending Simulator applications required. This included configuring the bus members, bus destination, JMS connection factory, JMS queue, and JMS activation specification.



# Exercise 12.Configuring WebSphere security

## What this exercise is about

This lab configures access to the administrative console by defining a number of roles and mapping those roles to existing users. To test the configuration, you attempt to complete various functions from the different users and verify that the security configurations correctly limit access to various functions.

The last part of this lab defines fine-grained access from different parts of the WebSphere environment.

## What you should be able to do

At the end of this exercise, you should be able to:

- Enable WebSphere security
- Configure administrative security by configuring access to administrative functions
- Configure fine-grained administrative security

## Introduction

This exercise verifies that WebSphere administrative security is enabled. With administrative security turned on (the default is that it is enabled during profile creation), there are several effects. The effects include the fact that administrative tools such as the administrative console, wsadmin, and many of the scripts (including stopServer and serverStatus) require authentication and authorization to run. The exercise then examines the process of defining new administrative users and granting them specific access to parts of the administrative console and verifies that access is limited to certain functions. To verify, first log in to the administrative console and provide the user that was created during the profile creation. This specific user has, by default, implicit rights to the administrative console as it is the initial user that was created. This exercise creates more users and defines which rights they have within the administrative console.

Next, the exercise configures fine-grained access for both PlantsByWebSphere and DefaultApplication. Fine-grained access, a new console feature in WebSphere Application Server V7, is achieved by defining administrative authorization groups. These groups map

specific scopes or objects to console users and roles, thus allowing those users the access of that role to those specific objects. When console users attempt to access other objects for which fine-grained access is not configured, they have only the same. Finally, security domains are explored. Security domains allow the administrator to define security attributes at a scope other than just at the cell level. For example, with the use of security domains, it is possible to define a cell level user registry. You can then define another user registry for a specific cluster, node, or application server access role level that is defined for them at the global level.

## Requirements

This exercise requires a workstation with WebSphere Application Server V8.5 installed and the successful completion of the previous exercises.

## Exercise instructions

### Section 1: Resetting the WebSphere environment



#### Note

If your WebSphere environment must be reset for any reason, see **Appendix A** for instructions to correctly reset the environment.

### Section 2: Verify administrative security

This exercise configures security access to the administrative tools. Before any security access takes effect, administrative security must be enabled, which happens by default during the creation of a profile.

In this section of the lab, the state of administrative security is verified.

- 1. Check the state of administrative security.
  - a. Log in to the administrative console with `wasadmin` for the user name and `websphere` for the password.



#### Information

You already know the answer to whether administrative security is enabled. The fact that the administrative console prompted for a user name and password verifies that.

- b. Click **Security > Global security**.
- c. Verify that the **Enable administrative security** option is checked.



#### Information

If administrative security is not enabled, you must check the box, save your changes, and make sure that all nodes are synchronized. Then, restart all of the WebSphere processes (deployment manager, node agents, and application servers).

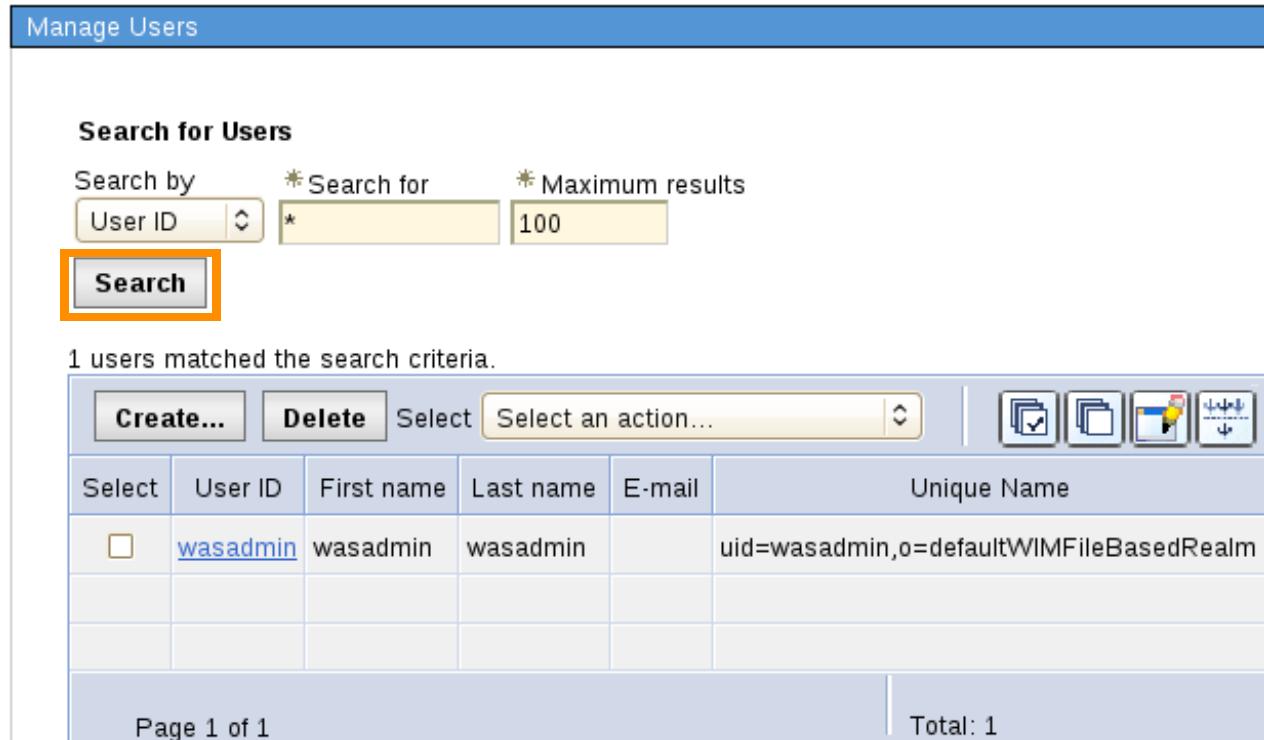
### Section 3: Defining WebSphere administrative console users

When WebSphere Application Server is installed and profiles are created, administrative security is enabled by default. Initially, the only user with access to the administrative console is the primary user that is specified during the profile creation, which is the `wasadmin` user. Initially, in the case of these labs, the only user that can access the

administrative console is wasadmin. In a real environment, it is desirable to have multiple administrative users and possibly have different rights for each user.

This section of the lab creates users and maps them to different levels of console access.

- \_\_ 1. Create WebSphere users for testing purposes.
  - \_\_ a. In the administrative console, expand **Users and Groups** and click **Manage Users**.
  - \_\_ b. Leave the defaults and click **Search**, which provides a list of current WebSphere administrative users.



The screenshot shows the 'Manage Users' interface with the title 'Search for Users'. It includes search fields for 'User ID' (set to '\*') and 'Maximum results' (set to 100). The 'Search' button is highlighted with an orange border. Below the search bar, a message says '1 users matched the search criteria.' A table lists one user: 'wasadmin' (User ID), 'wasadmin' (First name), 'wasadmin' (Last name), and 'uid=wasadmin,o=defaultWIMFileBasedRealm' (Unique Name). The table has columns for 'Select', 'User ID', 'First name', 'Last name', 'E-mail', and 'Unique Name'. At the bottom, it shows 'Page 1 of 1' and 'Total: 1'.

- \_\_ c. The `wasadmin` user was created during the profile creation (the `busUser` user is created in a previous exercise). To create more users for this lab, click **Create**.



### Information

Other users might exist, depending on which exercises you already completed.

- \_\_\_ d. Enter the User ID as: wasadm  
 Enter anything for the **First name** and **Last name**. Enter websphere for the **Password** and **Confirm password** fields.

**Create a User**

* User ID wasadm	Group Membership
* First name Joe	* Last name Admin
E-mail <input type="text"/>	
* Password *****	* Confirm password *****
<b>Create</b> <b>Cancel</b>	

- \_\_\_ e. Click **Create**.  
 \_\_\_ f. Click **Create Like** to create more users.  
 \_\_\_ g. Repeat this process for more user IDs of wascfg, wasmon, and wasoper. Again, enter anything for the **First name** and **Last name** fields, but use websphere for the passwords.  
 \_\_\_ h. Click **Close** when you create the last user.

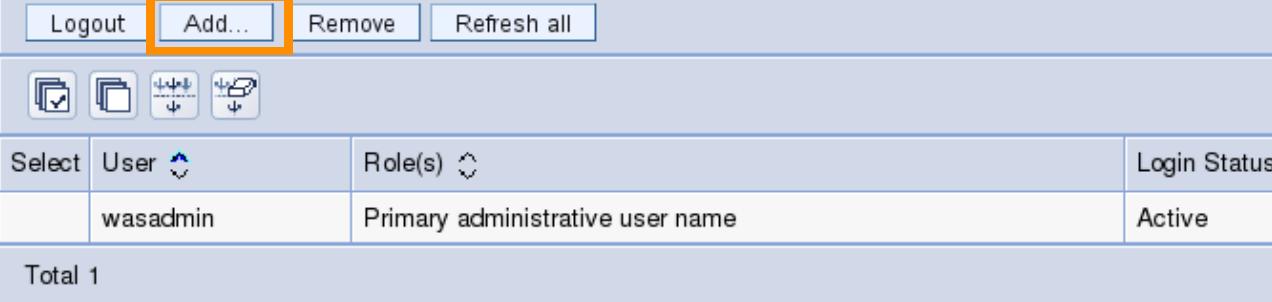


#### Information

These user IDs are being created inside of the WebSphere file-based user repository, not in the local operating system user registry or in LDAP.

- \_\_\_ 2. Map these new user IDs to their appropriate administrative console roles.  
 \_\_\_ a. Under Users and Groups, select **Administrative user roles**.

\_\_ b. Click Add.



Select	User	Role(s)	Login Status
	wasadmin	Primary administrative user name	Active

Total 1



### Information

Notice that the `wasadmin` entry listed is mapped to **Primary administrative user name**, which means the security user that is defined during the profile creation. As such, it has implicit mappings to all security roles.

\_\_ c. Click Search to view the list of available users.

- \_\_\_ d. Select wasadm from the Available column near the bottom of the screen and click the right arrow to move it to the **Mapped to role** column. Next, select the **Administrator** role near the top of the screen. Click **OK** to create the mapping between the wasadm user and the Administrator role.

**Administrative user roles**

**Administrative user roles > User**

Use this page to add, update or to remove administrative roles to users. Assigning administrative roles to users enables them to administer application servers through the administrative console or through wsadmin scripting.

\* Role(s)

Admin Security Manager	↑
<b>Administrator</b>	☰
Auditor	↓
Configurator	

**Search and Select Users**

Decide how many results to display, enter a search string (use \* for wildcard), and click Search. Select users from the Available list and add them to the Mapped to role list. Users which have already been mapped to a role will not be returned in the search results.

Search string: \*

Search

Maximum results to display: 20

Available	Mapped to role
wasadm	➡
wascfg	⬅
wasmon	
wasoper	

Select All   Deselect All   Select All   Deselect All

OK   Reset   Cancel

**Information**

Depending on what components you installed, it is possible for there to be more roles within the list.

Depending on which labs you completed, your list of users might be different from the screen capture.

- \_\_ e. Repeat these last two steps for `wasoper`, `wascfg`, and `wasmon` (mapping each user to the appropriate role). Examine the various roles.

<input type="button" value="Logout"/> <input type="button" value="Add..."/> <input type="button" value="Remove"/> <input type="button" value="Refresh all"/>				
Select	User	Role(s)	Login Status	
<input type="checkbox"/>	<a href="#">wasadm</a>	Administrator	Not Active	
	wasadmin	Primary administrative user name	Active	
<input type="checkbox"/>	<a href="#">wascfg</a>	Configurator	Not Active	
<input type="checkbox"/>	<a href="#">wasmon</a>	Monitor	Not Active	
<input type="checkbox"/>	<a href="#">wasoper</a>	Operator	Not Active	

Total 5

- \_\_ f. Click **System administration > Save changes to the master repository**.
- \_\_ g. Click **Save**.
- \_\_ h. Navigate to **System administration > Nodes** and select nodes **was85hostNode01** and **was85hostNode02**. Click **Full Resynchronize**.
- \_\_ i. Log out of the administrative console. Close the browser window.

**Information**

Changes to built-in, file-based repositories are not automatically replicated to managed nodes in a federated repositories configuration. You must use the administrative console to replicate the changes you make to a built-in, file-based repository. The full resynchronize operation resolves conflicts among configuration files and can take several minutes to complete.

Since these particular users are stored within the WebSphere configuration information, it is a good idea to do a full resynchronize with the nodes after creating new users.

- \_\_\_ 3. Optional: Go back and add a console user that is called `wassecmgr` and map it to the Admin Security Manager role.

This user can now be used to map security roles for console users. The only user that has this ability is `wasadmin` because it is the primary user (and therefore has this ability implicitly).

## **Section 4: Authenticate to the WebSphere administrative console and test mapped users**

In this part of the exercise, access to the administrative console is granted only to correctly mapped users. Depending on the role to which they are mapped, the administrative console allows those users to complete only certain functions.



### Information

By default, the user that is used to define the authentication mechanism has implicit access as an administrator role.

- \_\_\_ 1. Start a new browser and log in to the administrative console.
  - \_\_\_ a. Log in as: `wasadm`
- \_\_\_ 2. Verify that full access to administrative functions is available.
  - \_\_\_ a. Click **Applications > Application Types > WebSphere enterprise applications**.
  - \_\_\_ b. Notice that all standard functions are available.

Select	Name	Application Status
<input type="checkbox"/>	<a href="#">DefaultApplication</a>	
<input type="checkbox"/>	<a href="#">PlantsByWebSphere</a>	
<input type="checkbox"/>	<a href="#">ivtApp</a>	
<input type="checkbox"/>	<a href="#">query</a>	

Total 4

- \_\_\_ 3. Now verify the available functions for other users.

- \_\_ a. Log out of the administrative console.



- \_\_ b. Log back in to the administrative console as: wasoper  
 \_\_ c. Click **Enterprise applications** and notice what functions are available on the page for the role.

Select	Name	Application Status
<input type="checkbox"/>	<a href="#">DefaultApplication</a>	
<input type="checkbox"/>	<a href="#">PlantsByWebSphere</a>	
<input type="checkbox"/>	<a href="#">ivtApp</a>	
<input type="checkbox"/>	<a href="#">query</a>	

Total 4

- \_\_ d. Notice that not all the same functions are available. Now Start, Stop, and Rollout Update are only available.  
 \_\_ 4. Log in as wascfg and wasmon and examine what functions are available.  
 \_\_ 5. **Close** all web browser windows, which ensures that there are no existing session cookies when starting the next section of the exercise.

## Section 5: Enabling fine-grained control

Now that users with different types of access to the administrative console exist, it might be interesting to control the access more specifically. For example, in the following example the exercise creates two new administrative users. The first, PlantsAppAdmin, is configured to have rights only on the PlantsByWebSphere application. The second, DefaultAppAdmin, is configured to have rights only on the DefaultApplication.

By creating this setup, the exercise demonstrates how fine-grained access controls can be granted to administrative users. These types of controls can be granted on many different types of objects, not just applications.

The fine-grained access is defined by mapping administrative authorization groups to administrative console users. The administrative authorization groups point at specific scopes or objects. When an administrative user attempts to access an object and does not have global access, the access that the administrative authorization groups define for the object is checked.

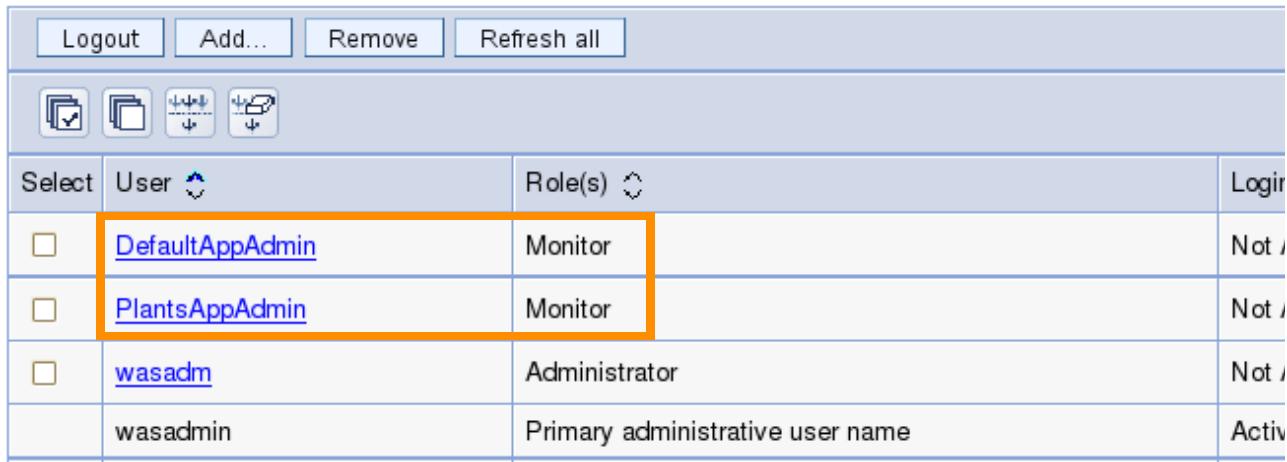
The user with fine-grained administrative access requires a minimum of global Monitor access.

- \_\_\_ 1. Create more users in the repository. As done in the previous section of this exercise, create two new users: `PlantsAppAdmin` and `DefaultAppAdmin`. Map these two new users to the monitor role.
- \_\_\_ a. Make sure that you are logged in to the administrative console with a user that gives you **Admin Security Manager** role access like `wasadmin`. The role provides permission to map console roles for console users and groups. Since `wasadmin` is the primary user, it has Admin Security Manager access implicitly.



- \_\_\_ b. In the administrative console, expand **Users and Groups** in the left navigation menu and select **Manage Users**.
- \_\_\_ c. Leave the defaults and click **Search**. The list of current WebSphere administrative users is displayed.
- \_\_\_ d. Click **Create**.
- \_\_\_ e. Enter the User ID: `PlantsAppAdmin`  
Enter anything for the **First name** and **Last name**. Enter `websphere` for the **Password** and **Confirm password**. Next, click **Create**.
- \_\_\_ f. Repeat this process to create the `DefaultAppAdmin` user.
- \_\_\_ 2. Map the two new users to the global monitor role. Any console user or group that is used for fine-grained access requires a minimum role mapping of monitor.
- \_\_\_ a. Under **Users and Groups** in the administrative console, select **Administrative user roles**. Click **Add**.

- \_\_ b. Click **Search** to view the list of available users. Select both **PlantsAppAdmin** and **DefaultAppAdmin** (hold down the Ctrl key to multi-select) from the Available column and click the **right arrow** to move them to the **Mapped to role** column. Select the **Monitor** role near the top of the screen and click **OK**.



The screenshot shows a user administration interface with the following columns: Select, User, Role(s), and Login status. Two users are selected and highlighted with an orange border: 'DefaultAppAdmin' and 'PlantsAppAdmin', both assigned to the 'Monitor' role. Other users listed are 'wasadm' (Administrator role) and 'wasadmin' (Primary administrative user name).

Select	User	Role(s)	Login
<input type="checkbox"/>	<a href="#">DefaultAppAdmin</a>	Monitor	Not Active
<input type="checkbox"/>	<a href="#">PlantsAppAdmin</a>	Monitor	Not Active
<input type="checkbox"/>	<a href="#">wasadm</a>	Administrator	Not Active
	wasadmin	Primary administrative user name	Active

- \_\_ c. **Save** the changes. Wait for the nodes to synchronize and click **OK**.
- \_\_ 3. Create the administrative authorization groups for the PlantsAppGroup and DefaultAppGroup.
- \_\_ a. In the administrative console, click **Security > Administrative Authorization Groups**.



- \_\_ b. Click **New** to create the authorization group.



- \_\_ c. Enter `PlantsAppGroup` for the **Name**.

- \_\_\_ d. Under Resources, expand all the entries and the subentries. Take note of all of the different levels that can be defined in an administrative authorization group. Expand **Applications**; select the box next to **PlantsByWebSphere**.

The screenshot shows the 'Administrative authorization groups' dialog box. The 'General Properties' tab is selected, displaying a field for 'Name' containing 'PlantsAppGroup', which is highlighted with an orange border. The 'Resources' tab is also visible, showing a tree view of resources under 'All scopes'. The 'Applications' node is expanded, revealing sub-applications like 'query', 'ivtApp', 'DefaultApplication', and 'PlantsByWebSphere', with the latter also highlighted with an orange border. At the bottom of the dialog are four buttons: 'Apply', 'OK', 'Reset', and 'Cancel'.

- \_\_\_ e. Click **Apply**.

- \_\_ f. On the right, under Additional Properties, click **Administrative user roles**.

#### Additional Properties

- [Administrative group roles](#)
- [Administrative user roles](#)



#### Information

In some cases, the Additional Properties are not rendered, and a browser refresh does not seem to solve the problem. In such cases, try the following steps:

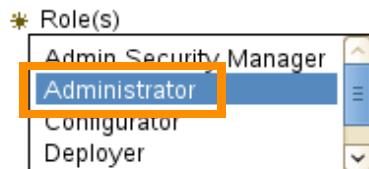
- Save the changes, navigate back through **Security > Administrative Authorization Groups**, and click **PlantsAppGroup**.
- **Close** the console browser and start a new console window in a new browser.

- \_\_ g. Click **Add** to map the administrative console `PlantsAppAdmin` user.

- \_\_\_ h. Select the **Administrator** role and then click **Search** to show all known users. Select `PlantsAppAdmin` and then click the **right arrow** to move the user from the **Available** column to the **Mapped to role** column.

[Administrative authorization groups > PlantsAppGroup > Administrative user roles > User](#)

Use this page to add, update or to remove administrative roles to users. Assigning administrative roles to users enables them to administer application servers through the administrative console or through wsadmin scripting.



[Search and Select Users](#)

Decide how many results to display, enter a search string (use \* for wildcard), and click **Search**. Select users from the Available list and add them to the Mapped to role list. Users which have already been mapped to a role will not be returned in the search results.

Available	Mapped to role
DefaultAppAdmin	
PlantsAppAdmin	
wasadm	
wascfg	
wasmon	
wasoper	
wassecmgr	

Available: DefaultAppAdmin, PlantsAppAdmin, wasadm, wascfg, wasmon, wasoper, wassecmgr  
Mapped to role: (empty)

Search string: \*

Maximum results to display: 20

Search button (highlighted)

Available button: Select All | Deselect All

Mapped to role button: Select All | Deselect All

OK | Reset | Cancel

- \_\_\_ i. Click **OK**.
- \_\_\_ j. **Save** the changes and make sure that the nodes are synchronized.
- \_\_\_ k. Repeat these steps to create the `DefaultAppGroup`, scoped to the `DefaultApplication`, and map `DefaultAppAdmin`. To begin, click **Security > Administrative Authorization Groups** and click **New**.

- I. **Save** the changes. Wait for the nodes to synchronize and click **OK**. The final list looks like the following screen capture:

Select	Name
<input type="checkbox"/>	<a href="#">DefaultAppGroup</a>
<input type="checkbox"/>	<a href="#">PlantsAppGroup</a>

Total 2



### Information

Two new administrative users are called `PlantsAppAdmin` and `DefaultAppAdmin`. Both have Monitor access to the cell, which means that by using the console they can view the contents of the cell, but they cannot modify anything.

Additionally, two new administrative authorization groups are called `PlantsAppGroup` and `DefaultAppGroup`, scoped at the `PlantsByWebSphere` and `DefaultApplication` applications. The groups define what administrative rights might be available for those two applications.

Finally, the `PlantsAppAdmin` user is mapped to `PlantsAppGroup`, and the `DefaultAppAdmin` user is mapped to `DefaultAppGroup`.

The result is that these two new users have Monitor access to everything in the cell. Additionally, they also have administrative rights, but only to their specific applications.

## Section 6: Test the fine-grained access

Now that the new administrative console users are created, and the administrative authorization groups are added and mapped to the applications, access by the users to the applications must be verified.

- 1. Open a new administrative console window and log in as: PlantsAppAdmin



2. After you log in, browse through various parts of the console. Notice that the PlantsAppAdmin user has **Monitor** rights to most areas. But also notice that the PlantsAppAdmin user has complete **Administrator** rights only to the PlantsByWebSphere enterprise application.

The screenshot shows the WebSphere administrative console interface. On the left, there is a navigation sidebar with the following items:

- Welcome
- Guided Activities
- Servers
- Applications** (selected)
  - All applications
  - Application Types**
    - WebSphere enterprise applications
    - Business-level applications
    - Assets
  - Edition Control Center
  - Global deployment settings
- Jobs
- Services
- Resources
- Runtime Operations
- Security
- Operational policies

The main content area is titled "Enterprise Applications". It displays the following information:

- Enterprise Applications**: A section header with the sub-instruction: "Use this page to manage installed applications. A single".
  - Preferences**
- Buttons: Start, Stop, Update, Rollout Update, and a magnifying glass icon.
- Resource management section: "Select Name" with a dropdown arrow. Below it, a table lists resources:
 

You can administer the following resources:	
<input type="checkbox"/>	<a href="#">PlantsByWebSphere</a>
- Monitoring section: "You can monitor the following resources:" with a table listing resources:
 

You can monitor the following resources:	
	<a href="#">DefaultApplication</a>
	<a href="#">ivtApp</a>
	<a href="#">query</a>
- Total count: Total 4

3. Log out of the administrative console, and log in again as: DefaultAppAdmin

4. Again, browse through various parts of the administrative console and notice that this user has **Monitor** access only. Navigate to the enterprise application list and notice that this user has administrative access to the DefaultApplication, but not to anything else.

The screenshot shows the WebSphere Application Server V8.5.5 Administration console. The left sidebar has a 'View' dropdown set to 'All tasks' and lists various navigation options: Welcome, Guided Activities, Servers, Applications (expanded), All applications, Application Types (WebSphere enterprise applications, Business-level applications, Assets), Edition Control Center, Global deployment settings, Jobs, Services, Resources, Runtime Operations, Security, and Operational policies. The main panel title is 'Enterprise Applications' with the sub-section 'Enterprise Applications'. It says 'Use this page to manage installed applications. A single click starts or stops an application.' Below are buttons for Start, Stop, Update, and Rollout Update, and icons for creating, deleting, and modifying applications. A 'Select' dropdown is next to a 'Name' search input field. A section titled 'You can administer the following resources:' contains a checkbox next to 'DefaultApplication', which is highlighted with an orange rectangle. Another section titled 'You can monitor the following resources:' lists 'PlantsByWebSphere', 'ivtApp', and 'query'. At the bottom, it says 'Total 4'.

## Section 7: Examine security domains

In the previous section, fine-grained access control was configured, thus allowing administrators the ability to have better control over which administrators have access to which portions of a cell.

Security domains allow an administrator to define alternative security configurations for a cell. Historically, security configurations were defined only at a cell level, which meant that if something such as a user registry was defined within a cell, it applied to the whole cell. There was no ability to define an alternative configuration for specific sections of a cell.

This section looks at how to use security domains to create alternative security configurations.

1. Define a new security domain.

- \_\_ a. Use the `wasadmin` account to access the administrative console.
- \_\_ b. Click **Security > Security domains**.



- \_\_ c. Click **New** to create a security domain.
- \_\_ d. Enter the name: `PlantsSecurityDomain`

The screenshot shows a dialog box titled 'Security domains > New...'. It has a field labeled 'Name' with the value 'PlantsSecurityDomain' (which is highlighted with an orange border). Below it is a 'Description' field containing an empty text area. At the bottom of the dialog are four buttons: 'Apply', 'OK' (which is highlighted with a yellow background), 'Reset', and 'Cancel'.

- \_\_ e. Click **OK**.

- \_\_ f. Click **PlantsSecurityDomain** to access the details.

The screenshot shows a table with columns for 'Select', 'Name', and 'Description'. A message above the table says 'You can administer the following resources:'. In the 'Name' column, there is one row containing a checkbox and the text 'PlantsSecurityDomain', which is highlighted with an orange rectangular box. Below the table, it says 'Total 1'.

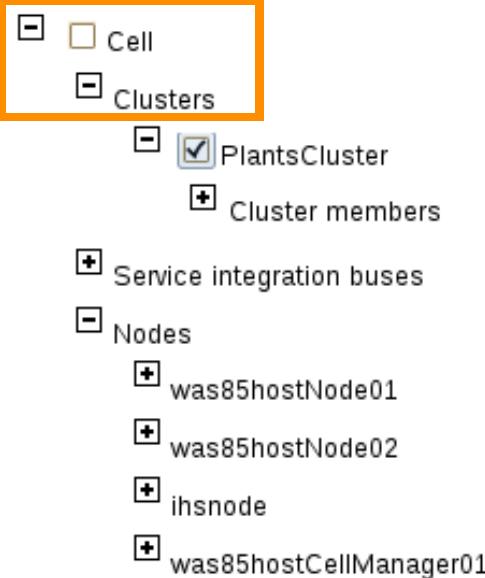
- \_\_ 2. Define the security domain details by associating a scope with specific security attributes.
- \_\_ a. There are two main sections on this screen. The first is the Assigned Scopes. Expand **Cell** and then **Clusters**.

#### Assigned Scopes

Assign the security domain to the entire cell or select the specific servers, clusters, and service integration buses to include in this security domain.

Show:

All scopes ▾





## Information

Security domains allow you to assign security configurations to defined portions of a cell. By assigning the scope, you are defining which portion of the cell is affected.

- \_\_\_ b. Select **PlantsCluster**.
- \_\_\_ c. Continue to explore the other available scopes.
- \_\_\_ d. Move down to the Security Attributes section of the screen and explore the various security attributes that are available to associate with the previously selected scope.
- \_\_\_ e. Expand the **Application Security** section. Notice that it is possible to enable Application Security to only the selected scope (in this case, server1).

**Security Attributes**

The screenshot shows the 'Security Attributes' section of a configuration interface. The 'Application Security' section is expanded, revealing two options: 'Use global security settings' (selected) and 'Customize for this domain'. The 'Customize for this domain' option has its子选项 'Enable application security' checked. Other collapsed sections include 'Java 2 Security', 'User Realm', and 'Trust Association', all currently disabled.

- Application Security:** Disabled
  - Use global security settings  
Do not enable application security
  - Customize for this domain  
 Enable application security
- Java 2 Security:** Disabled
- User Realm:** Administrative realm
- Trust Association:** Disabled

- \_\_ f. Expand the **Java 2 Security** section. Notice that it is possible to enable Java 2 Security to only the selected scope.

#### Security Attributes

**Application Security:** Disabled

**Java 2 Security:** Disabled

Use global security settings

Do not use Java 2 security to restrict application access to local resources

Customize for this domain

Use Java 2 security to restrict application access to local resources

Warn if applications are granted custom permissions

Restrict access to resource authentication data

**User Realm:** Administrative realm

**Trust Association:** Disabled

- \_\_ g. Expand the **User Realm** section. Notice that it is possible to define another user registry for only selected scopes.

#### Security Attributes

**Application Security:** Disabled

**Java 2 Security:** Disabled

**User Realm:** Administrative realm

Use global security settings

Repository type: Federated repositories

Customize for this domain

Realm type

Federated repositories



[Configure...](#)

**Trust Association:** Disabled

- \_\_ 3. Cancel the security domain settings. The intention of this section was to show what some of the possibilities are for security domains but not actually create one.
- \_\_ a. Click **Cancel** if available.
- \_\_ b. At the top of the next page, you have the opportunity to save the changes. Instead, click **Review**.

- \_\_\_ c. To ensure that no changes are saved, click **Discard** and click **Yes** to confirm.
  - \_\_\_ d. Finally, click **OK** to complete the process of dismissing the changes.
- \_\_\_ 4. Log out of any administrative console windows and close all browsers.

## **End of exercise**

## Exercise review and wrap-up

The exercise looked at setting up security for accessing the administrative console, which is done by creating new administrative console users and mapping them to global access roles. Then, two new console users are mapped to administrative authorization groups to create fine-grained access to the PlantsByWebSphere application and the DefaultApplication.

Finally, the process of creating security domains was explored.

# Exercise 13.Configuring application security

## What this exercise is about

This exercise uses WebSphere application security to secure the administration module of the PlantsByWebSphere application. The application is tested with application security enabled. An explanation, through exploration of configuring application security in IBM Assembly and Deploy Tools, is also provided.

## What you should be able to do

At the end of the exercise, you should be able to:

- Define Java EE security roles
- Define access for resources in an application
- Enable and verify application security

## Introduction

This lab deals with configuring application security for the PlantsByWebSphere application by using the administrative console and the IBM Assembly and Deploy Tools.

## Requirements

This exercise requires a workstation with WebSphere Application Server V8.5 installed and completion of previous exercises.

## **Exercise instructions**

### **Section 1: Resetting the WebSphere environment**



#### **Note**

If your WebSphere environment must be reset for any reason, see **Appendix A** for instructions to correctly reset the environment.

### **Section 2: Enabling application security**

In a previous exercise, administrative security was configured. This exercise enables and configures application security. The application security allows WebSphere to provide authentication and authorization for the enterprise applications. So, unlike administrative security (which secures the administrative interfaces), application security controls who has access to which parts of the enterprise applications that are run within the application servers.

In this section of the lab, WebSphere application security is enabled through the administrative console.

- \_\_ 1. Configure the application security setting.
  - \_\_ a. Make sure that you are logged in to the administrative console with a user that gives you administrator privileges, such as: `wasadmin` or `wasadm`
  - \_\_ b. Click **Security > Global security**.

- c. Check the box next to **Enable application security** and click **Apply**.

The screenshot shows the 'Global security' configuration panel. At the top, there are two buttons: 'Security Configuration Wizard' and 'Security Configuration Report'. Below them, under the heading 'Administrative security', there is a checked checkbox for 'Enable administrative security' followed by three links: 'Administrative user roles', 'Administrative group roles', and 'Administrative authentication'. Under the heading 'Application security', there is a checked checkbox for 'Enable application security'. Finally, under the heading 'Java 2 security', there are three unchecked checkboxes: 'Use Java 2 security to restrict application access to local resources', 'Warn if applications are granted custom permissions', and 'Restrict access to resource authentication data'.



### Information

WebSphere administration security is already enabled during profile creation. All that is being done is the enabling of application security. As such, it is not necessary to define user registries or authentication mechanisms.

- d. **Save** the changes and make sure to synchronize the nodes.  
— 2. Restart all application server processes.  
— a. Click **Servers > Clusters > WebSphere application server clusters**.

- \_\_ b. Select the **PlantsCluster** and click **Ripplestart**.

The screenshot shows the WebSphere Application Server Administration console. At the top, there is a toolbar with buttons for New..., Delete, Start, Stop, Ripplestart (which is highlighted with an orange box), and ImmediateStop. Below the toolbar is a row of icons: a folder with a checkmark, a folder with a question mark, a plus sign, and a minus sign. The main area has a table with columns for Select, Name, and Status. A checkbox next to 'PlantsCluster' is checked and highlighted with an orange box. The status column shows a green circular icon with a white arrow. At the bottom left, it says 'Total 1'.



### Information

In older versions of WebSphere, all processes within the cell required a restart. Because global security was already enabled at profile creation time, the inclusion of application security only required the restarting of the application servers.

## Section 3: Securing the *PlantsByWebSphere* application

When running with application security enabled, enterprise applications can take advantage of role-based application security to restrict access to servlet and EJB resources. The *PlantsByWebSphere* application administration module is already configured to take advantage of application security by having a security role that is called SampAdmin and mapping to the administration module. All that the administrator is still required to do is to map the SampAdmin security role to the users or groups that exist in the runtime environment.



### Information

**Java 2 security** can also be used to provide fine-grained access to system resources, such as ports or sockets. Java 2 security is orthogonal to Java Platform, Enterprise Edition or Java EE security and does not require the enforcement of administrative security. In this exercise, you do not use Java 2 security.

- \_\_ 1. Create a user named **PlantsUser** to use for application authentication.
  - \_\_ a. Click **Users and Groups** and click **Manage Users**.
  - \_\_ b. Leave the defaults and click **Search**, which displays the list of current administrative users.
  - \_\_ c. Click **Create**.

- \_\_\_ d. Enter the User ID: PlantsUser  
 Enter anything for the **First name** and **Last name**. Enter web1sphere for the **Password** and **Confirm password** fields and click **Create**.

**Create a User**

\* User ID  
PlantsUser

\* First name  
Plants

\* Last name  
User

E-mail

\* Password  
\*\*\*\*\*

\* Confirm password  
\*\*\*\*\*

**Create** **Cancel**

- \_\_\_ e. Click **Close**.
- \_\_\_ 2. Test the application before mapping the roles to users and groups.
- \_\_\_ a. Close all of your current browser windows.
- \_\_\_ b. With all the servers up and running, use a new browser to access the admin servlet by going to the following address:

<http://was85host:9080/PlantsByWebSphere/admin.html>

You can also access this servlet by going to the PlantsByWebSphere home page and clicking the **Help** link. From there, click **Admin Home**.

**PLANTS BY WEBSPHERE**

Your shopping cart is currently empty

Flowers Fruits & Vegetables Trees Accessories

HOME : SHOPPING CART : LOGIN : **HELP**

Gardens of Summer

[View Server Info](#)

[Admin Home](#)



Flowers : Fruits & Vegetables : Trees : Accessories : Home : Shopping  
Account : Login : Help



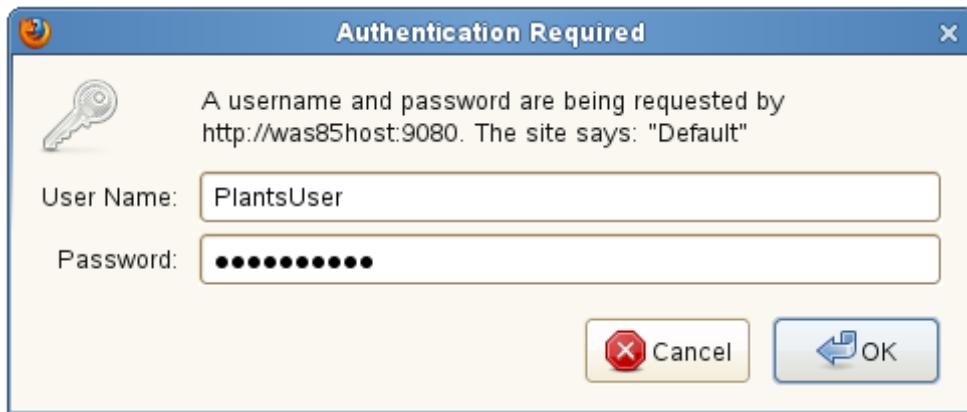
### Important

Existing browsers might already have authentication information for connection to the administrative console, which can interfere with attempts to log in to the PlantsByWebSphere application.

To solve this problem, either close all current browser windows or clear the current browser information. To clear any authentication in a Firefox window, click **Tools > Clear Recent History** and click **Clear Now**.

Other solutions include the use of a second browser type or configuring Firefox profiles.

- \_\_\_ c. Attempt to log in using `PlantsUser` with `web1sphere` as the password.



- \_\_\_ d. Notice that you are not authorized to access this page because the user PlantsUser is not granted access to the PlantsByWebSphere application. Some browsers repeat the authentication request if authorization fails.



#### Information

Only the administration part of the application is enabled for security. The rest of the application can be accessed just as before application security was enabled.

- \_\_\_ 3. Create a registry group and user that can be mapped to the PlantsByWebSphere application.



#### Information

It would be easiest to map the application security role to a list of users. But it is a much better practice to use a group instead.

- \_\_\_ a. Log back in to a new administrative console window and expand **Users and Groups**.  
\_\_\_ b. Click **Manage Groups**.

- \_\_ c. Click **Search**, which displays the list of current WebSphere administrative groups.

The screenshot shows the 'Manage Groups' interface with a search form. The search criteria are set to 'Group name' with a value of '\*' and 'Maximum results' set to 100. The 'Search' button is highlighted with an orange box. Below the search form, a message states '0 groups matched the search criteria.' A table below contains two buttons: 'Create...' and 'Delete'. At the bottom, it shows 'Page 1 of 1' and 'Total: 0'.

- \_\_ d. Click **Create**.  
\_\_ e. Enter PlantsGroup for the **Group name** and anything for the **Description**. Click **Create**.

The screenshot shows the 'Create a Group' dialog. The 'Group name' field is filled with 'PlantsGroup' and is highlighted with an orange box. Below it is a large 'Description' text area. At the bottom, there are two buttons: 'Create' and 'Cancel', with 'Create' highlighted with an orange box.

- \_\_ f. Click **Close**.

\_\_ g. Click **PlantsGroup**.

Select	Group name	Description	Unique Name
<input type="checkbox"/>	<a href="#">PlantsGroup</a>		cn=PlantsGroup,o=defaultWIMFileBasedRealm

Page 1 of 1      Total: 1

\_\_ h. Click the **Members** tab.

Manage Groups

Group Properties

General Members Groups

\* Group name  
PlantsGroup

Description

OK Apply Cancel

\_\_ i. Click **Add Users**.

- \_\_ j. On the next page, click **Search**. The result shows the list of known users.

Manage Groups

Add Users to a Group

Group name  
PlantsGroup

Search for users that will be members of this group.

Search by \* Search for \* Maximum results

User ID \* 100

**Search**

9 users matched the search criteria.

DefaultAppAdmin  
PlantsAppAdmin  
PlantsUser  
wasadm  
wasadmin  
wascfg  
wasmon  
wasoper  
wassecmgr

Add Close

- \_\_ k. Select **PlantsUser** and click **Add**.

- \_\_ l. Click **Close** to verify that the **PlantsUser** is added to **PlantsGroup**.



### Information

You created a user that is called **PlantsUser**, and this user is added to the newly created **PlantsGroup**. Next, you map the **PlantsGroup** to the **PlantsByWebSphere** application, thus granting any members of the **PlantsGroup** access to the restricted parts of the **Plants** application.

- \_\_\_ 4. Map users and groups to Java EE security roles defined within the enterprise application. The SampleAdmin role exists inside of PlantsByWebSphere, along with mappings to various methods.
  - \_\_\_ a. Click **Applications > Application Types > WebSphere enterprise applications**.
  - \_\_\_ b. Click the **PlantsByWebSphere** application link.
  - \_\_\_ c. Under Detail Properties, click **Security role to user/group mapping**. No users or groups are mapped to the `SampAdmin` security role.

**Enterprise Applications**

**Enterprise Applications > PlantsByWebSphere > Security role to user/group mapping**

Security role to user/group mapping

Each role that is defined in the application or module must map to a user or group from the domain user registry. `accessIds`: The accessIds are required only when using cross realm communication in a multi domain scenario. For all other scenarios the accessId will be determined during the application start based on the user or group name. The accessIds represent the user and group information that is used for Java Platform, Enterprise Edition authorization when using the WebSphere default authorization engine. The format for the accessIds is `user:realm/uniqueUserId`, `group:realm/uniqueGroupId`. Entering wrong information in these fields will cause authorization to fail.

`AllAuthenticatedInTrustedRealms`: This indicates that any valid user in the trusted realms be given the access. `AllAuthenticated`: This indicates that any valid user in the current realm be given the access.

		Map Users...	Map Groups...	Map Special Subjects ▾
Select	Role	Special subjects	Mapped users	Mapped groups
<input type="checkbox"/>	SampAdmin	None		



### Information

There are four types of users: Everyone, All authenticated, Mapped users, and Mapped groups. The first two do not apply to this exercise as you do not want Everyone to access the administration module; and since there is no previous opportunity to authenticate, that rules out All authenticated.

The mapped entries point to users and groups in the current user registry.

- \_\_ d. Select the box next to the **SampAdmin** role and click **Map Groups**.

Map Users...	Map Groups...	Map Special Subjects ▾		
Select	Role	Special subjects	Mapped users	Mapped groups
<input checked="" type="checkbox"/>	SampAdmin	None		

- \_\_ e. Make sure that the Search String is \* and click **Search**. The Available list fills up with the group defined within the WebSphere user registry.

- \_\_\_ f. Select **PlantsGroup** and click the right arrow to move the entry to the Selected list.

Enterprise Applications

[Enterprise Applications > PlantsByWebSphere > Security role to user/group](#)

Use this page to search for users or groups and add them to the selected roles.

■ SampAdmin

**Search and Select Groups**

Decide how many results to display, enter a search string (use \* for wildcard), or add them to the Mapped to role list.

Display a maximum of  results

Search string

Available:  PlantsGroup

Selected:

- \_\_ g. Click **OK**.

		Map Special Subjects ▾		
Select	Role	Special subjects	Mapped users	Mapped groups
<input type="checkbox"/>	SampAdmin	None		PlantsGroup

- \_\_ h. Click **OK** again. The changes do not get set until the second OK.

- \_\_ i. **Save** the changes.

- \_\_ 5. Ensure that the nodes are fully synchronized.

- \_\_ a. Click **System administration > Nodes**.

- \_\_ b. Select both the nodes in the cluster and click **Full Resynchronize**.

Add Node	Remove Node	Force Delete	Synchronize	Full Resynchronize	Stop
Select	Name ↗	Host Name ↘	Version ↘	Discovery Pro	
You can administer the following resources:					
<input type="checkbox"/>	ihsnode	was85host	Not applicable	TCP	
<input type="checkbox"/>	<a href="#">was85hostCellManager01</a>	was85host	ND 8.5.0.0	TCP	
<input checked="" type="checkbox"/>	<a href="#">was85hostNode01</a>	was85host	ND 8.5.0.0	TCP	
<input checked="" type="checkbox"/>	<a href="#">was85hostNode02</a>	was85host	ND 8.5.0.0	TCP	
Total 4					



### Information

Remember, changes to built-in, file-based repositories are not automatically replicated to managed nodes in a federated repositories configuration. You must use the administrative console to replicate the changes you make to a built-in, file-based repository. The full resynchronize operation resolves conflicts among configuration files.

- \_\_ 6. Close all web browser windows. Open a new web browser to access the admin servlet.
- Close the existing browsers.
  - Open a new browser (or clear all state information for the existing browser).



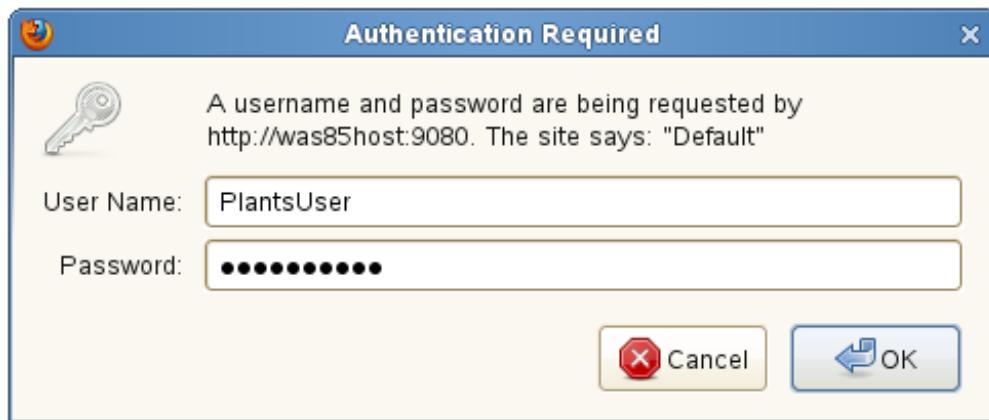
## Information

It is also possible to clear any authentication in the browser. In a Firefox window, click **Tools > Clear Recent History > Clear Now**.

- c. Access the PlantsByWebSphere admin servlet page through the **Help > Admin Home** link or though the following URL:

`http://was85host:9080/PlantsByWebSphere/admin.html`

- d. Log in using `PlantsUser` as the user name and `websphere` as the password.



## Warning

By default, the application server has a 10-minute authentication cache timeout. Therefore, if information is still cached, it might not time out for up to 10 minutes. If the timeout is a problem, you can either wait until the timeout happens or restart the application server.

To view your security authentication timeout settings, click **Security > Global security > Authentication cache settings** (which is under Authentication mechanisms).

- \_\_ e. This time, you are authenticated properly and allowed access.

PLANTS BY WEBSPHERE

HOME : ADMIN HO

[Manage BackOrders](#) - View backorder inventory, order from supplier stock to inventory.

[Supplier Configuration](#) - Configure the Supplier.

Powered by

=====

[Supplier Configuration](#) - Click Supplier Configuration to explore further.

PLANTS BY WEBSPHERE

HOME : ADMIN HO

[Admin Home](#)

## Supplier Configuration

Enter the Supplier's Configuration Information

Full Name	Greenhouse By WebSphere
Street Address	4205 Miami Blvd.
City	Durham
State	NC
Zip	27709
Phone	919-555-1212

- \_\_ g. Click **Home** to return to the main part of the PlantsByWebSphere application.

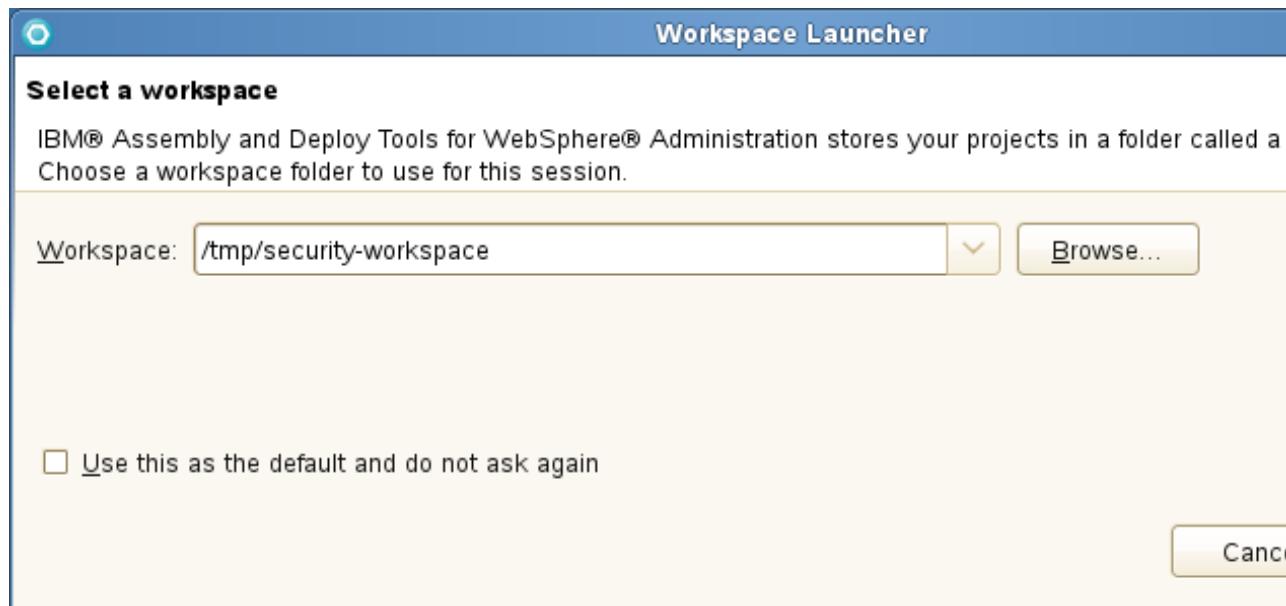
- \_\_\_ 7. Disable application security.
  - \_\_\_ a. Return to the global security screen in the administrative console and clear the box next to **Enable application security**.
  - \_\_\_ b. Click **Apply** and **Save** the changes.
  - \_\_\_ c. Synchronize the nodes.
  - \_\_\_ d. Restart the application servers.

### **Section 4: Exploring the details (optional)**

To complete this part of the exercise, you must start the Assembly and Deploy Tools to the workspace you created earlier in this class. IBM Assembly and Deploy Tools is used to explore the EAR file to discover how security is configured.

Instructions on using IBM Assembly and Deploy Tools itself are short and to the point; you are already familiar with the Assembly and Deploy Tools from previous exercises.

- \_\_\_ 1. Start the Assembly and Deploy Tools and point it to a workspace where you explore the security attributes.
  - \_\_\_ a. To start IBM Assembly and Deploy Tools, enter the following command in a terminal window:  
`/opt/IBM/SDP/eclipse`
  - \_\_\_ b. When prompted for a workspace, enter:  
`/tmp/security-workspace`



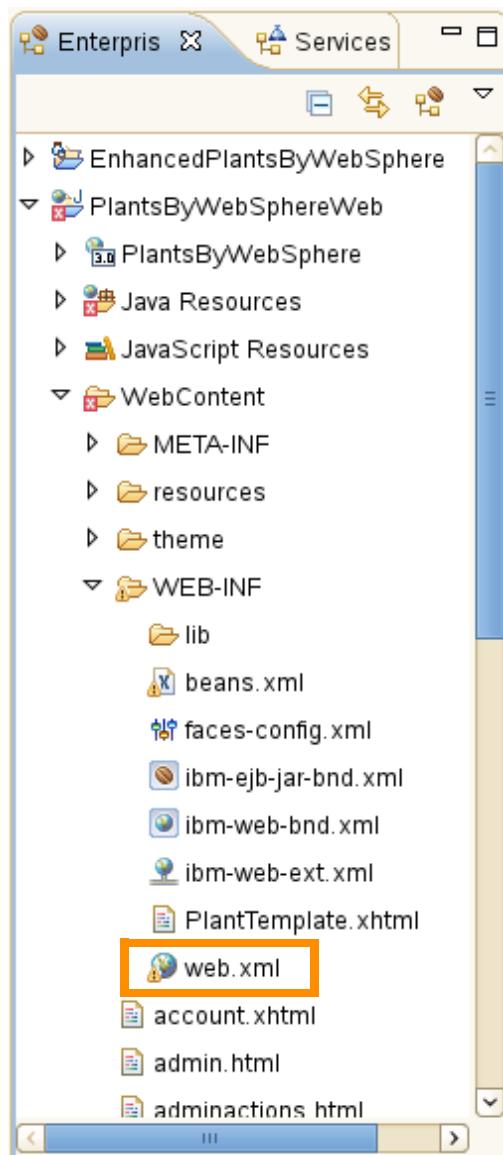
- \_\_\_ c. Click **OK**.
- \_\_\_ 2. Import the PlantsByWebSphere EAR.
  - \_\_\_ a. Click **File > Import > Java EE > EAR file**.

- \_\_ b. Click **Next**.
- \_\_ c. Browse to: /usr/software/ears/EnhancedPlantsByWebSphere.ear
- \_\_ d.
  - Click **OK**.
- \_\_ e. Click **Finish**.
- \_\_ f. Switch to the **Java EE perspective**, if it is not already set.



- \_\_ 3. Explore the details.
  - \_\_ a. If there are warnings and errors, do not worry as you are exploring the contents of the EAR.
  - \_\_ b. In the Enterprise Explorer, expand **PlantsByWebSphereWeb > WebContent > WEB-INF**.
  - \_\_ c. Double-click **web.xml**, which represents the deployment descriptor. The Web Application Deployment Descriptor Editor for this module opens on the editor pane in the upper right corner of the window.

- \_\_\_ d. Double-click the **web.xml** tab to maximize the Web Application Deployment Descriptor Editor. The editor allows you to view the Web Application Structure more easily.



Click **Login Configuration** in the Web Application structure.

The screenshot shows the 'Web Application 3.0 Deployment Descriptor Editor'. In the left pane, under 'Overview', there is a tree structure with nodes like 'Web Application (PlantsByWeb)', 'Context Parameter (javax.fac...', 'Error Page (/error.jsp)', 'Error Page (/ViewExpired.xht...', and 'Login Configuration (Default)'. The 'Login Configuration (Default)' node is selected and highlighted with a blue background. In the right pane, under 'Properties for the login configuration', there are two fields: 'Authentication Method:' set to 'BASIC' and 'Realm Name:' set to 'Default'. Both of these properties are enclosed in orange rectangular boxes. Below this, there is a section titled 'Form Login Configuration (optional)' with fields for 'Form Login Page\*' and 'Form Error Page\*'. There are also 'Up' and 'Down' buttons for reordering items in the tree.

You might see errors that are detected at the top of the page, which can be ignored.



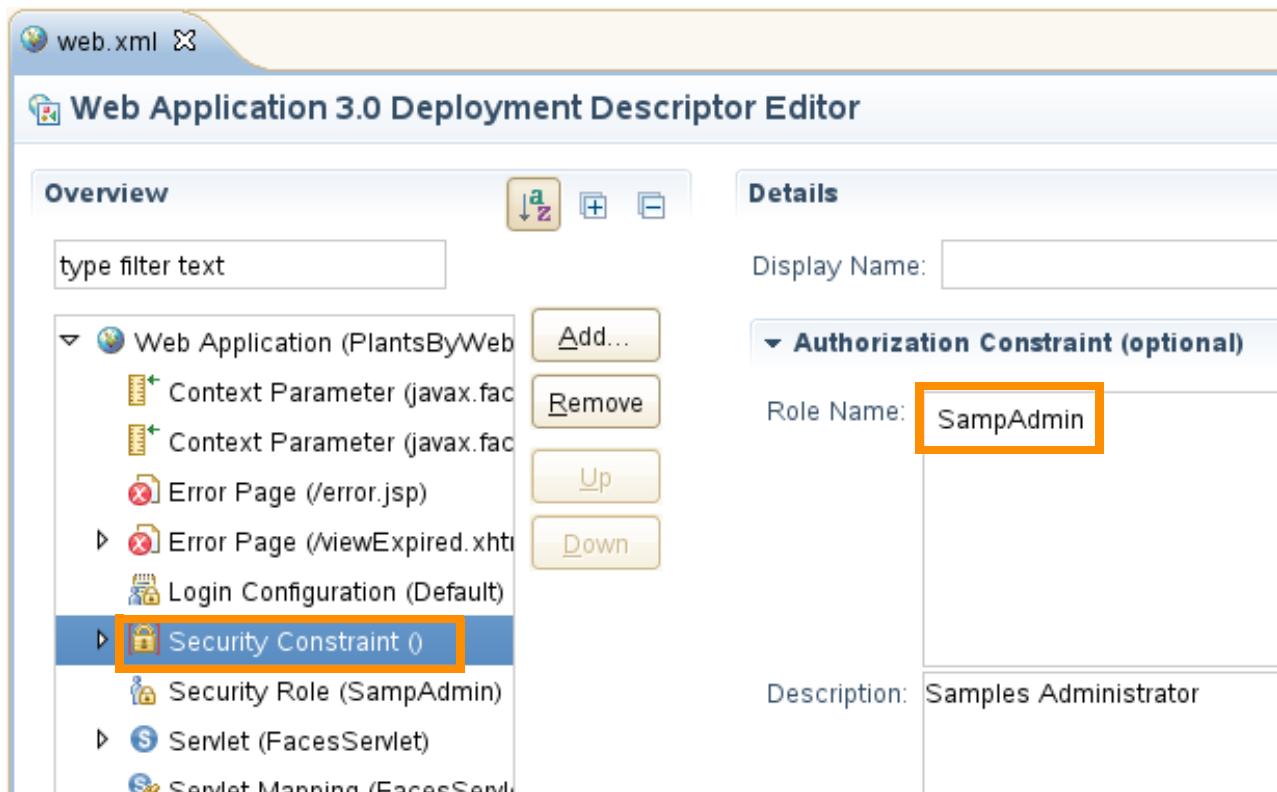
### Note

The Authentication Method is set to `BASIC`, which means that when authentication is needed for a page, the browser is sent a message to challenge the user with a basic authentication dialog box. Perhaps a better alternative would be to use a form-based challenge, which would allow the developer to specify a specific form-based login page that would be used to challenge the user.

When a non-authenticated user accesses a protected resource, WebSphere Application Server presents the login challenge instead of the requested resource. After successful authentication the originally requested resource is served.

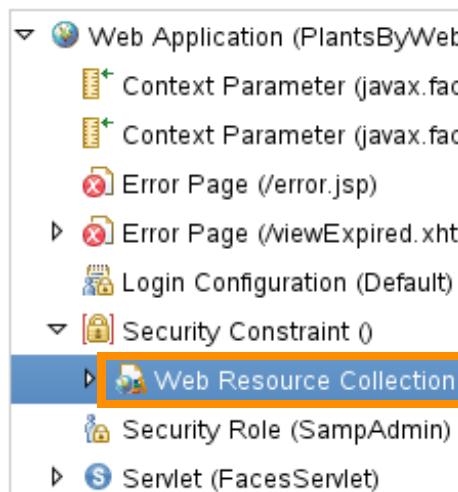
4. Explore the resources.

- \_\_ a. Click **Security Constraint ()** in the Web Application structure of the editor.



Notice the role name SampAdmin in the Authorization Constraint area.

- \_\_ b. Click the **Web Resource Collection** (under Security Constraint) in the Web Application structure of the editor.



In the Details section of the Web Resource Collection, you see a set of HTTP access methods and URL patterns, which can be assigned to an authorized role.

**Details**

Web Resource Name\*: SampAdmin

URL Pattern\*:

/adminactions.html	Add
/adminbanner.html	Remove
/backorderadmin.jsp	Up
/servlet/AdminServlet	Down
/supplierconfig.jsp	

Description: Sample Admin

In this case, you see that one of the URL patterns is /adminactions.html.

### Information

The /adminactions.html URL pattern is associated with both a web module (PlantsByWebSphereWeb) and a security role (SampAdmin). You also see that a basic login type is associated with the web module.

When a user attempts to access a protected URL, the runtime interrupts the process to verify that the user has the required authority to proceed. In this case, the user (PlantsUser) would be required to be authenticated, and the user's user ID must be mapped to the security role (SampAdmin). In this case, the user PlantsUser is part of the PlantsGroup, which in turn is mapped to the SampAdmin role.

If initial authentication is unsuccessful, the user does not gain access to the URL and is instead returned a message. The message indicates that the user was unable to authenticate.

If the authentication is successful, but the user is not mapped to the security role (either directly or through a group), the user again does not gain access to the URL. Instead, the user is returned a message. The message indicates that the user is not authorized to access the URL.

Both `adminactions.html` and `adminbanner.html` are fragments of the `admin.html` page that the browser requests.

- \_\_\_\_\_ 5. Close the IBM Assembly and Deploy tools when you finish exploring.

## End of exercise

## Exercise review and wrap-up

The first part of the exercise explored setting up security for accessing the WebSphere administrative console. Then, application security was enabled and access to the PlantsByWebSphere application was tested.

Finally, an explanation of how Java EE security is configured in the assembly and deploy tool was presented

# Exercise 14.Configuring SSL for WebSphere

## What this exercise is about

This exercise explores some of the features and configurations within the WebSphere SSL environment. It creates a profile and then examines the certificates that are created specifically for the node within the profile. It then explores some of the administration tasks that are required for managing the certificates within a cell.

Finally, in an optional part of the exercise, IBM HTTP Server is configured to use a self-signed certificate to secure the communications between a browser and the web server.

## What you should be able to do

At the end of this exercise, you should be able to:

- Define the certificate life span of a profile
- Use the administrative console to find and view certificates within the cell
- Configure and run the certificate expiration service
- Propagate the generated plug-in keystore out to the plug-in
- Create a keystore for a web server
- Generate a self-signed key
- Configure IBM HTTP Server to load and use HTTPS

## Introduction

WebSphere Application Server V8.5 configures and manages many of the SSL configurations that are required to secure communication within a cell. But it is important to understand how this infrastructure works so that it can be maintained correctly. This exercise creates a profile and examines the certificates and keystores that are created for that new profile or node. The exercise then looks at the interfaces that deal with expiring certificates. It also examines the log files and security reports that are helpful in tracking when certificates are about to expire.

Another important step in managing a WebSphere environment is propagating keystores out to the web server plug-in. This lab goes

through the steps to view the plug-in generated keystore, followed by propagating them out to the web server.

As an optional part of the lab, the last section configures IBM HTTP Server for inbound SSL. iKeyman is used to generate a new keystore and self-signed certificate. IBM HTTP Server is then configured to support HTTPS communications by using the newly created certificate.

## Requirements

This lab requires a computer that is properly set up with WebSphere Application Server V8.5 installed, IBM HTTP Server installed, and the necessary startup and program files.

# Exercise instructions

## Preface



### Important

The labs use two variables to define various installation paths. On Linux, the variable definitions are as follows:

```
<was_root>: /opt/IBM/WebSphere/AppServer
<profile_root>: /opt/IBM/WebSphere/AppServer/profiles
```

## **Section 1: Resetting the WebSphere environment**



### Note

If your WebSphere environment must be reset for any reason, see **Appendix A** for instructions to correctly reset the environment.

## **Section 2: Create a backup**

Since this exercise changes the existing environment, which, if done incorrectly, can cause problems for the rest of the exercises, creating a backup is a good idea.

- 1. Create a backup for the deployment manager.
  - a. In a command window, navigate to the /opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin directory.
  - b. Stop the deployment manager process if it is running by issuing the following command:

```
./stopManager.sh -username wasadmin -password web1sphere
```

- \_\_\_ c. After the deployment manager is stopped, issue the following command to back up the entire profile:

```
./manageprofiles.sh -backupProfile -profileName Dmgr -backupFile
/usr/software/backups/pre-SSL.zip
```

The screenshot shows a terminal window with the following text:

```
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin # ./stopManager.sh -use
rname wasadmin -password websphere
ADMU0116I: Tool information is being logged in file
 /opt/IBM/WebSphere/AppServer/profiles/Dmgr/logs/dmgr/stopServer.log
ADMU0128I: Starting tool with the Dmgr profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server dmgr stop completed.

was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin #
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin # ./manageprofiles.sh -
backupProfile -profileName Dmgr -backupFile /usr/software/backups/pre-SSL.zip
INSTCONFSUCCESS: Success: The profile backup operation was successful.
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin #
```

### **Section 3: Create a profile**

To better understand the various pieces of SSL within the WebSphere Application Server environment, a new custom profile is created.

- \_\_\_ 1. Restart the deployment manager.
- \_\_\_ a. From a terminal window, in the bin directory for the deployment manager, enter the following command:  
  
./startManager.sh
- \_\_\_ 2. Create a custom profile named SSL and federate it to the deployment manager.
  - \_\_\_ a. Start the Profile management tool by issuing the following command in a terminal window:  
  
/opt/IBM/WebSphere/AppServer/bin/ProfileManagement/pmt.sh
  - \_\_\_ b. The WebSphere Customization Toolbox window starts. Click **Create** on the right to create a profile.
  - \_\_\_ c. Select the **Custom profile** option and click **Next**.
  - \_\_\_ d. On the Profile Creation Options page, select **Advanced profile creation** and click **Next**.

\_\_ e. For the profile name and location, enter the following information:

- Profile name: SSL
- Profile directory: /opt/IBM/WebSphere/AppServer/profiles/SSL

Click **Next**.

Profile Management Tool 8.5

**Profile Name and Location**

Specify a profile name and directory path to contain the files for the run-time environment, such as configuration files, and log files. Click **Browse** to select a different directory.

Profile name:  
SSL

Profile directory:  
/opt/IBM/WebSphere/AppServer/profiles/SSL

Make this profile the default.

Each installation of WebSphere Application Server always has one default profile. Commands that refer to a specific profile use the default profile. Select this option to make this profile the new default.

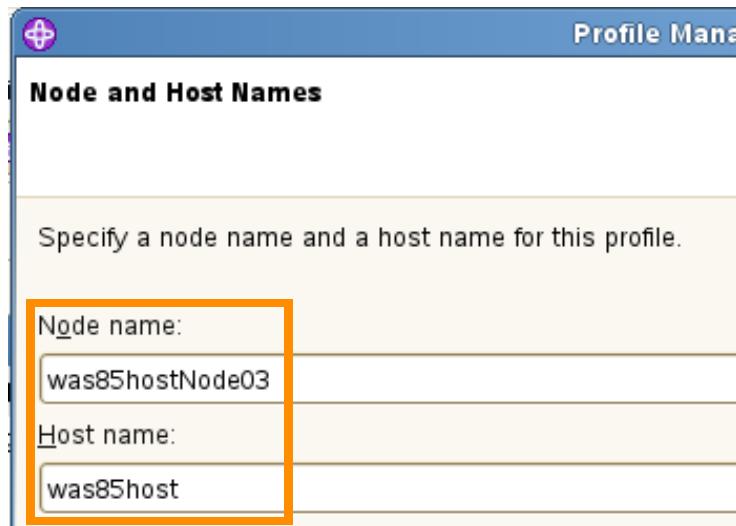
**Important:** Deleting the directory a profile is in does not completely delete the profile. Use the **manage** command to completely delete a profile.

< Back      **Next >**      Cancel

\_\_ f. On the Node and Host Names page, enter the following values:

- Node name: was85hostNode03
- Host name: was85host

Click **Next**.



- g. On the Federation page, enter `was85host` for the deployment manager host name. The default SOAP port (8879) is shown. Enter `wasadmin` for the security User name and `web1sphere` for the Password. Be sure that the **Federate this node later** option is not selected. The node is automatically federated to the cell during creation.

Specify the host name or IP address and the SOAP port number occur only if the deployment manager is running.

Deployment manager host name or IP address:

Deployment manager SOAP port number (Default 8879):

Deployment manager authentication  
Provide a user name and password that can be authenticated, i deployment manager.

User name:

Password:

Federate this node later.

- h. Click **Next**.
- i. On the next page, accept the defaults for creating the default personal certificate and a new root signing certificate. Click **Next**.

- \_\_ j. On the next screen, which specifies the node certificate information, accept the defaults and click **Next**. Make sure that you read the information block as it explains the importance of these entries and how they relate to SSL.

**Profile Management Tool 8.5**

### Security Certificate (Part 2)

Modify the certificate information to create new certificates during profile creation. If you are importing existing certificates from keystores, use the information to verify whether the selected certificates contain the appropriate information. If the selected certificates do not, click **Back** to import different certificates.

**Restore Defaults**

Default personal certificate (a personal certificate for this profile, public and private key):

Issued to distinguished name:  
cn=was85host,ou=was85hostNode01Cell,ou=was85hostNode03,o=IBM,c=US

Issued by distinguished name:  
cn=was85host,ou=Root Certificate,ou=was85hostNode01Cell,ou=was85hostNode03,o=IBM,c=US

Expiration period in years:  
1

Root signing certificate (personal certificate for signing other certificates, public and private key):

Expiration period in years:  
15

Default keystore password:  
\*\*\*\*\*

Confirm the default keystore password:  
\*\*\*\*\*



### Information

Unlike in older versions of WebSphere Application Server, each node (or profile) no longer gets a single self-signed certificate. Instead, as of WebSphere Application Server V7, two certificates are created. The first one is the node personal certificate that is used by default for secured communication with the node and any application servers on that node. This personal certificate has a default life span of one year and it is not a self-signed certificate.

Instead, the second certificate that is specified on this page signed it. The second certificate is the root certificate. This relationship is called a chained certificate.

Unlike the node personal certificate, the root certificate has a default life span of 15 years. This longer life span helps when the personal certificate is renewed as it gets close to its expiration date. Since the same root certificate signed all the personal certificates, any processes that must communicate securely already have access to a valid copy of the node root signer certificate. This condition is true regardless of whether the personal certificates are updated.

This model helps solve some of the certificate propagation problems since updating personal certificates no longer requires any certificate propagation to occur.

Signer certificate propagation within a cell is accomplished through standard node synchronization. All of the node signer certificates are included in the cell default truststore file, which is synchronized throughout the cell. Propagation to the web server plug-in, however, is tricky.

For the plug-in to be able to communicate securely with the application servers, they need access to the appropriate signer certificates. The root signer certificates are made available to the plug-in in their generated key rings. Since the root certificates are now being used as the signers, updating expiring personal certificates is no longer a problem.

The keystore password default is: WebAS

- \_\_\_\_\_ k. Accept the default ports on the next page. Click **Next**.
  - \_\_\_\_\_ l. On the summary page, click **Create**.
  - \_\_\_\_\_ m. The profile creation is now complete; clear the check box for **Launch the First steps console** and click **Finish**.
  - \_\_\_\_\_ n. **Close** the Profile Management Tool.
- \_\_\_\_\_ 3. Verify the new node in the administrative console.
- \_\_\_\_\_ a. Open an instance of the administrative console and use `wasadmin` with the password `web1sphere` to log in to the deployment manager.
  - \_\_\_\_\_ b. Click **System administration > Nodes** and verify that the new node, `was85hostNode03`, now exists.
  - \_\_\_\_\_ c. Stop the node agent for node `was85hostNode03`. Since it is not necessary for the node agent to be running for this exercise, select the node and click **Stop** (which helps free up some of the system resources).

## **Section 4: Examine the node certificates**

This new node has a couple of certificates that are associated with it. This section of the exercise uses the administrative console to examine them.

- 1. Examine the node certificates.
  - a. In the administrative console, click **Security > SSL certificate and key management**.
  - b. On the right side, under Related Items, click **Key stores and certificates**.

**SSL certificate and key management**

**SSL configurations**

The Secure Sockets Layer (SSL) protocol provides secure communications between remote server processes or endpoints. SSL security can be used for establishing communications inbound to and outbound from an endpoint. To establish secure communications, a certificate and an SSL configuration must be specified for the endpoint.

In previous versions of this product, it was necessary to manually configure each endpoint for Secure Sockets Layer (SSL). In this version, you can define a single configuration for the entire application-serving environment. This capability enables you to centrally manage secure communications. In addition, trust zones can be established in multiple node environments by overriding the default, cell-level SSL configuration.

If you have migrated a secured environment to this version using the migration utilities, the old Secure Sockets Layer (SSL) configurations are restored for the various endpoints. However, it is necessary for you to re-configure SSL to take advantage of the centralized management capability.

**Configuration settings**

[Manage endpoint security configurations](#)  
[Manage certificate expiration](#)  
[Manage FIPS](#)

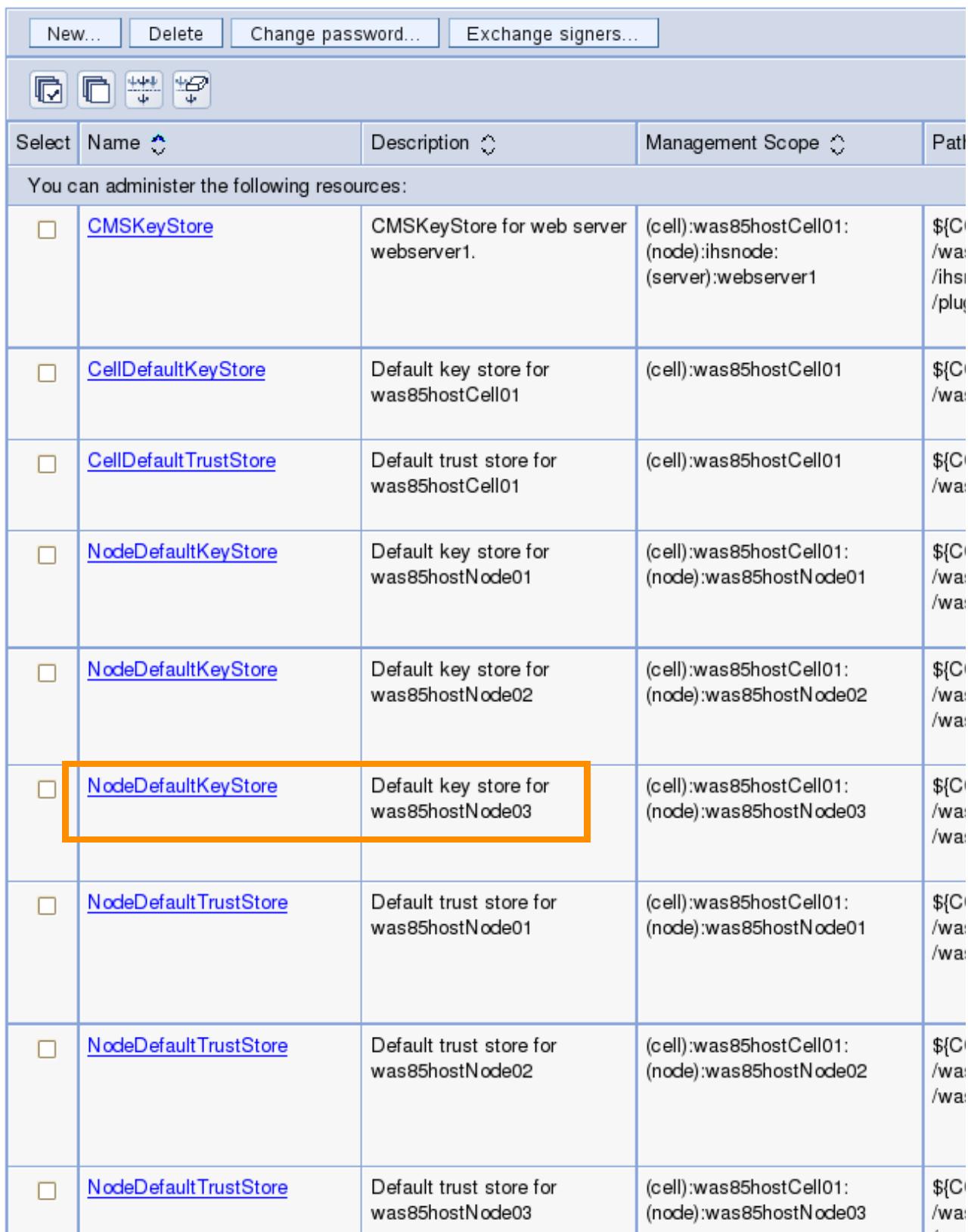
Dynamically update the run time when SSL configuration changes occur

[Apply](#) [Reset](#)

**Related Items**

- [SSL configurations](#)
- [Dynamic outbound endpoint SSL configurations](#)
- **[Key stores and certificates](#)**
- [Key sets](#)
- [Key set groups](#)
- [Key managers](#)
- [Trust managers](#)
- [Certificate Authority \(CA\) client configurations](#)

- \_\_\_ c. This page shows a list of the keystores and trust files for the cell. Click **NodeDefaultKeyStore** for the node that was created (was85hostNode03) earlier in this exercise.



The screenshot shows a table with columns: Select, Name, Description, Management Scope, and Path. The 'Name' column contains links to CMSKeyStore, CellDefaultKeyStore, CellDefaultTrustStore, NodeDefaultKeyStore, NodeDefaultKeyStore, NodeDefaultKeyStore, NodeDefaultTrustStore, NodeDefaultTrustStore, and NodeDefaultTrustStore. The last row, which corresponds to the 'NodeDefaultKeyStore' for 'was85hostNode03', is highlighted with an orange border around its entire row.

Select	Name	Description	Management Scope	Path
You can administer the following resources:				
<input type="checkbox"/>	<a href="#">CMSKeyStore</a>	CMSKeyStore for web server webserver1.	(cell):was85hostCell01: (node):ihsnode: (server):webserver1	\$(C:/wa/ihsi/plug)
<input type="checkbox"/>	<a href="#">CellDefaultKeyStore</a>	Default key store for was85hostCell01	(cell):was85hostCell01	\$(C:/wa/)
<input type="checkbox"/>	<a href="#">CellDefaultTrustStore</a>	Default trust store for was85hostCell01	(cell):was85hostCell01	\$(C:/wa/)
<input type="checkbox"/>	<a href="#">NodeDefaultKeyStore</a>	Default key store for was85hostNode01	(cell):was85hostCell01: (node):was85hostNode01	\$(C:/wa/wa)
<input type="checkbox"/>	<a href="#">NodeDefaultKeyStore</a>	Default key store for was85hostNode02	(cell):was85hostCell01: (node):was85hostNode02	\$(C:/wa/wa)
<input type="checkbox"/>	<a href="#">NodeDefaultKeyStore</a>	Default key store for was85hostNode03	(cell):was85hostCell01: (node):was85hostNode03	\$(C:/wa/wa)
<input type="checkbox"/>	<a href="#">NodeDefaultTrustStore</a>	Default trust store for was85hostNode01	(cell):was85hostCell01: (node):was85hostNode01	\$(C:/wa/wa)
<input type="checkbox"/>	<a href="#">NodeDefaultTrustStore</a>	Default trust store for was85hostNode02	(cell):was85hostCell01: (node):was85hostNode02	\$(C:/wa/wa)
<input type="checkbox"/>	<a href="#">NodeDefaultTrustStore</a>	Default trust store for was85hostNode03	(cell):was85hostCell01: (node):was85hostNode03	\$(C:/wa/wa)

- \_\_\_ d. The next page shows the basic information for the node keystore. On the right, under Additional Properties, click **Personal certificates**.

The screenshot shows the 'Properties' tab for the 'NodeDefaultKeyStore' under 'Key stores and certificates'. The 'Additional Properties' section is visible on the right, with the 'Personal certificates' option highlighted by a red box.

faultKeyStore	com.ibm.ws.crypto.faultkeystore
ion	key store for was85hostNode03
ent scope	was85hostCell01:(node):was85hostNode03
rd	IG_ROOT}/cells/was85hostCell01/nodes/was85hostNode03/key.p12

- \_\_\_ e. This page shows the keystore entries for the node that was created. Notice there are two chained certificates. The first, whose alias is default, is the personal certificate for the new node. Notice that it is set to expire in one year.

The second is the root certificate for the new node, which expires in 15 years. Note the serial number for the root certificate (Serial #: \_\_\_\_\_) and expiration date (date: \_\_\_\_\_).

<input type="checkbox"/> Receive from a certificate authority...	Replace...	Extract...	Import...	Export...	Revoke...	Renew
Alias	Issued To	Issued By	Serial Number	Expiration		
The following resources:						
<a href="#">default</a>	CN=was85host, OU=was85hostNode01Cell, OU=was85hostNode03, O=IBM, C=US	CN=was85host, OU=Root Certificate, OU=was85hostCell01, OU=was85hostCellManager01, O=IBM, C=US	1671126513980145	Valid from Sep 15, 2012 to Sep 15, 2013.		
	CN=was85host, OU=Root Certificate, OU=was85hostCell01, OU=was85hostCellManager01, O=IBM, C=US	CN=was85host, OU=Root Certificate, OU=was85hostCell01, OU=was85hostCellManager01, O=IBM, C=US	1234393675450	Valid from Aug 15, 2012 to Aug 12, 2027.		

- \_\_\_ f. Tracking the serial numbers can be helpful. In other parts of the administrative console, the representation of the certificates can change. To tell which certificate is which, knowing the serial number is helpful. Click the alias default for further information. Also, take note of the fingerprint. Usually knowing just the last couple of bits is sufficient: \_\_\_\_\_ (51:38)

**General Properties**

Alias	default
Version	X509 V3
Key size	2048 bits
Serial number	1671126513980145
Validity period	Valid from Sep 15, 2012 to Sep 15, 2013.
Issued to	CN=was85host, OU=was85hostNode01Cell, OU=was85hostNode03, O=IBM, C=US
Issued by	CN=was85host, OU=Root Certificate, OU=was85hostCell01, OU=was85hostCellManager01, O=IBM, C=US
Fingerprint (SHA digest)	D4:EA:6E:B4:59:BB:26:FC:D1:8F:C3:9F:D7:AE:D8:3C:87:D4:51:38
Signature algorithm	SHA1withRSA(1.2.840.113549.1.1.5)

[Back](#)

- \_\_\_ g. Using the breadcrumb trail, return to the **NodeDefaultKeyStore**.
- \_\_\_ h. Click **Signer certificates**. Notice that none are listed (signers are stored in the truststore files while personal certificates are stored in keystores).
- \_\_\_ 2. Examine the cell signer certificates.
- \_\_\_ a. Using the breadcrumb trail, return to the **Keystores and certificates**.

- \_\_\_ b. Click **CellDefaultTrustStore**. The file includes all of the signer certificates within the cell.

<input type="checkbox"/>	<a href="#">CellDefaultTrustStore</a>	Default trust store for was85hostCell01	(cell):was85hostCell01
--------------------------	---------------------------------------	-----------------------------------------	------------------------

- \_\_\_ c. On the right, under Additional Properties, click **Signer certificates**.



- \_\_\_ d. Notice a root signer certificate in the cell default truststore. Specifically, this certificate is the cell root signer certificate (not the personal certificate), and is the signer for all of the node certificates in the cell. Notice that it has a 15-year life span.

The screenshot shows a list of signer certificates. The first entry is 'root', which is highlighted with an orange box. The details for 'root' are shown: CN=was85host, OU=Root Certificate, OU=was85hostCell01, OU=was85hostCellManager01, O=IBM, C=US. The Fingerprint (SHA Digest) is 02:59:BA:96:96:BB:55:19:D5:7F:EB:CF:5D:11:8C:78:21. The status bar at the bottom indicates 'Total 1'.

- \_\_\_ e. Click **root** for the details.

**General Properties**

Alias	root
Version	3
Key size	2048
Serial number	1234393675450
Validity period	Valid from Aug 15, 2012 to Aug 12, 2027.
Issued to	CN=was85host, OU=Root Certificate, OU=was85hostCell01, OU=was85hostCellManager01, O=IBM, C=US
Issued by	CN=was85host, OU=Root Certificate, OU=was85hostCell01, OU=was85hostCellManager01, O=IBM, C=US
Fingerprint (SHA digest)	02:59:BA:96:96:BB:55:19:D5:7F:EB:CF:5D:11:8C:78:2F:B0:8F:E6
Signature algorithm	SHA1withRSA(1.2.840.113549.1.1.5)

[Back](#)

- \_\_\_ f. Notice that the serial number matches the serial number from the previous step where you noted the information for the root certificate. This match verifies the fact that the signer certificate for the root certificate is indeed added to the cell default truststore. And since the cell default truststore is synchronized to all nodes within a cell, all nodes and all application servers have access to the cell root signer certificate.

The plug-in needs the cell root signer certificate so that it can communicate securely with the application servers. That subject is covered later in this exercise.

Take note of the serial number and fingerprint for the root certificate:  
 \_\_\_\_\_ (5450). Usually knowing just the last couple of bits is sufficient: \_\_\_\_\_ (8F:E6)

## **Section 5: Examine certificate expiration and updating**

Since the personal certificates have a life span of only one year, administrators must be aware that these certificates expire. Fortunately, WebSphere has a built-in mechanism to automatically renew these certificates when they are about to expire. And, since the signer certificates remain the same, it is not necessary to propagate anything new to the remote nodes or plug-in.

- 1. Examine the certificate expiration settings.
  - a. In the administrative console, click **Security > SSL certificate and key management**.
  - b. Under Configuration settings, click **Manage certificate expiration**.

**Configuration settings**

[Manage endpoint security configurations](#)

[Manage certificate expiration](#)

[Manage FIPS](#)

Dynamically update the run time when  
SSL configuration changes occur

**Apply**   **Reset**

Built into WebSphere is a service that runs through the list of all certificates and replaces those certificates that are about to expire. This screen configures when that service is run. It can be run immediately by clicking **Start now**, or it can be scheduled. The default is to run on every fourth Sunday at 21:30. This service can be turned off by clearing the Enable check box. It is also possible to run the checking

service but to not automatically replace the existing certificates or to not delete the replaced expiring certificates.

SSL certificate and key management

[SSL certificate and key management > Manage certificate expiration](#)

Configures the certificate expiration monitor.

**General Properties**

\* Expiration notification threshold  
60 days

\* Certificate pre-notification threshold  
0 days

Enable checking

**Expiration checking**

Scheduled time of day to check for expired certificates  
21 : 30 A.M. P.M. 24-hour  
 Check by calendar  
Weekday Sunday \* Repeat interval  
4 weeks

Check by number of days  
\* Repeat interval  
7 days

Next start date  
Sunday, October 7, 2012  
9:30 PM

Expiration check notification  
MessageLog

Automatically replace expiring self-signed and chained certificates

Delete expiring certificates and signers after replacement

By default, the expiration notifications are written to the log file. More notifications, including email, can be configured by clicking **Notifications** under Related Items.



### Information

The name of the **Expiration notification threshold** field can be misleading. When the expiration notification thread is run (either manually or through scheduling), it checks whether any certificates are within X days (defined by the expiration notification threshold) of the current date. If there are any, they are subject to replacement.

Another setting is not visible through the administrative console that defines how many days out the notification (but not replacement) can occur.

- \_\_\_ c. Run the expiration notification service now by clicking **Start now**. Using a text editor, open the `SystemOut.log` file for the deployment manager. Near the end of the file, an entry that starts with “Expiration Monitor” looks like the following screen capture:

```

SystemOut.log (/opt/IBM/WebSphere/AppServer/profiles/Dmgr/logs/dmgr) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
SystemOut.log X
com.ibm.ws.webcontainer.servlet.ServletWrapper init SRVE0242I: [isclite] [/ibm/console] [/secure/layouts/singleRadioButtonLayout.jsp]: Initialization successful.
[9/16/12 21:04:21:746 EDT] 000000df WSNotifier I CWPKI0037I: Expiration
monitor reports the following information:
**** Subject: Expiration Monitor ****;

Hostname: was85host
Profile UUID: Dmgr-DEPLOYMENT_MANAGER-84078619-7e03-4048-97e9-e4ed7e215844
Process type: DeploymentManager

Checking for expired certificate and certificates in the 60 days threshold period.

CWPKI0735I: All certificates were searched and no expiration issues were found.
.
Ln 1, Col 1 INS

```

- \_\_\_ d. Since the cell is newly created, no certificates need replacing. If any changes must be saved, feel free to either save or discard.
- \_\_\_ 2. Use the security report to view the list of all the certificates and their expiration dates.
  - \_\_\_ a. In the administrative console, click **Security > Global security**.

- \_\_ b. Click Security Configuration Report.

**Global security**

Use this panel to configure administration and the default application security policy. for all administrative functions and is used as a default security policy for user applic customize the security policies for user applications.

**Administrative security**

Enable administrative security

- [Administrative user roles](#)
- [Administrative group roles](#)
- [Administrative authentication](#)

- \_\_ c. A new browser window shows the HTML report. Scroll to the bottom of the report and find the Certificate Management section.

Certificate Management	default ( CellDefaultKeyStore )	Valid from Aug 15, 2012 to Aug 13, 2022.
Certificate Management	root ( CellDefaultTrustStore )	Valid from Aug 15, 2012 to Aug 12, 2027.
Certificate Management	default ( CellRSATokenKeyStore )	Valid from Aug 15, 2012 to Aug 13, 2022.
Certificate Management	root ( CellRSATokenTrustStore )	Valid from Aug 15, 2012 to Aug 12, 2027.
Certificate Management	root ( DmgrDefaultRootStore )	Valid from Aug 15, 2012 to Aug 12, 2027.
Certificate Management	dummyserversigner ( DmgrDefaultDeletedStore )	Valid from Jul 30, 2003 to Oct 13, 2021.
Certificate Management	dummyclientsigner ( DmgrDefaultDeletedStore )	Valid from Jul 30, 2003 to Oct 13, 2021.
Certificate Management	root ( DmgrDefaultSignersStore )	Valid from Aug 15, 2012 to Aug 12, 2027.
Certificate Management	root ( DmgrRSATokenRootStore )	Valid from Aug 15, 2012 to Aug 12, 2027.
Certificate Management	default ( NodeDefaultKeyStore )	Valid from Aug 15, 2012 to Aug 13, 2022.
Certificate Management	default ( NodeDefaultTrustStore )	Valid from Aug 15, 2012 to Aug 12, 2027.
Certificate Management	root ( NodeDefaultTrustStore )	Valid from Aug 14, 2012 to Aug 11, 2027.
Certificate Management	default ( NodeDefaultKeyStore )	Valid from Aug 15, 2012 to Aug 13, 2022.
Certificate Management	default ( NodeDefaultTrustStore )	Valid from Aug 15, 2012 to Aug 12, 2027.
Certificate Management	root ( NodeDefaultTrustStore )	Valid from Aug 15, 2012 to Aug 12, 2027.
Certificate Management	default ( CMSKeyStore )	Valid from Aug 15, 2012 to Aug 13, 2022.
Certificate Management	CN=was85host, OU=Root Certificate, OU=was85 OU=was85hostCellManager01, O=IBM, C=US ( CMSKeyStore )	Valid from Aug 15, 2012 to Aug 12, 2027.

- \_\_ d. Notice the list of the certificates and the expiration dates. This report can be a helpful tool for administrators in dealing with their certificate management.
- \_\_ e. Close the security report window.

## Section 6: Plug-in key ring propagation

Not only are the processes within cells (deployment managers, node agents, and application servers) required to have certificates and know about other signer certificates, so are the web server plug-ins. To secure the communication between the web server plug-ins and the application servers, the plug-ins and application servers must be able to negotiate an SSL session. They must have personal certificates (by default the application servers use the node personal certificate) and have access to the other signer certificates.

WebSphere is able to make sure that all of the required certificates are available to the web server plug-in by creating the plug-in keystores from within WebSphere. By doing so, WebSphere can make sure that not only does the plug-in have a valid personal certificate, but it also has the necessary cell root signer certificate. At the same time, WebSphere can ensure that the plug-in signer certificate is also available in the cell truststore.

The real problem with this approach is that after WebSphere generates the plug-in keystore, it still must be propagated to the host that is running the web server. The propagation process of plug-in keystores is similar to the propagation of the `plugin-cfg.xml` file. It is usually done manually, but in some cases can be configured to be done automatically (usually not desirable).

- \_\_\_ 1. View the contents of the plug-in keystore.
  - \_\_\_ a. In the administrative console, click **Servers > Server Types > Web servers**.
  - \_\_\_ b. Click your web server link, **webserver1**.

- \_\_ c. Under Additional Properties, click **Plug-in properties**.

The screenshot shows the 'Web servers' interface with the path 'Web servers > webserver1 > Plug-in properties'. The 'Runtime' tab is selected. The 'Plug-in properties' section contains several configuration options:

- Ignore DNS failures during Web server startup
- \* Refresh configuration interval:  
60 seconds

The 'Repository copy of Web server plug-in files:' section includes:

- \* Plug-in configuration file name: plugin-cfg.xml (highlighted)
- Automatically generate the plug-in configuration file
- Automatically propagate plug-in configuration file

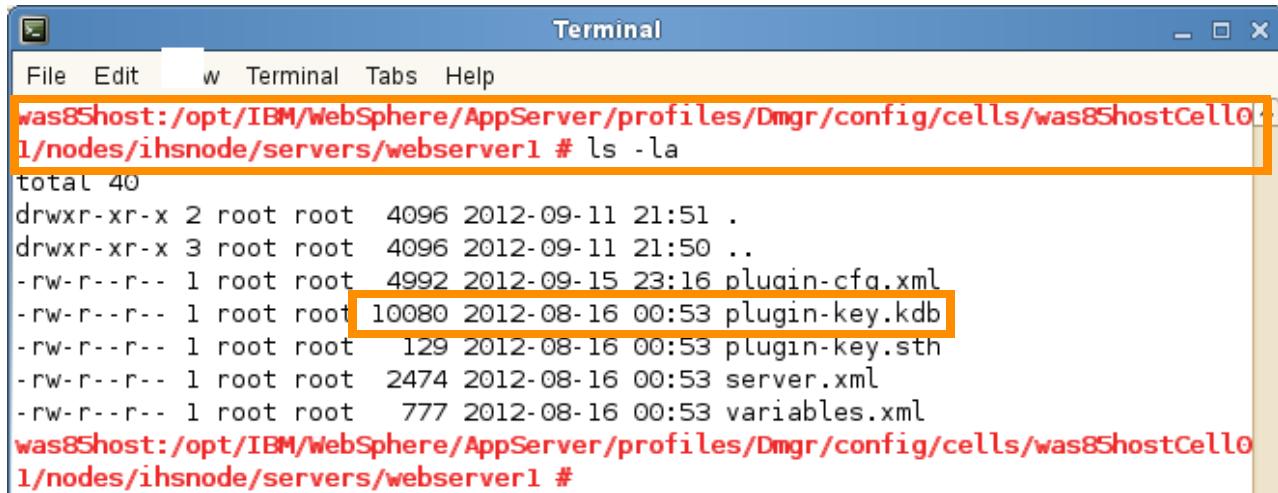
The 'Plug-in key store file name' section includes:

- \* Plug-in key store file name: plugin-key.kdb (highlighted)
- 
- 

- \_\_ d. Notice the plug-in keystore file name. Initially this file can be found within the configuration directory structure of the deployment manager. It can be found under the `ihsnode` directory. More specifically, the directory would be:

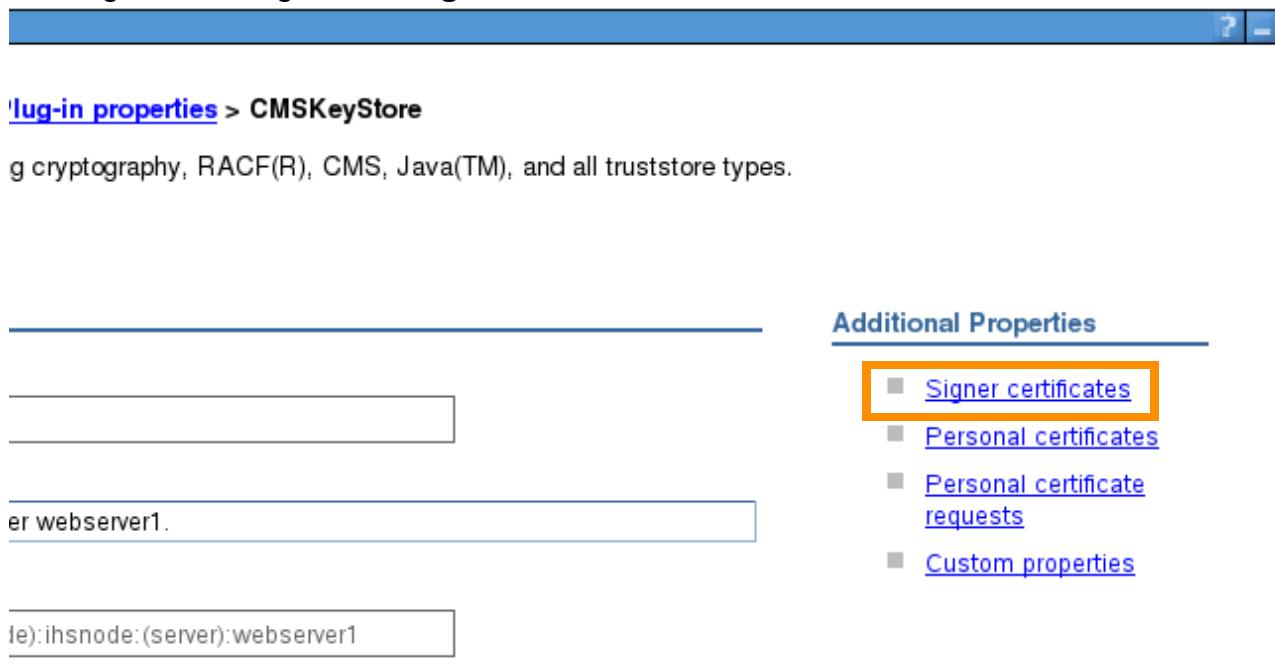
```
/opt/IBM/WebSphere/AppServer/profiles/Dmgr/config/cells/
<cell-name>/nodes/<node-name>/servers/webserver1/nodes/ihsnode/
servers/<web-server>
```

The directory is also the same directory where the web server-specific version of the plugin-cfg.xml file exists.



```
Terminal
File Edit w Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/config/cells/was85hostCell01/nodes/ihsnode/servers/webserver1 # ls -la
total 40
drwxr-xr-x 2 root root 4096 2012-09-11 21:51 .
drwxr-xr-x 3 root root 4096 2012-09-11 21:50 ..
-rw-r--r-- 1 root root 4992 2012-09-15 23:16 plugin-cfg.xml
-rw-r--r-- 1 root root 10080 2012-08-16 00:53 plugin-key.kdb
-rw-r--r-- 1 root root 129 2012-08-16 00:53 plugin-key.sth
-rw-r--r-- 1 root root 2474 2012-08-16 00:53 server.xml
-rw-r--r-- 1 root root 777 2012-08-16 00:53 variables.xml
was85host:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/config/cells/was85hostCell01/nodes/ihsnode/servers/webserver1 #
```

- \_\_\_ e. Take note of the size of the file.
- \_\_\_ f. Next, examine the contents of this file. In the console, click **Manage keys and certificates**.
- \_\_\_ g. On the right, click **Signer certificates**.



Plug-in properties > CMSKeyStore

g cryptography, RACF(R), CMS, Java(TM), and all truststore types.

---

Additional Properties

- [Signer certificates](#)
- [Personal certificates](#)
- [Personal certificate requests](#)
- [Custom properties](#)

- \_\_\_ h. There is one signer certificate. Verify that the cell root signer certificate is among the list. Notice that the fingerprint matches what was seen previously in this exercise.

The screenshot shows a table with columns: Select, Alias, Issued to, and Fingerprint (SHA Digest). The first row, which is highlighted with an orange border, contains the following information:

<input type="checkbox"/>	<a href="#">CN=was85host, OU=Root Certificate, OU=was85hostCell01, OU=was85hostCellManager01, O=IBM, C=US</a>	CN=was85host, OU=Root Certificate, OU=was85hostCell01, OU=was85hostCellManager01, O=IBM, C=US	02:59:BA:96:96:BB:55:19:D5:7F:E
--------------------------	---------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------	---------------------------------

Total 1

Fingerprint (SHA Digest)	Expiration
02:59:BA:96:96:BB:55:19:D5:7F:EB:CF:5D:11:8C:78:2F:B0:8F:E6	Valid from Aug 15, 2012 to Aug 12, 2027.

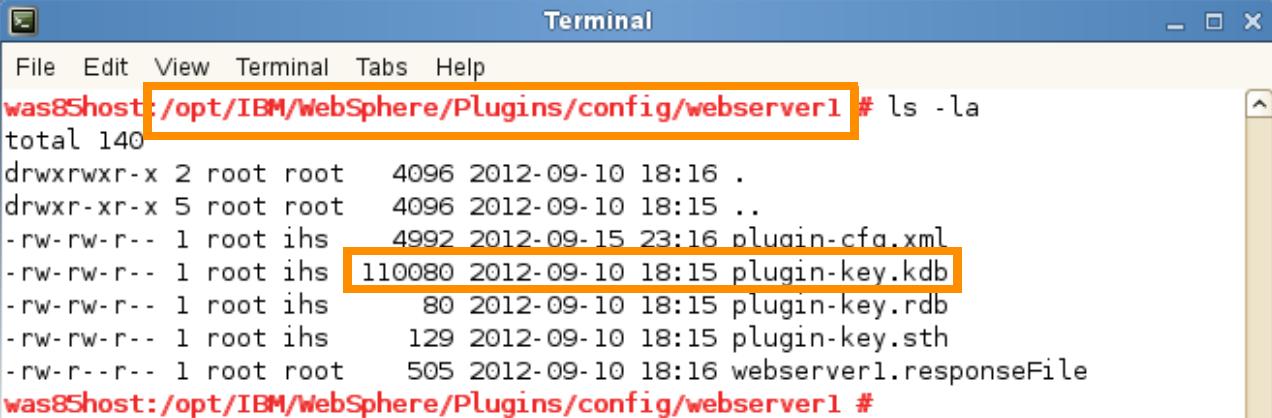


### Information

Seeing the cell root signer certificate (in this case, the certificate that ends with a fingerprint of 01:66) validates the fact that WebSphere generated the keystore for the plug-in and included the cell root certificate signer.

- \_\_\_ 2. Propagate the plug-in keystore file. Although propagation is usually a manual process, there are some cases where it can be configured to be done automatically or through the administrative console.
- \_\_\_ a. Using command window, navigate to the /opt/IBM/WebSphere/Plugins/config/webserver1 directory.

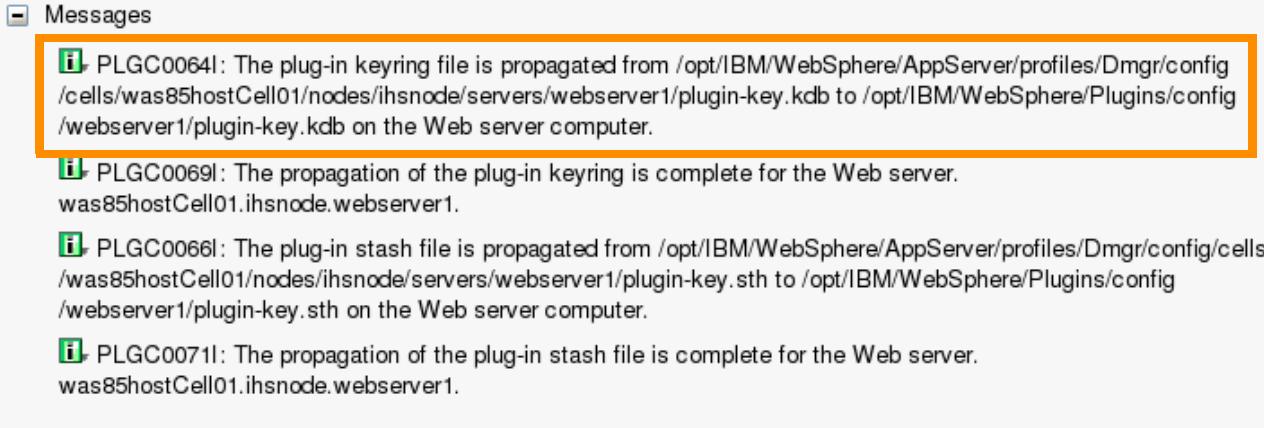
- \_\_\_ b. Use the `ls -la` command to get a directory listing. Take note of the size and date-time stamp for the current `plugin-key.kdb`:



```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/Plugins/config/webserver1 # ls -la
total 140
drwxrwxr-x 2 root root 4096 2012-09-10 18:16 .
drwxr-xr-x 5 root root 4096 2012-09-10 18:15 ..
-rw-rw-r-- 1 root ihs 4992 2012-09-15 23:16 plugin-cfg.xml
-rw-rw-r-- 1 root ihs 110080 2012-09-10 18:15 plugin-key.kdb
-rw-rw-r-- 1 root ihs 80 2012-09-10 18:15 plugin-key.rdb
-rw-rw-r-- 1 root ihs 129 2012-09-10 18:15 plugin-key.sth
-rw-r--r-- 1 root root 505 2012-09-10 18:16 webserver1.responseFile
was85host:/opt/IBM/WebSphere/Plugins/config/webserver1 #
```

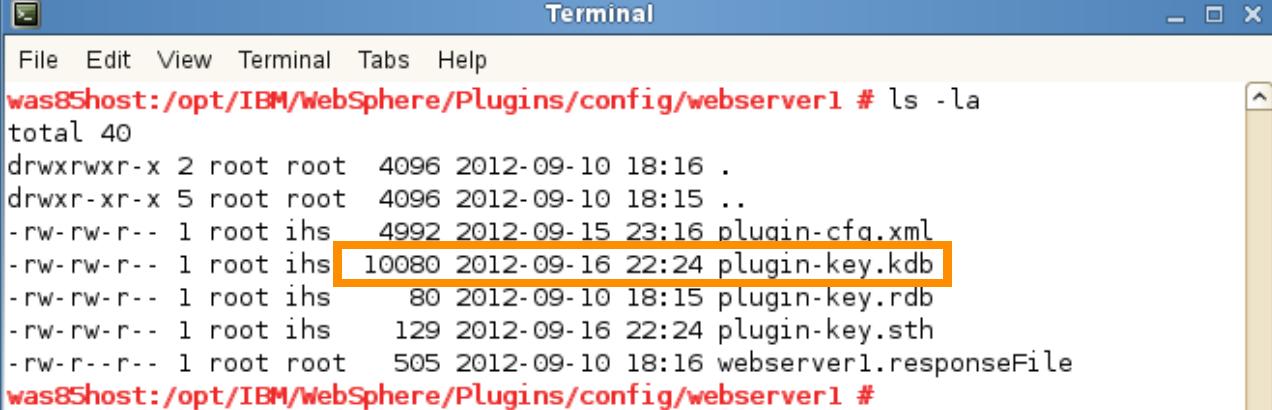
- \_\_\_ c. Return to the administrative console and return to the web server plug-in properties page. Click **Copy to Web server key store directory**.

Notice that the message at the top of the administrative console window shows a successful copying of the files:



Message
PLGC0064I: The plug-in keyring file is propagated from /opt/IBM/WebSphere/AppServer/profiles/Dmgr/config/cells/was85hostCell01/nodes/ihsnode/servers/webserver1/plugin-key.kdb to /opt/IBM/WebSphere/Plugins/config/webserver1/plugin-key.kdb on the Web server computer.
PLGC0069I: The propagation of the plug-in keyring is complete for the Web server. was85hostCell01.ihsnode.webserver1.
PLGC0066I: The plug-in stash file is propagated from /opt/IBM/WebSphere/AppServer/profiles/Dmgr/config/cells/was85hostCell01/nodes/ihsnode/servers/webserver1/plugin-key.sth to /opt/IBM/WebSphere/Plugins/config/webserver1/plugin-key.sth on the Web server computer.
PLGC0071I: The propagation of the plug-in stash file is complete for the Web server. was85hostCell01.ihsnode.webserver1.

- \_\_\_ d. Return to the Explorer window and verify that the size of the file (and the date-time stamp) is changed, which validates that the propagation did occur.



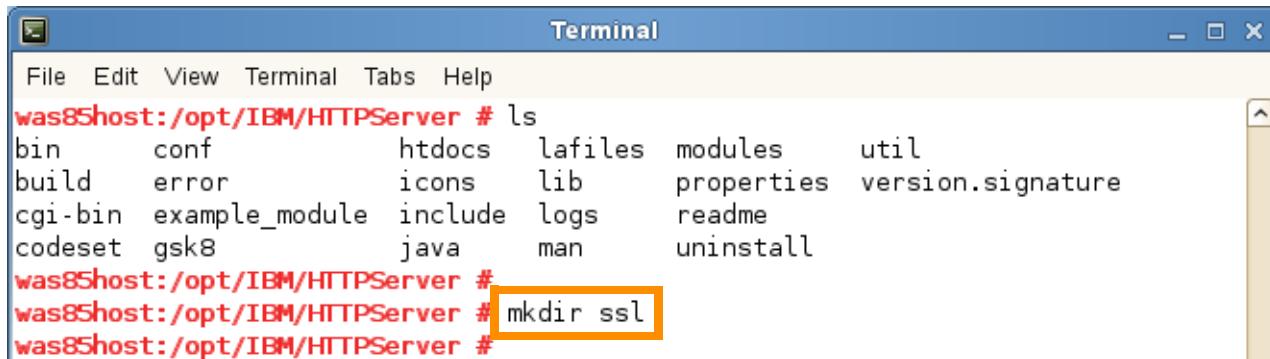
```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/Plugins/config/webserver1 # ls -la
total 40
drwxrwxr-x 2 root root 4096 2012-09-10 18:16 .
drwxr-xr-x 5 root root 4096 2012-09-10 18:15 ..
-rw-rw-r-- 1 root ihs 4992 2012-09-15 23:16 plugin-cfg.xml
-rw-rw-r-- 1 root ihs 10080 2012-09-16 22:24 plugin-key.kdb
-rw-rw-r-- 1 root ihs 80 2012-09-10 18:15 plugin-key.rdb
-rw-rw-r-- 1 root ihs 129 2012-09-16 22:24 plugin-key.sth
-rw-r--r-- 1 root root 505 2012-09-10 18:16 webserver1.responseFile
was85host:/opt/IBM/WebSphere/Plugins/config/webserver1 #
```

- \_\_\_ 3. If you have extra time, use iKeyman to open the .kdb file on the web server host and verify that the new signer certificate is propagated.

## Section 7: Configuring SSL for IBM HTTP Server (optional)

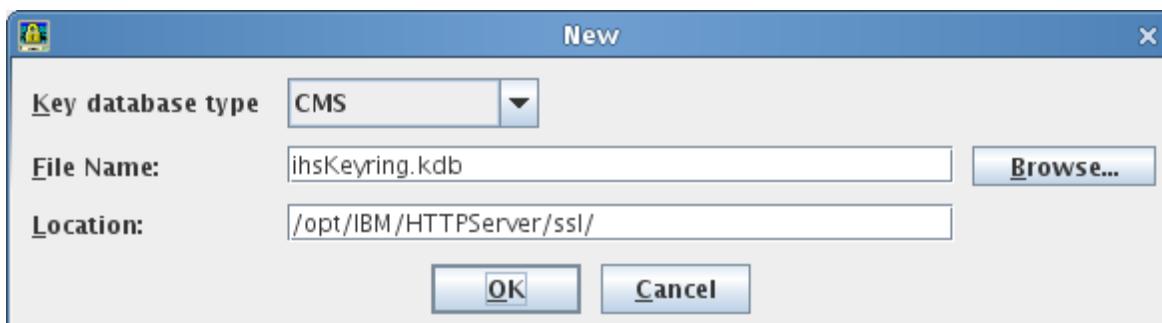
This part of the exercise examines the process of creating a certificate and a key ring for the web server. The steps are used to configure SSL on the connection between the client browser and the web server.

- \_\_\_ 1. Create a directory to hold the key ring.  
 \_\_\_ a. Using the `mkdir` command, create the directory `ssl` in `/opt/IBM/HTTPServer`.



```
File Edit View Terminal Tabs Help
was85host:/opt/IBM/HTTPServer # ls
bin conf htdocs lafiles modules util
build error icons lib properties version.signature
cgi-bin example_module include logs readme
codeset gsk8 java man uninstall
was85host:/opt/IBM/HTTPServer #
was85host:/opt/IBM/HTTPServer # mkdir ssl
was85host:/opt/IBM/HTTPServer #
```

- \_\_\_ 2. Create a key ring with a self-signed certificate for IBM HTTP Server.  
 \_\_\_ a. Run the iKeyman for IBM HTTP Server by issuing the following command:  
`/opt/IBM/HTTPServer/bin/ikeyman`
- \_\_\_ b. Create a new key ring by clicking **Key Database File > New**.
- \_\_\_ c. Supply the following information:
- Key database type: CMS
  - File Name: `ihsKeyring.kdb`
  - Location: `/opt/IBM/HTTPServer/ssl`



- \_\_\_ d. Click **OK**.

- \_\_\_ e. When prompted for a password for the key ring, enter and confirm `websphere` as the password. Modify the expiration time if you want to. Check the **Stash password to a file** check box.



### Warning

The stash file is created containing an encoded form of the password. This encoding prevents casual viewing of the password, but is not highly secure. Therefore, you must protect this file by using operating system file permissions to prevent all access from unauthorized principals.

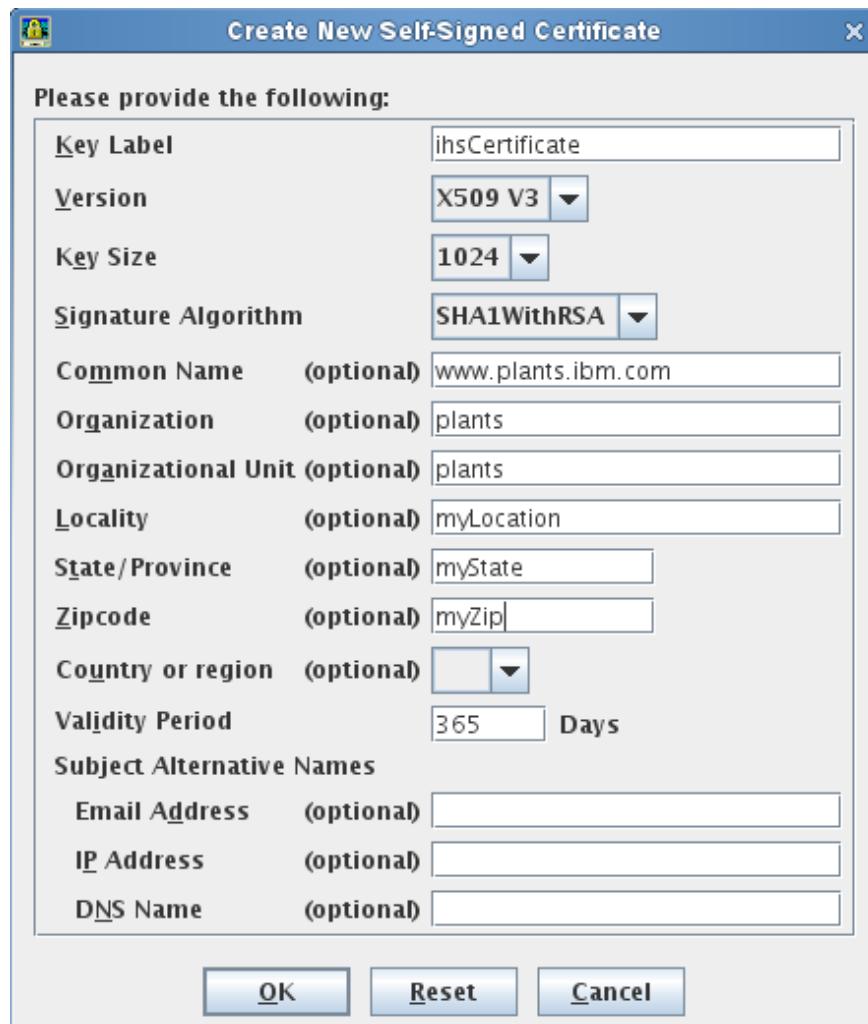
The file name of the stash file is the same as the name of the key file, only it has a `.sth` suffix. The stash file gets stored in the same directory as the key file.

- \_\_\_ f. Click **OK**.
- \_\_\_ 3. Create a new self-signed certificate.
- \_\_\_ a. In iKeyman, click **Create > New Self-Signed Certificate** and enter the following information:

**Table 14: Self-signed certificate details**

Example	Description
Key label	<code>ihsCertificate</code>
Common name	<code>www.plants.ibm.com</code>
Organization	<code>plants</code>
Organization unit	<code>plants</code>
Locality	<code>myLocation</code>
State or province	<code>myState</code>
Zip code	<code>myZipcode</code>

- \_\_ b. Accept the defaults for the Version, Key Size, and Validity Period.



- \_\_ c. Click **OK**.



### Information

This operation stores the certificate in the key file in the **Personal Certificates** section. Optionally, it is possible to extract the public signing certificate so that clients can use it. To extract, click **Extract Certificate**, and then enter a **File Name** and **Location**. Click **OK**.

- \_\_ d. Exit iKeyman by clicking **Key Database File > Exit**.
- \_\_ e. Check the contents of the `/opt/IBM/HTTPServer/ssl/` directory and verify that the following files were created: `ihsKeyring.kdb`, `ihsKeyring.sth`, and `ihsKeyring.rdb`.

- \_\_\_ 4. Configure IBM HTTP Server for HTTPS, which requires modifying the `httpd.conf` file to define the required setting to enable SSL for IBM HTTP Server. It also includes loading the SSL module, defining a listener port, defining a virtual host, and enabling SSL.
- \_\_\_ a. Add `www.plants.ibm.com` to the hosts file by editing the file `/etc/hosts` and adding a line at the bottom to define the host name: `www.plants.ibm.com`. Map it to the IP address for your system. You can use the command `ifconfig` in a terminal window to find your IP address.

```

hosts (/etc) - gedit
File Edit View Search Tools Documents Help
hosts X
#
hosts This file describes a number of hostname-to-address
mappings for the TCP/IP subsystem. It is mostly
used at boot time, when no name servers are running.
On small systems, this file can be used instead of a
"named" name server.
#
Syntax:
#
IP-Address Full-Qualified-Hostname Short-Hostname
#
127.0.0.1 was85host localhost dbhost
192.168.1.17 www.plants.ibm.com

```

- \_\_\_ b. **Save** and exit the file.
- \_\_\_ c. Use the following command in a terminal window to confirm that you can reach `www.plants.ibm.com`:
- ```
ping -c 4 www.plants.ibm.com
```
- ___ d. Back up the `httpd.conf` file. Copy the `httpd.conf` file in `/opt/IBM/HTTPServer/conf` to `httpd-backup.conf`.
- ___ e. Using a text editor, open `httpd.conf` in `/opt/IBM/HTTPDServer/conf`.

- ___ f. Add a virtual host definition for HTTPS, which allows for the definition of HTTPS on a separate virtual host from HTTP. Place these lines near the bottom of the httpd.conf file after the VirtualHost examples and just before the comment:

```
# Enable IBM HTTP Server diagnostic features  
  
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so  
Listen 0.0.0.0:443  
<VirtualHost www.plants.ibm.com:443>  
SSLEnable  
</VirtualHost>  
KeyFile "/opt/IBM/HTTPServer/ssl/ihskyring.kdb"  
SSLDisable
```



Information

There are sample configuration files in /usr/software/ssl/ that can be used to copy and paste. These files include only this section of a completed httpd.conf file.

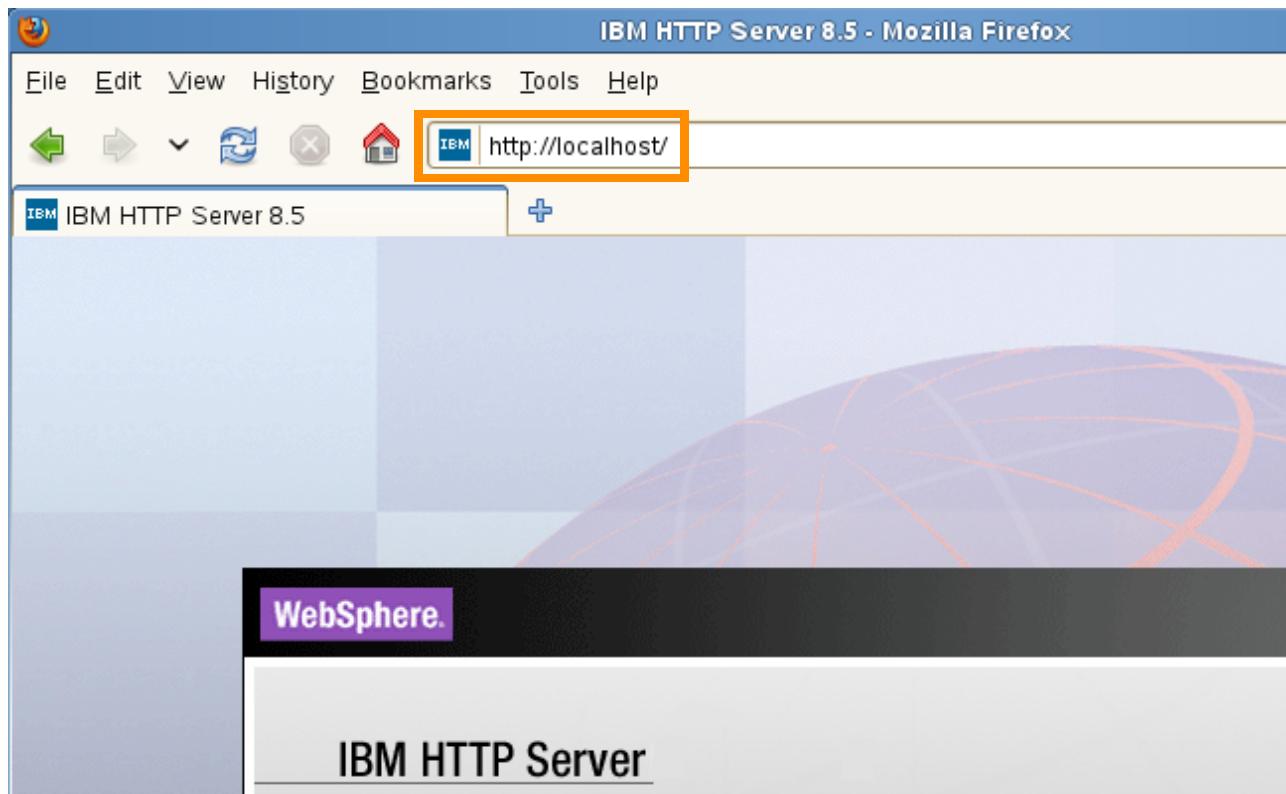
- ___ g. **Save** your changes and exit the editor.

Section 8: Testing the SSL connection

- ___ 1. Restart the IBM HTTP Server process so that the new httpd.conf settings take effect.
 - ___ a. Using a terminal window, use the following command to restart the web server process:
`/opt/IBM/HTTPServer/bin/apachectl restart`
 - ___ b. Verify that the IBM HTTP Server process is running by checking the system process list for httpd. If IBM HTTP Server failed to start, check the `/opt/IBM/HTTPServer/logs/error.log` and `/opt/IBM/WebSphere/Plugins/logs/webserver1/http_plugin.log` files for the possible cause.
- ___ 2. Use HTTPS to connect to IBM HTTP Server.

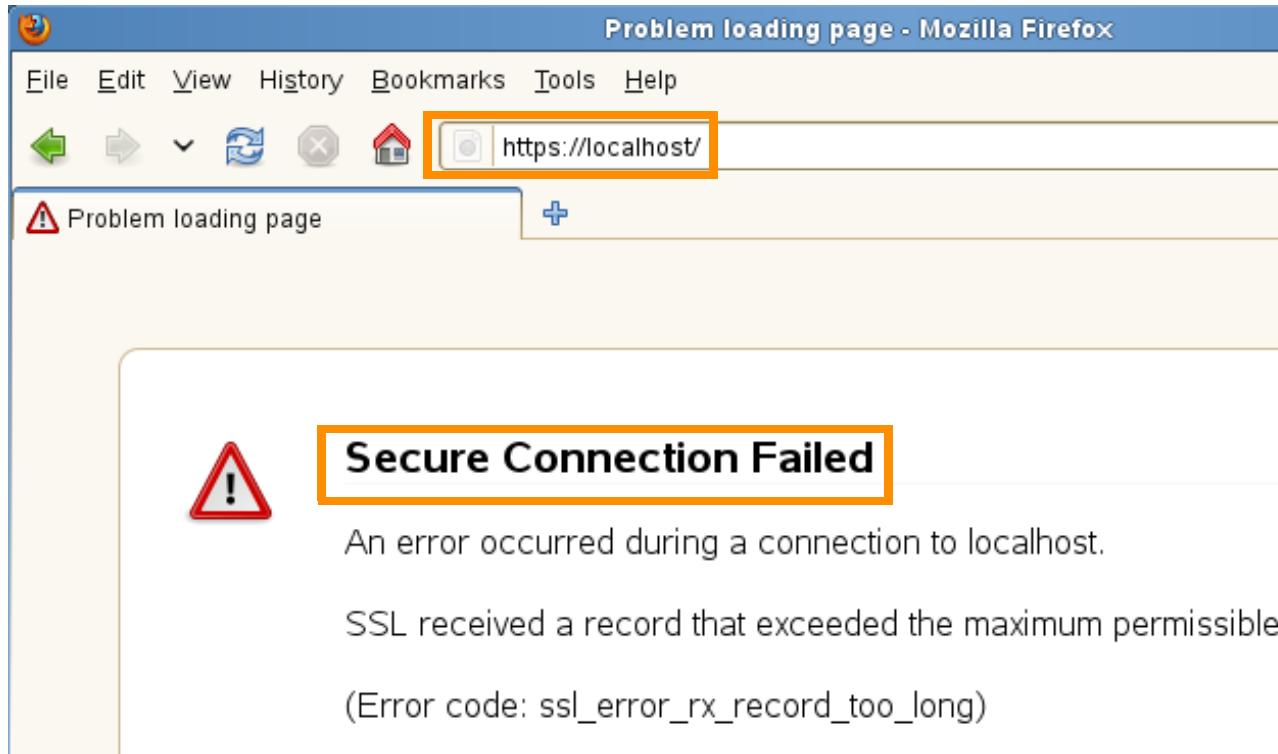
- __ a. First, verify that the web server is actually running. Connect to the following site:

<http://localhost/>



- __ b. Now that the web server is known to be running, enter the following address to verify that HTTPS is working (notice, the only difference is that the HTTP protocol is replaced with HTTPS):

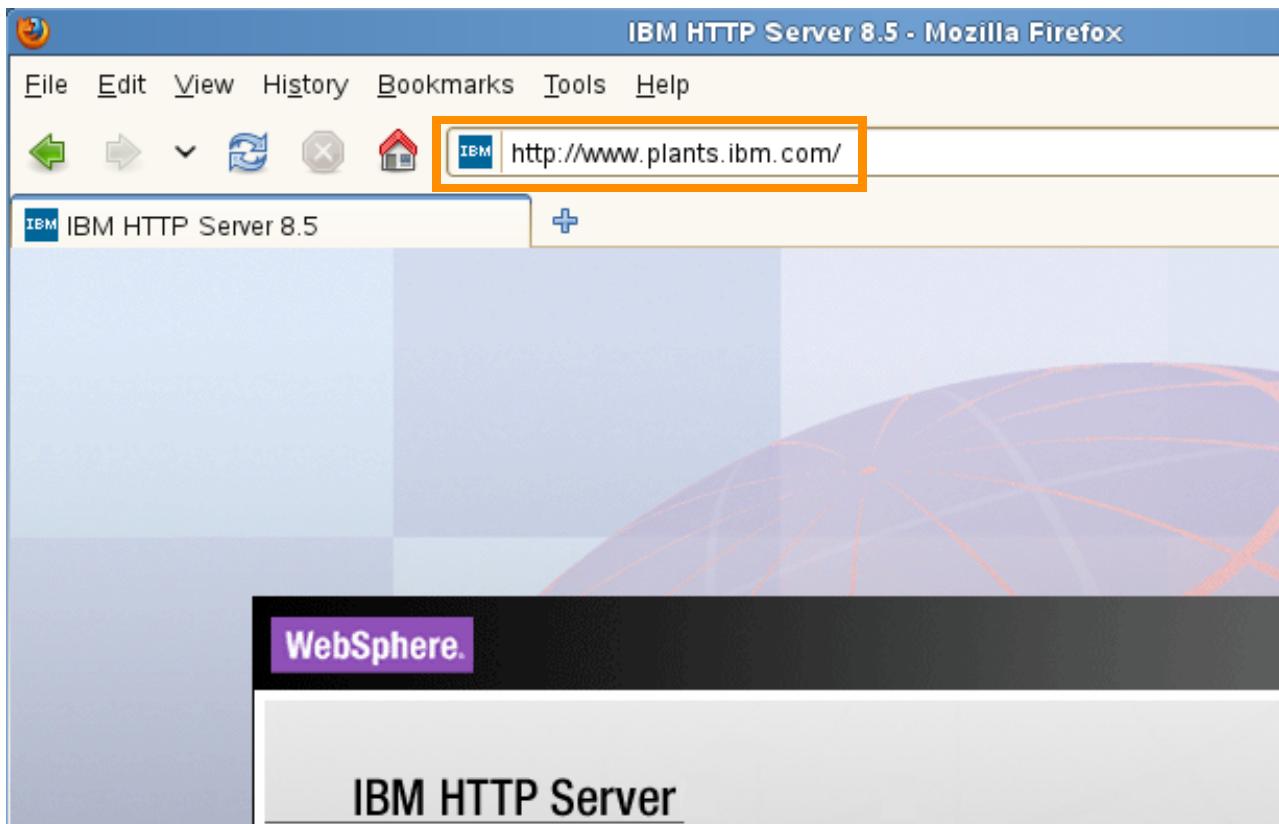
`https://localhost/`



Localhost fails because the virtual host definition for the SSL configuration is defined for the host named `www.plants.ibm.com`.

- ___ c. Using the same browser window, enter the following URL:

<http://www.plants.ibm.com/>



This request works since the host name `www.plants.ibm.com` is mapped to the local system and there is no https necessary.

- ___ d. Now that you verified that the new host name works, change the protocol to HTTPS so that the URL is:

<https://www.plants.ibm.com/>

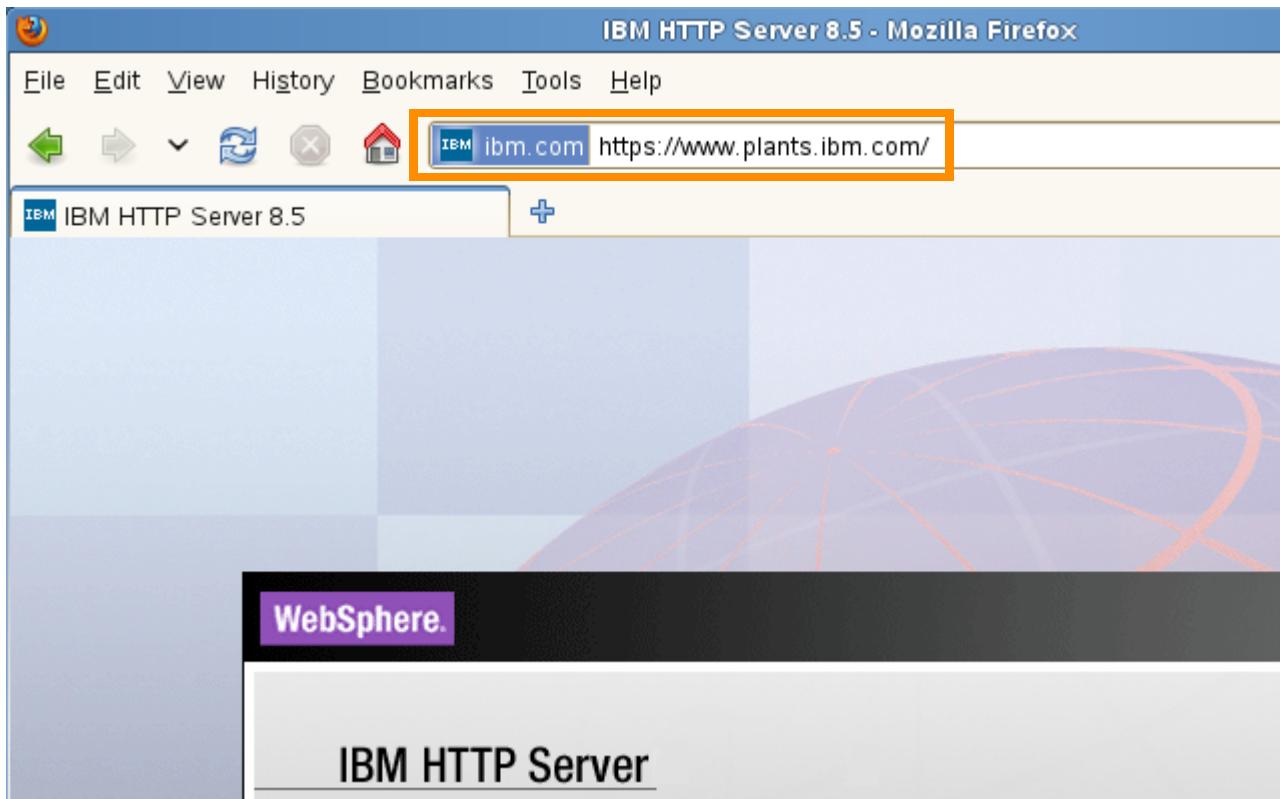
- ___ e. Assuming that the certificate is not added to the browser key ring, you receive a certificate warning. The warning happens because the SSL connection is presenting a self-signed certificate; and therefore, the browser is unable to validate the signer. Expand the **I Understand the Risks** section and click **Add Exception**.

The screenshot shows a Firefox browser window with the following interface elements:

- Untrusted Connection**: A red warning icon with a yellow exclamation mark.
- Add Exception**: A blue plus sign button.
- This Connection is Untrusted**: The main title of the warning message.
- Icon**: A yellow square icon containing a person holding a briefcase.
- Message**: "You have asked Firefox to connect securely to **www.plants.ibm.com**, but we can't verify that the connection is secure."
- Text**: "Normally, when you try to connect securely, sites will present trusted identification that's easy to check. This site is going to the right place. However, this site's identity can't be verified."
- Section**: "What Should I Do?"
- Text**: "If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue."
- Button**: "Get me out of here!"
- Section**: "► Technical Details"
- Section**: "▼ I Understand the Risks" (highlighted with an orange border)
- Text**: "If you understand what's going on, you can tell Firefox to start trusting this site's identity. **But be careful! If you trust the site, this error could mean that someone is tampering with your connection.**"
- Text**: "Don't add an exception unless you know there's a good reason why this site doesn't present trusted identification."
- Button**: "Add Exception..." (highlighted with an orange border)

- ___ f. Click **Confirm Security Exception**.

- ___ g. The browser then takes you to the HTTPS connection for the web server home page.



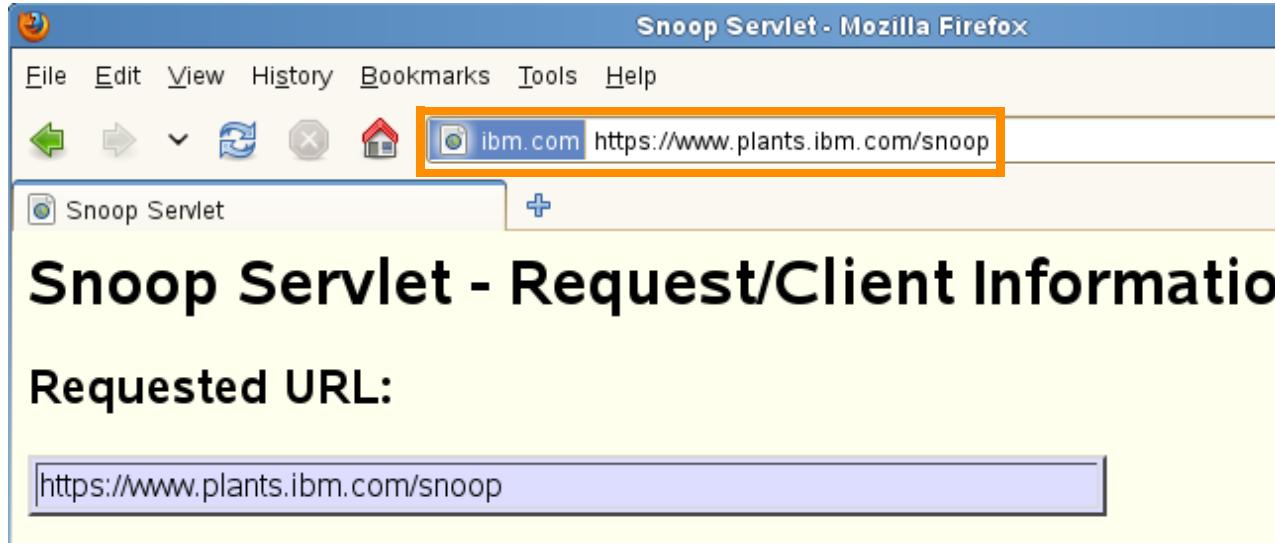
The page works as expected because the host name (`www.plants.ibm.com`) matches the virtual host in the SSL configuration in the `httpd.conf` file.

- ___ h. Click **ibm.com** to the left of the https to view more information about the connection.



- __ i. Click **More Information** and then **View Certificate** to view detailed information about the certificate that is being used for the SSL connection.
- __ 3. Next, use HTTPS to connect to the application server.
 - __ a. Verify that the **PlantsCluster** is running.
 - __ b. Using the existing browser, enter the following URL to access the snoop servlet:

`https://www.plants.ibm.com/snoop`

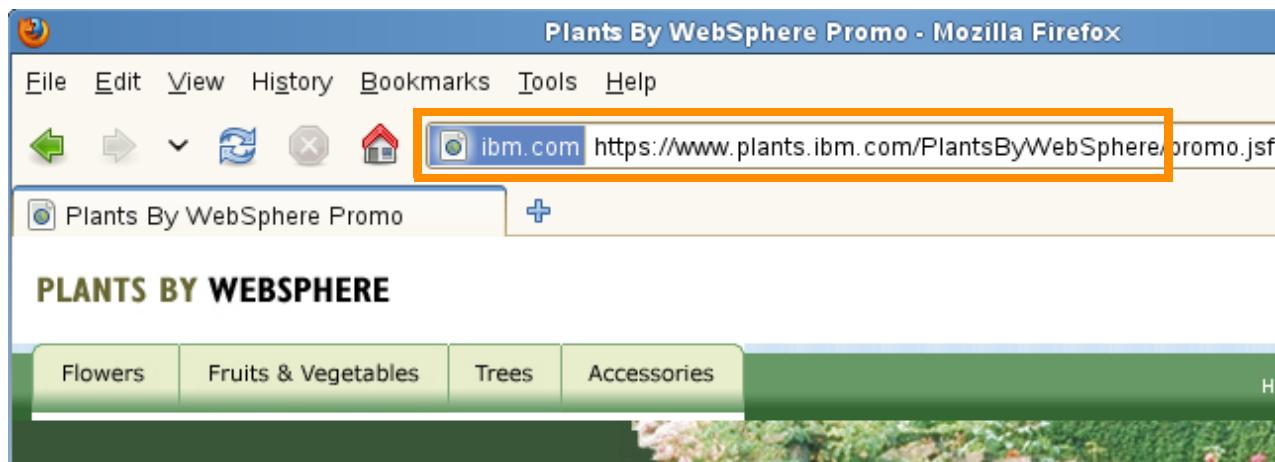


The screenshot shows a Mozilla Firefox browser window titled "Snoop Servlet - Mozilla Firefox". The address bar displays the URL `https://www.plants.ibm.com/snoop`, which is highlighted with an orange rectangle. Below the address bar, the title bar reads "Snoop Servlet". The main content area contains the heading "Snoop Servlet - Request/Client Information" and the sub-section "Requested URL:" followed by the URL `https://www.plants.ibm.com/snoop`.

Notice that it works as expected. If it does not, start by verifying that the plug-in key ring was propagated (which was done earlier in the exercise) and checking the plug-in log file for information.

- __ c. Finally, use the following URL to access the PlantsByWebSphere application:

`https://www.plants.ibm.com/PlantsByWebSphere`



The screenshot shows a Mozilla Firefox browser window titled "Plants By WebSphere Promo - Mozilla Firefox". The address bar displays the URL `https://www.plants.ibm.com/PlantsByWebSphere/promo.jsp`, which is highlighted with an orange rectangle. Below the address bar, the title bar reads "Plants By WebSphere Promo". The main content area features the heading "PLANTS BY WEBSPHERE" and a navigation menu with categories: Flowers, Fruits & Vegetables, Trees, and Accessories.

End of exercise

Exercise review and wrap-up

This exercise introduced basic HTTPS configuration concepts for both IBM HTTP Server and WebSphere Application Server.

Exercise 15. Working with the Liberty profile

What this exercise is about

In this exercise, you use IBM Assembly and Deploy Tools to install the Liberty profile runtime environment. You create an instance of the Liberty profile server. Then, you configure it for data source access and security. Finally, you deploy applications to it.

What you should be able to do

At the end of this exercise, you should be able to:

- Use IBM Assembly and Deploy Tools to install the Liberty Profile Runtime Environment
- Start and stop a Liberty profile application server by using the command line and through IBM Assembly and Deploy Tools
- Deploy a simple application by using IBM Assembly and Deploy Tools
- Deploy an application by using the `dropins` directory
- Deploy an application with a data source
- Configure SSL for a Liberty profile application server
- Configure a user registry for a Liberty profile application server
- Configure application security for a Liberty profile application server
- Use flexible configuration to create shared configurations
- Configure a Liberty profile server to generate a plug-in configuration file

Introduction

This exercise starts by using the IBM Assembly and Deploy Tools to install the Liberty profile runtime environment. A Liberty profile application server is then created and explored. The server is started and stopped, through both the command line and the tool. A simple `HelloWorld` application is deployed through both the tool and the `dropins` directory.

Then, a more complex application, the blogApplication, is deployed. As part of this deployment, the blog data source is configured. This data source allows the blog application to access a Derby database.

The next section of this exercise configures security for the application server. The first security element is configuring the application server to support SSL on its secured port. Next, the exercise looks at the configurations for both an LDAP and the Basic user registry. Finally, application security configuration is explored.

Flexible configuration is then explored by using several shared configuration files. Flexible configuration allows multiple Liberty servers to share configuration. The use of scoped variables is also explored.

Other configurations are also investigated, including plug-in generation.

Requirements

This exercise does not depend on any previous exercise. To complete this exercise, the following files are needed: `HelloWorld.war`, `HelloWorldDropin.war`, `blog.war`, and `derby.jar`. These files are provided in the `/usr/software/Liberty` directory on the course VMware image.

Exercise instructions



Important

The labs use two variables to define various installation paths. On Linux, the variable definitions are as follows:

```
<was_root>: /opt/IBM/WebSphere/AppServer  
<profile_root>: /opt/IBM/WebSphere/AppServer/profiles
```

Section 1: Resetting the WebSphere environment



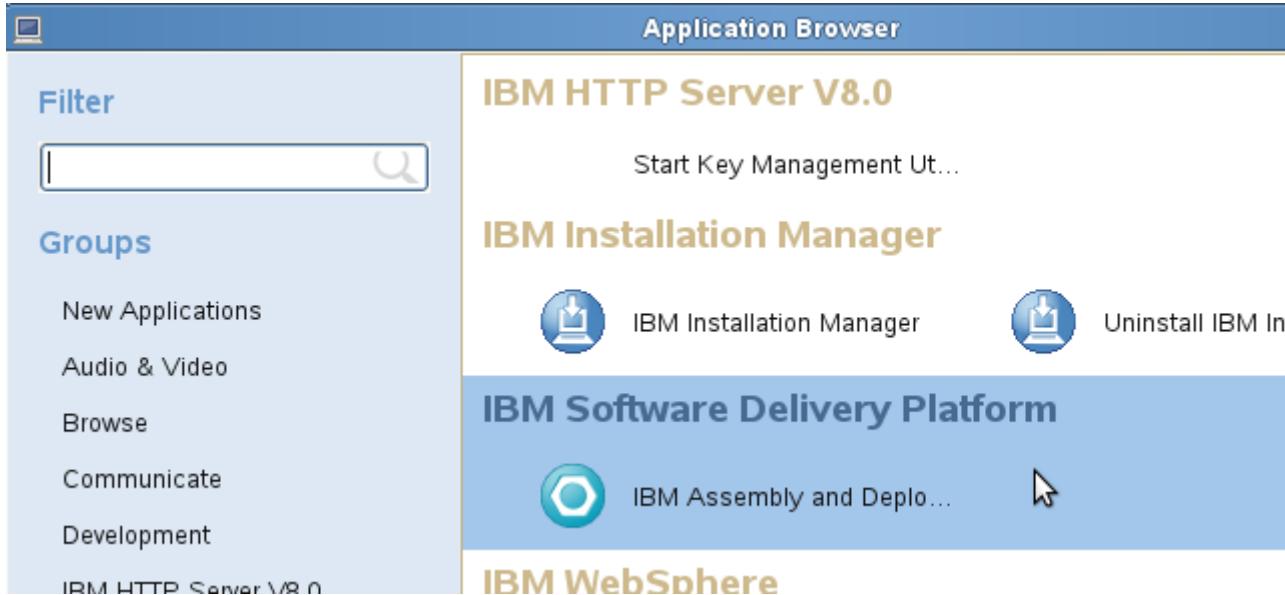
Note

To reset your WebSphere environment, read **Appendix A** for instructions on how to complete this procedure.

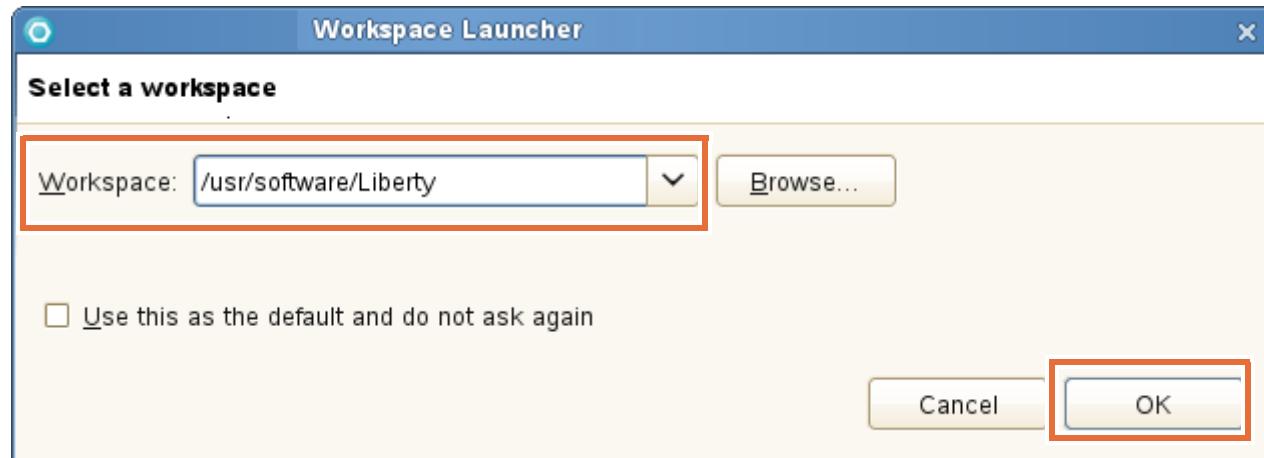
Section 2: Start the IBM Assembly and Deploy Tools for WebSphere Administration and install the Liberty profile

- ___ 1. Stop all WebSphere servers that might be running from previous exercises.
 - ___ a. Change directory to:
`/opt/IBM/WebSphere/AppServer/profiles/profile1/bin`
 - ___ b. Enter the command: `./stopServer.sh server1`
 - ___ c. Enter the command: `./stopNode.sh`
 - ___ d. Change directory to:
`/opt/IBM/WebSphere/AppServer/profiles/profile2/bin`
 - ___ e. Enter the command: `./stopServer.sh server2`
 - ___ f. Enter the command: `./stopNode.sh`
 - ___ g. Change directory to: `/opt/IBM/WebSphere/AppServer/profiles/Dmgr/bin`
 - ___ h. Enter the command: `./stopManager.sh`

- __ 2. Start IBM Assembly and Deploy Tools for WebSphere Administration.
- __ a. Click **Computer > More Applications > IBM Software Delivery Platform > IBM Assembly and Deploy Tools for WebSphere Administration** to start IBM Assembly and Deploy Tools.



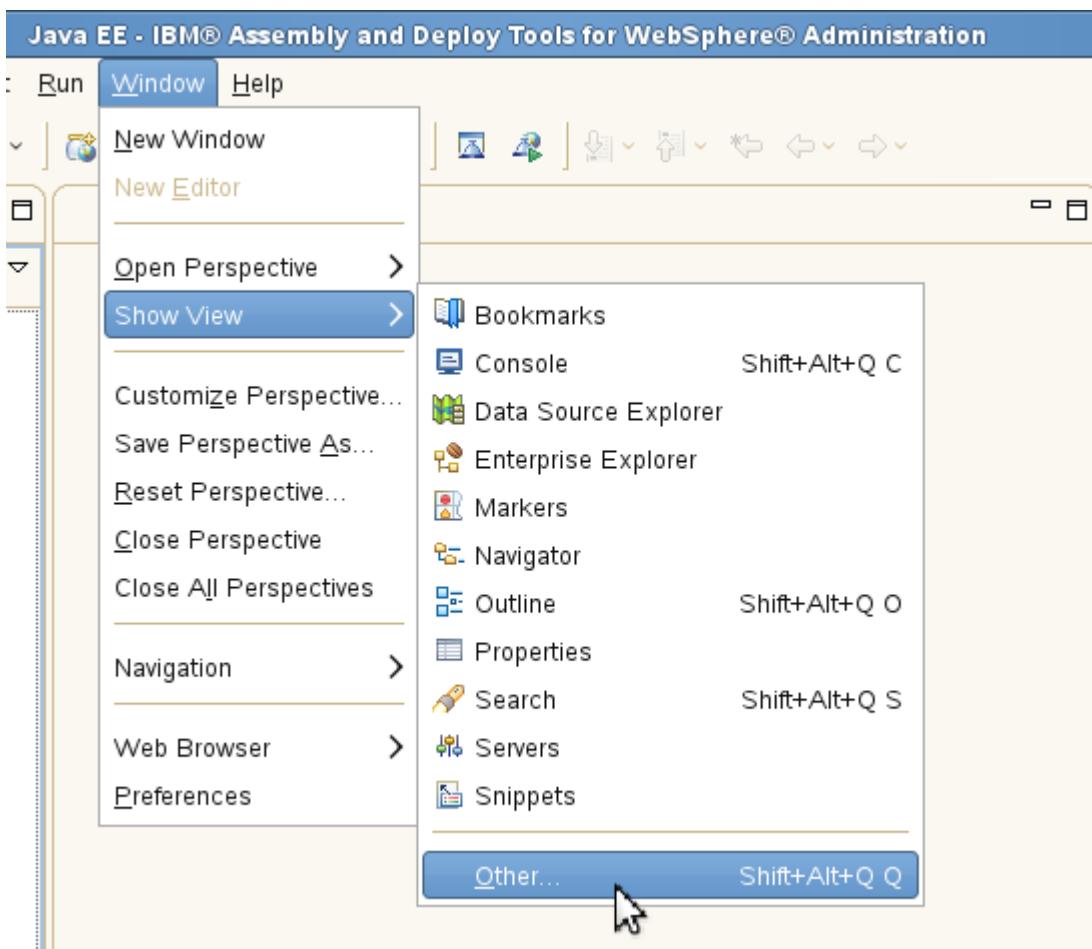
- __ b. Enter `/usr/software/Liberty` for the workspace and click **OK**.



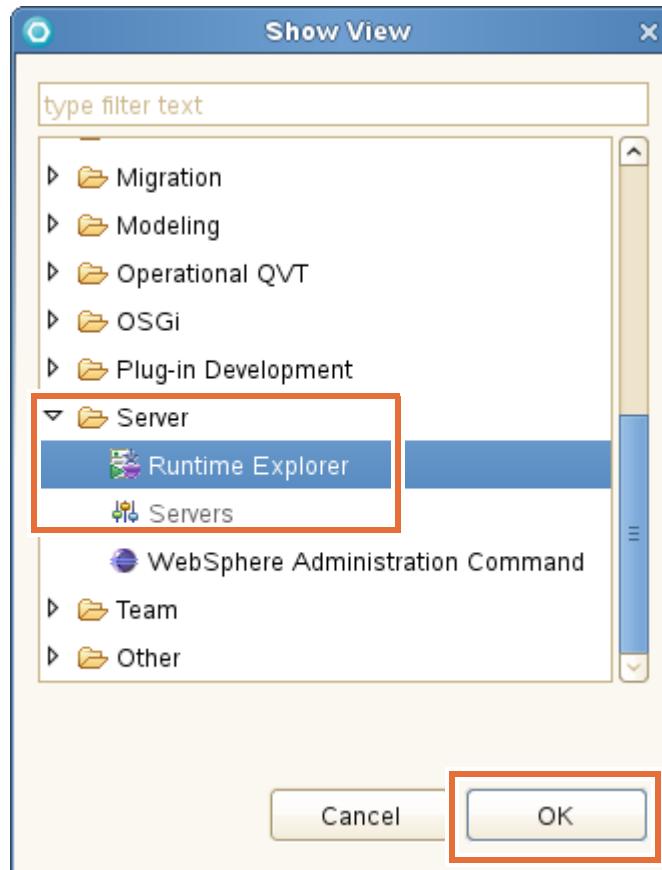
- __ c. Wait several seconds for the workspace to load. When the workspace opens, you are in the **Java EE** perspective to begin assembling the application. This perspective is the default, and it is the title of the workbench.

3. Create the Liberty runtime environment.

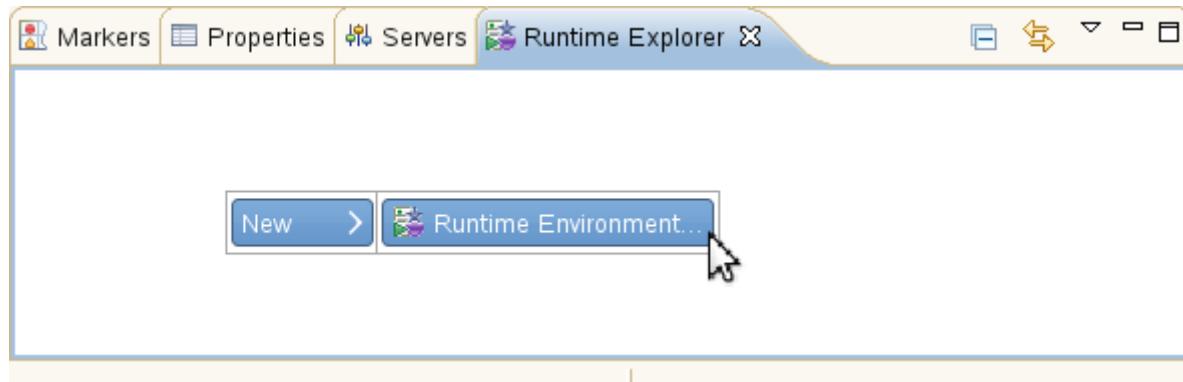
a. Click **Window > Show View > Other**.



- __ b. Expand **Server**, select **Runtime Explorer**, and click **OK**.



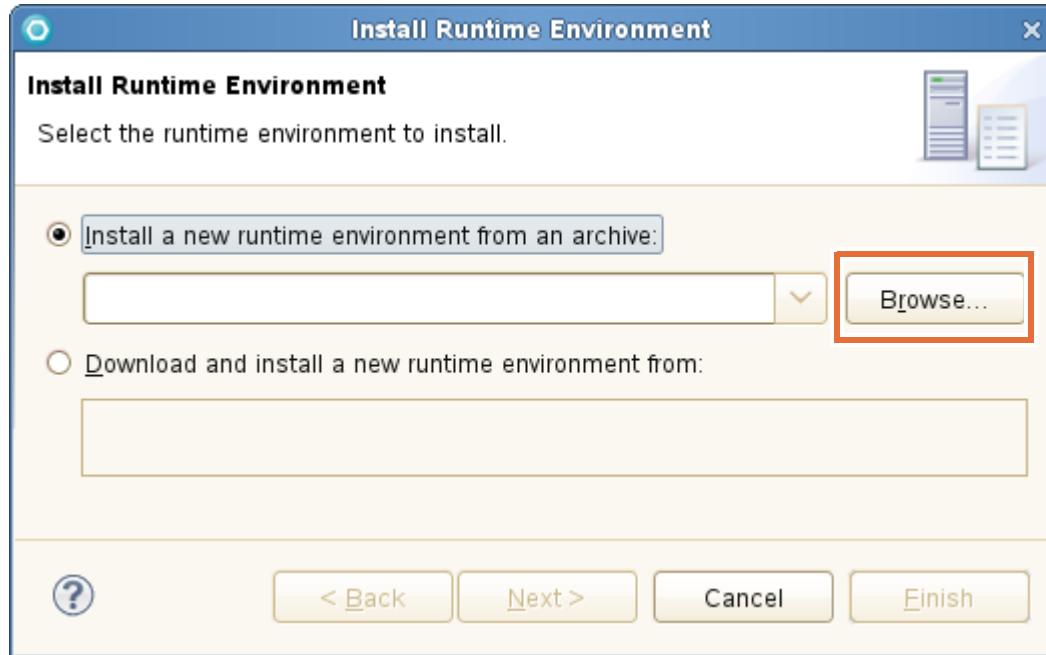
- __ c. Right-click the blank space in the Runtime Explorer tab, and select **New > Runtime Environment**.



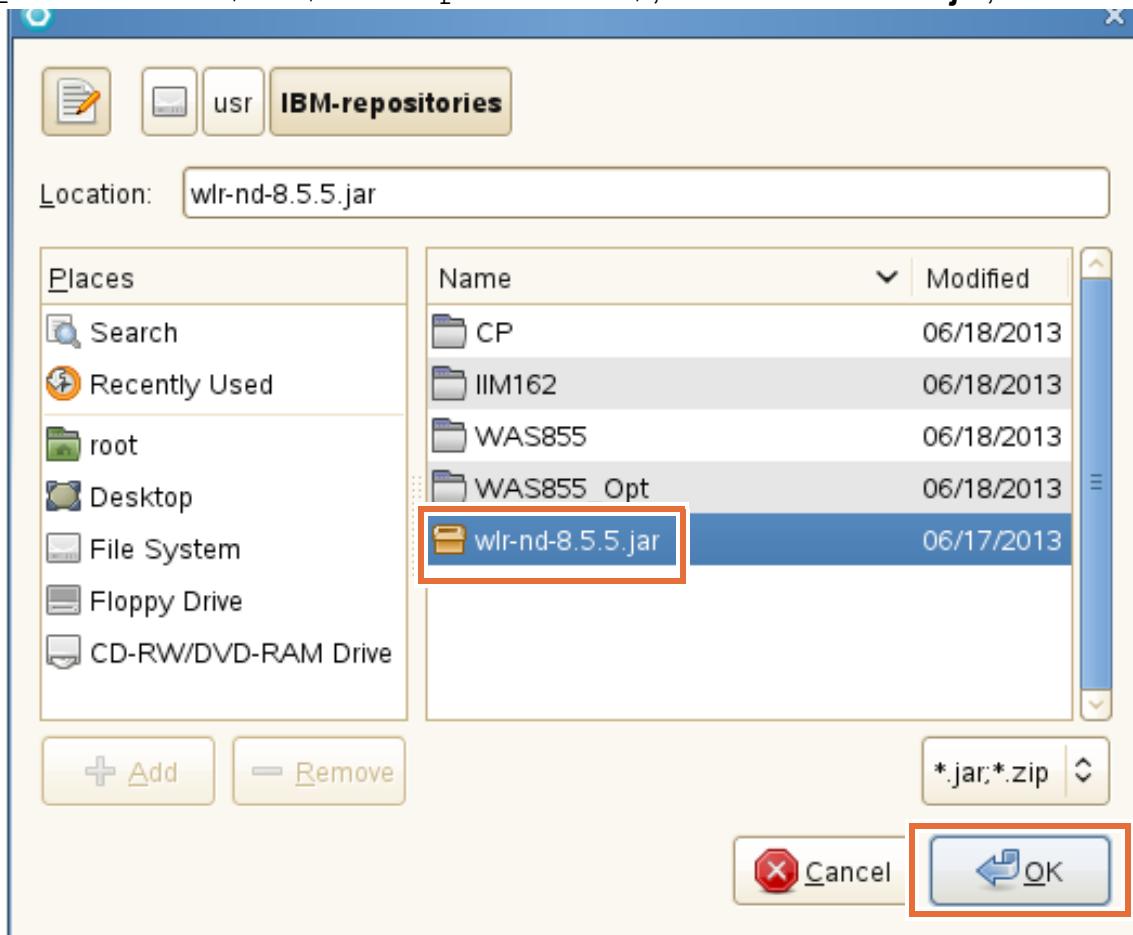
- __ d. Click **Download or install**.



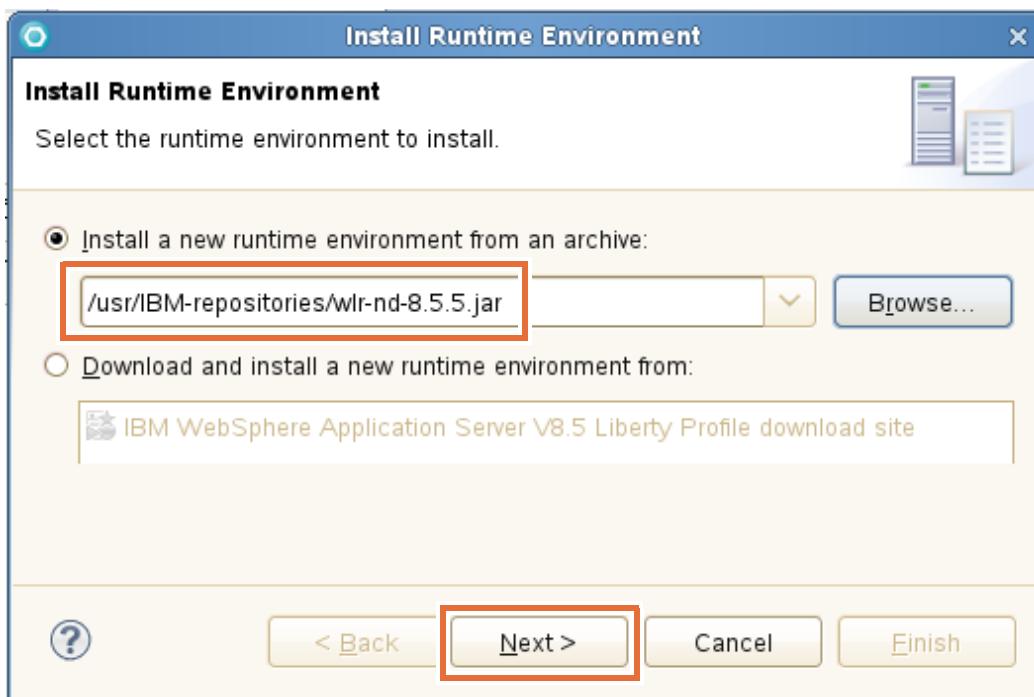
- __ e. Select **Install a new runtime environment from an archive**, and click **Browse...**.



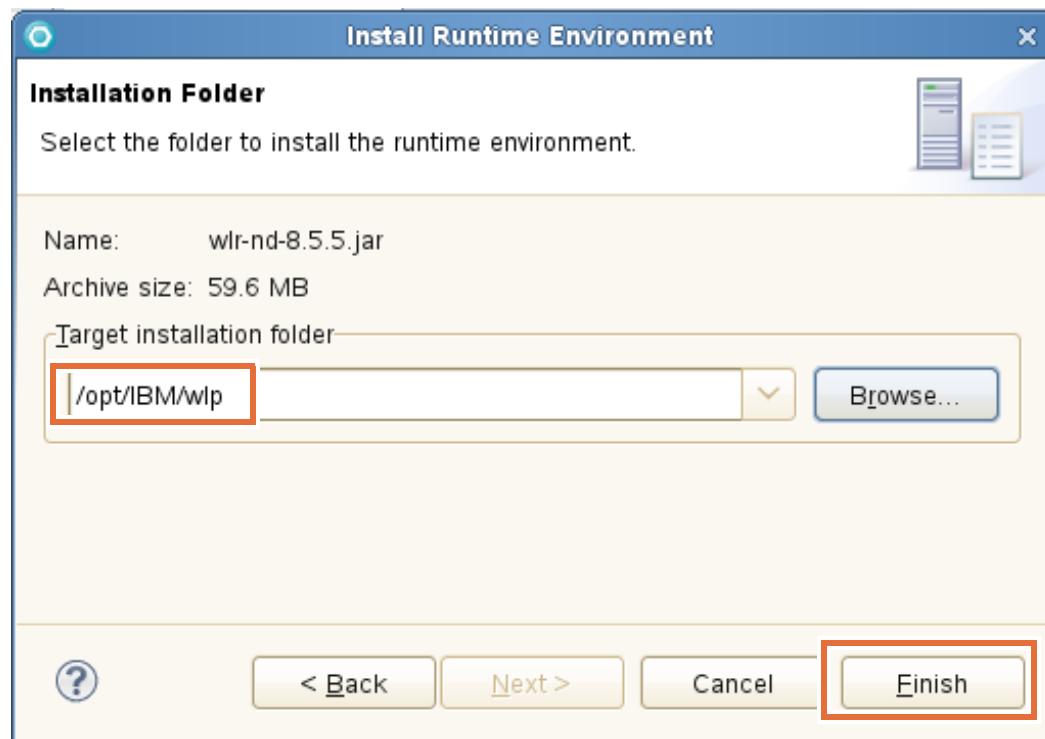
- f. Browse to /usr/IBM-repositories/, select **wlr-nd-8.5.5.jar**, and click **OK**.



- g. Verify the location of the archive, and click **Next**.



- __ h. Accept the license agreement and click **Next**.
- __ i. Enter the Target installation folder: /opt/IBM/wlp
Click **Finish**.

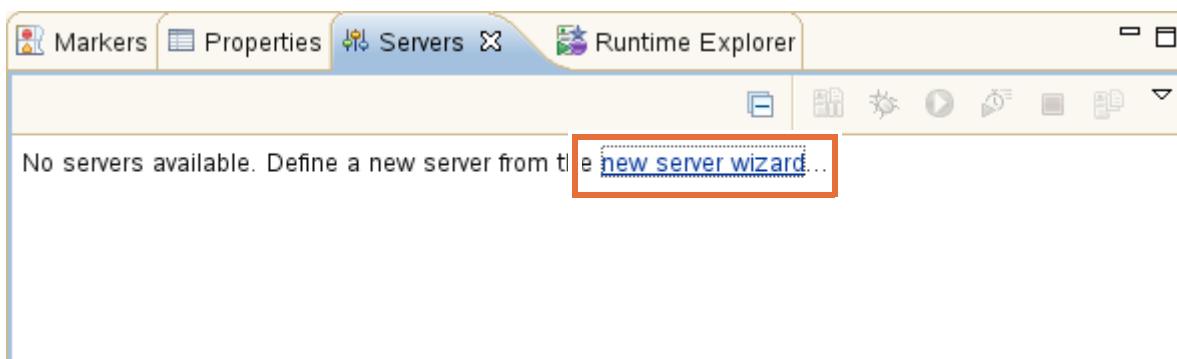


- __ j. The Liberty profile runtime environment is installed (uncompressed to the installation folder) and you are returned to the previous dialog box. Click **Finish**.
- __ k. Notice that WebSphere Application Server V8.5 Liberty Profile runtime is now listed in the Runtime Explorer tab.

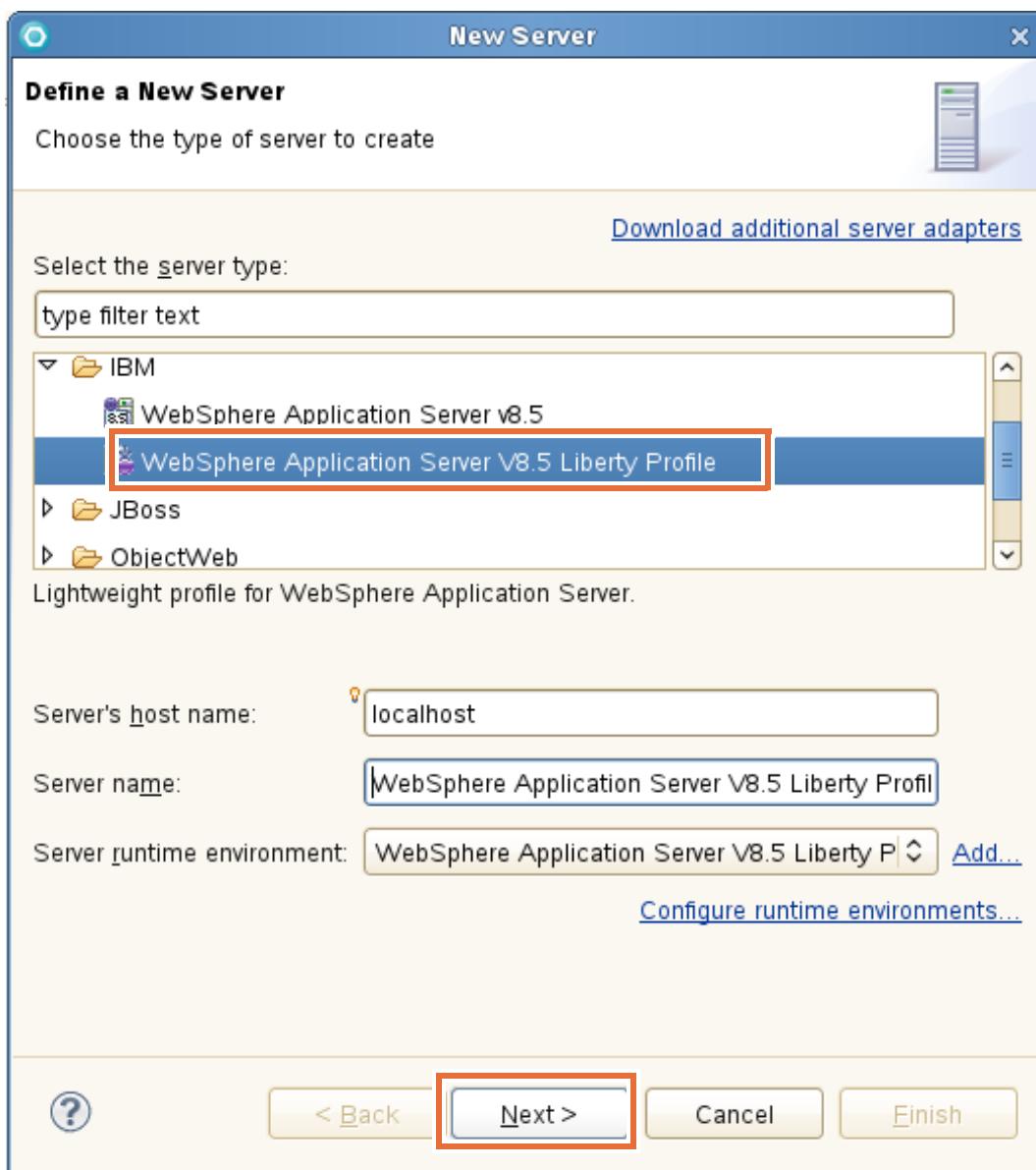


- __ l. Notice that though there is now a runtime, a Liberty profile server does not yet exist. Next, you define a Liberty profile server.

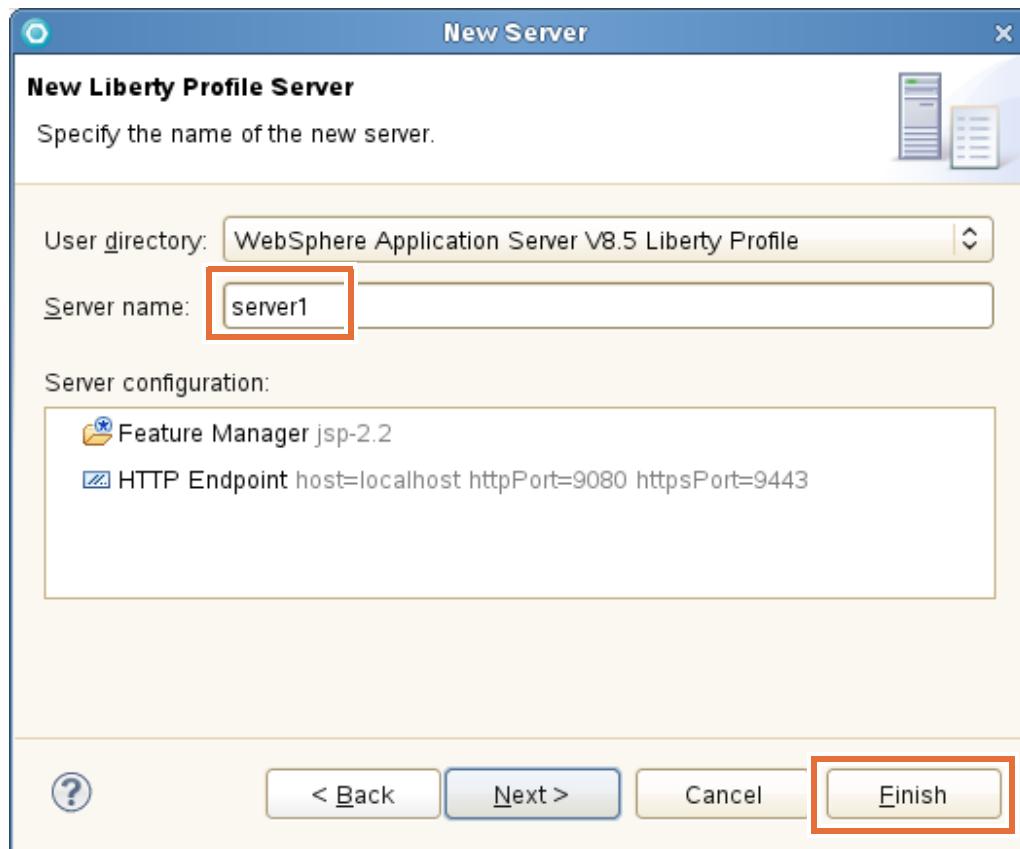
- 4. Define a Liberty profile server.
— a. Select the **Servers** tab, and click **new server wizard**.



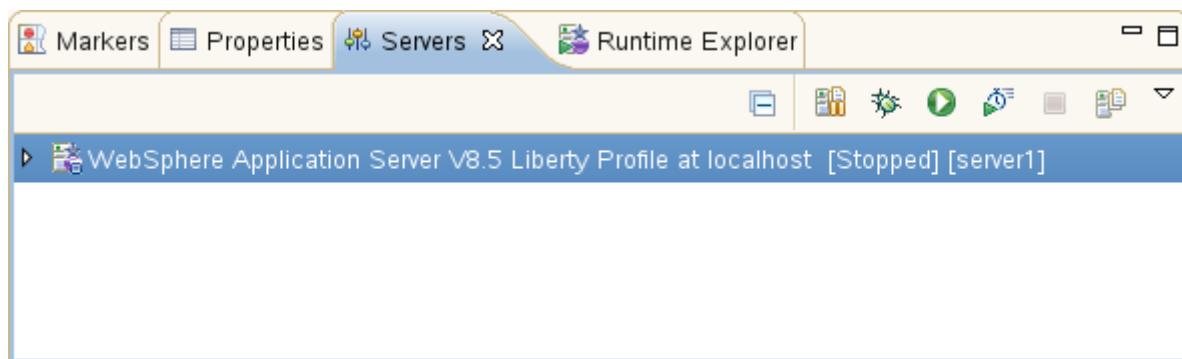
- b. Select **WebSphere Application Server V8.5 Liberty Profile**, and click **Next**.



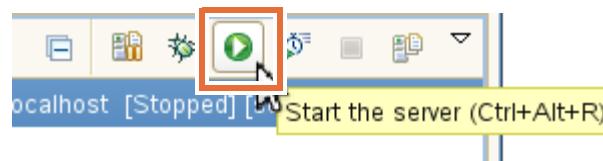
- __ c. Enter `server1` for the server name, and click **Finish**.



- __ d. A Liberty profile server instance is created. Select the **Servers** tab to see the listing for the new server. Click the server to see its status.



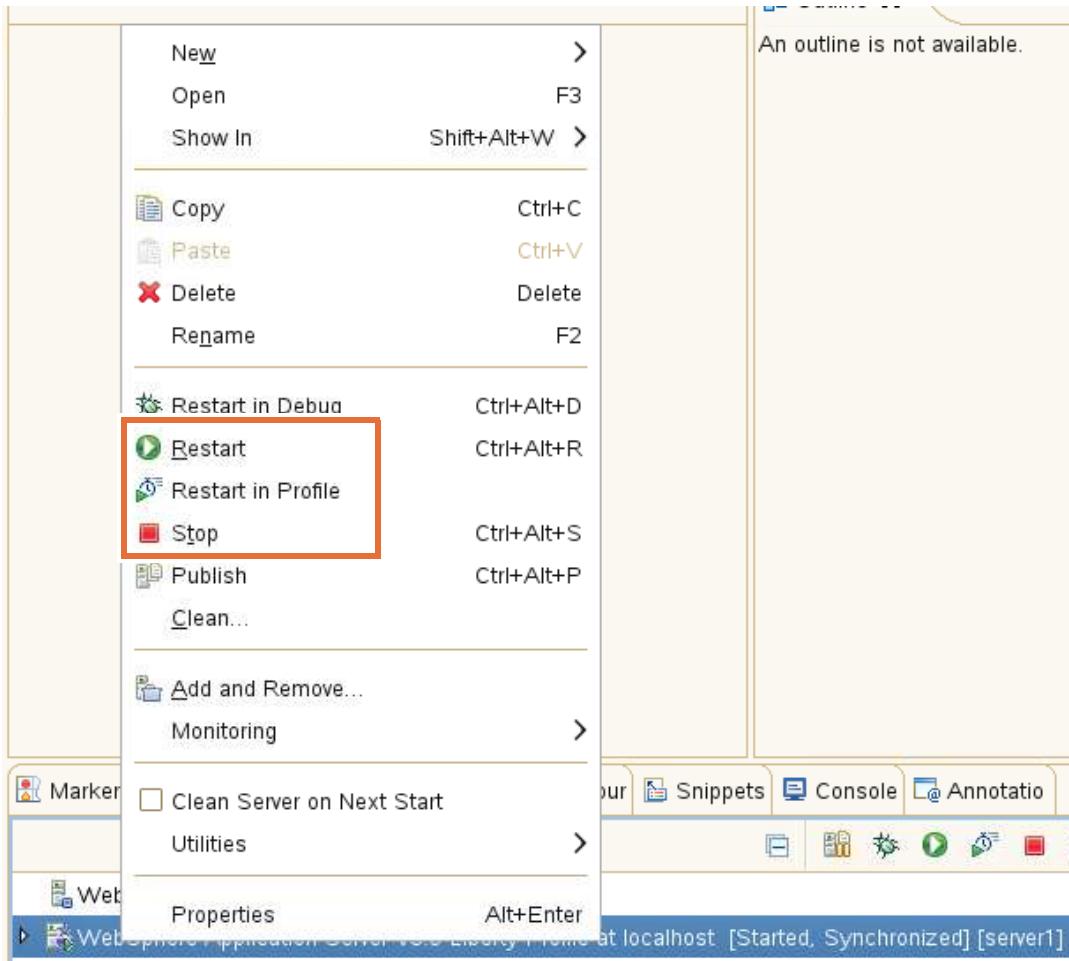
- __ e. Notice that the status of the server is Stopped. Start the server by selecting the server and clicking the **Start the server** icon.



- __ f. As the server starts up, the Console view shows the start time messages.



- __ g. You can also right-click the server and select **Start** (or **Restart** since it is already started) and **Stop** from the menu. Later in the exercise, you can use this menu to easily restart a server.



- __ h. Click the **Servers** tab and verify that server status is now Started.
 __ i. Stop server1 by clicking the **Stop** icon.

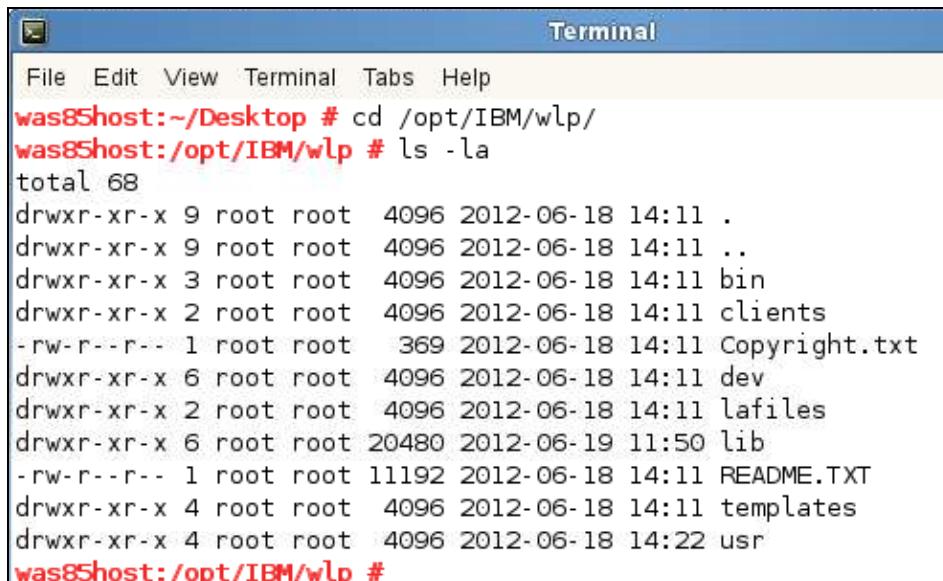
Section 3: Explore the Liberty profile directory

In this section, you explore the Liberty profile directory structure.

— 1. Explore the Liberty profile root directory.

- a. In a terminal window, navigate to /opt/IBM/wlp. Use the following command to get a listing of the directory:

```
ls -la
```



The screenshot shows a terminal window titled "Terminal". The window contains the following text:

```
File Edit View Terminal Tabs Help  
was85host:~/Desktop # cd /opt/IBM/wlp/  
was85host:/opt/IBM/wlp # ls -la  
total 68  
drwxr-xr-x 9 root root 4096 2012-06-18 14:11 .  
drwxr-xr-x 9 root root 4096 2012-06-18 14:11 ..  
drwxr-xr-x 3 root root 4096 2012-06-18 14:11 bin  
drwxr-xr-x 2 root root 4096 2012-06-18 14:11 clients  
-rw-r--r-- 1 root root 369 2012-06-18 14:11 Copyright.txt  
drwxr-xr-x 6 root root 4096 2012-06-18 14:11 dev  
drwxr-xr-x 2 root root 4096 2012-06-18 14:11 lafiles  
drwxr-xr-x 6 root root 20480 2012-06-19 11:50 lib  
-rw-r--r-- 1 root root 11192 2012-06-18 14:11 README.TXT  
drwxr-xr-x 4 root root 4096 2012-06-18 14:11 templates  
drwxr-xr-x 4 root root 4096 2012-06-18 14:22 usr  
was85host:/opt/IBM/wlp #
```



Information

The README.TXT file:

The information in this file is worth reading because it explains some of the key pieces of the Liberty profile. You can open the file in an editor by entering: gedit README.TXT

```
README.TXT X
WebSphere Application Server Version 8.5.5.0 Liberty Profile - (8.5.5.0-20130524-0951)

I. CONTROLLING WEBSPHERE APPLICATION SERVER V8.5 LIBERTY PROFILE

The bin directory contains a server script to help control the server process.
The script supports the following actions:

* create    -- creates a new server
* start     -- launches the server as a background process
* run       -- launches the server in the foreground
* debug     -- launches the server in the foreground with JVM debug options
* stop      -- stops a running server
* status    -- check to see if a specified server is running.
* package   -- packages server runtime and target server configuration/application(s) i
* dump      -- dump diagnostic information from the server into an archive
* javadump  -- dump diagnostic information from the server JVM
* version   -- displays the version of the server runtime
* help      -- get command-line/script help, including descriptions of additional optio
```

This file describes many useful aspects of the Liberty profile, including the directory structure, the installation processes, and variable definitions.

2. Explore the bin directory.

- __ a. Change to the bin directory and use the ls -la command again to get another directory listing.

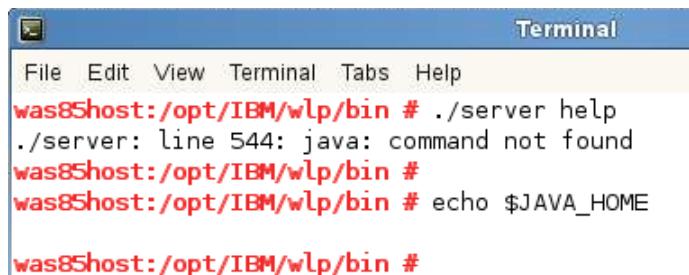
```
-rwxr--r-- 1 root root 3547 2012-06-18 14:11 securityUtility
-rw-r--r-- 1 root root 1966 2012-06-18 14:11 securityUtility.bat
-rwxr--r-- 1 root root 22538 2012-06-18 14:11 server
-rw-r--r-- 1 root root 18242 2012-06-18 14:11 server.bat
drwxr-xr-x 2 root root 4096 2012-06-18 14:11 tools
was85host:/opt/IBM/wlp/bin #
```

- __ b. Notice the server command. This command is used to start and stop the application server.
- __ c. Enter the following command to view the usage of the server command:

`./server help`

- ___ d. Notice that this command produces an error. The reason is that your terminal window does not have a `JAVA_HOME` definition. Enter the following command to confirm that the `JAVA_HOME` variable is empty:

```
echo $JAVA_HOME
```

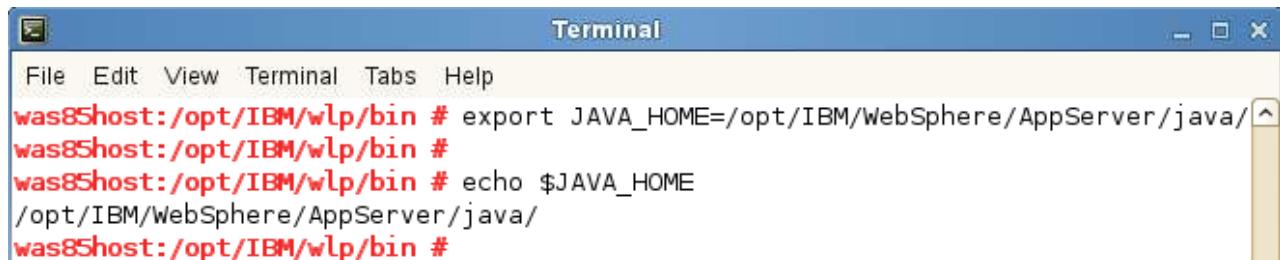


A screenshot of a terminal window titled "Terminal". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The main area shows the following text:
was85host:/opt/IBM/wlp/bin # ./server help
. ./server: line 544: java: command not found
was85host:/opt/IBM/wlp/bin #
was85host:/opt/IBM/wlp/bin # echo \$JAVA_HOME

was85host:/opt/IBM/wlp/bin #

- ___ e. Define the `JAVA_HOME` variable with the following command:

```
export JAVA_HOME=/opt/IBM/WebSphere/AppServer/java
```



A screenshot of a terminal window titled "Terminal". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The main area shows the following text:
was85host:/opt/IBM/wlp/bin # export JAVA_HOME=/opt/IBM/WebSphere/AppServer/java/
was85host:/opt/IBM/wlp/bin #
was85host:/opt/IBM/wlp/bin # echo \$JAVA_HOME
/opt/IBM/WebSphere/AppServer/java/
was85host:/opt/IBM/wlp/bin #



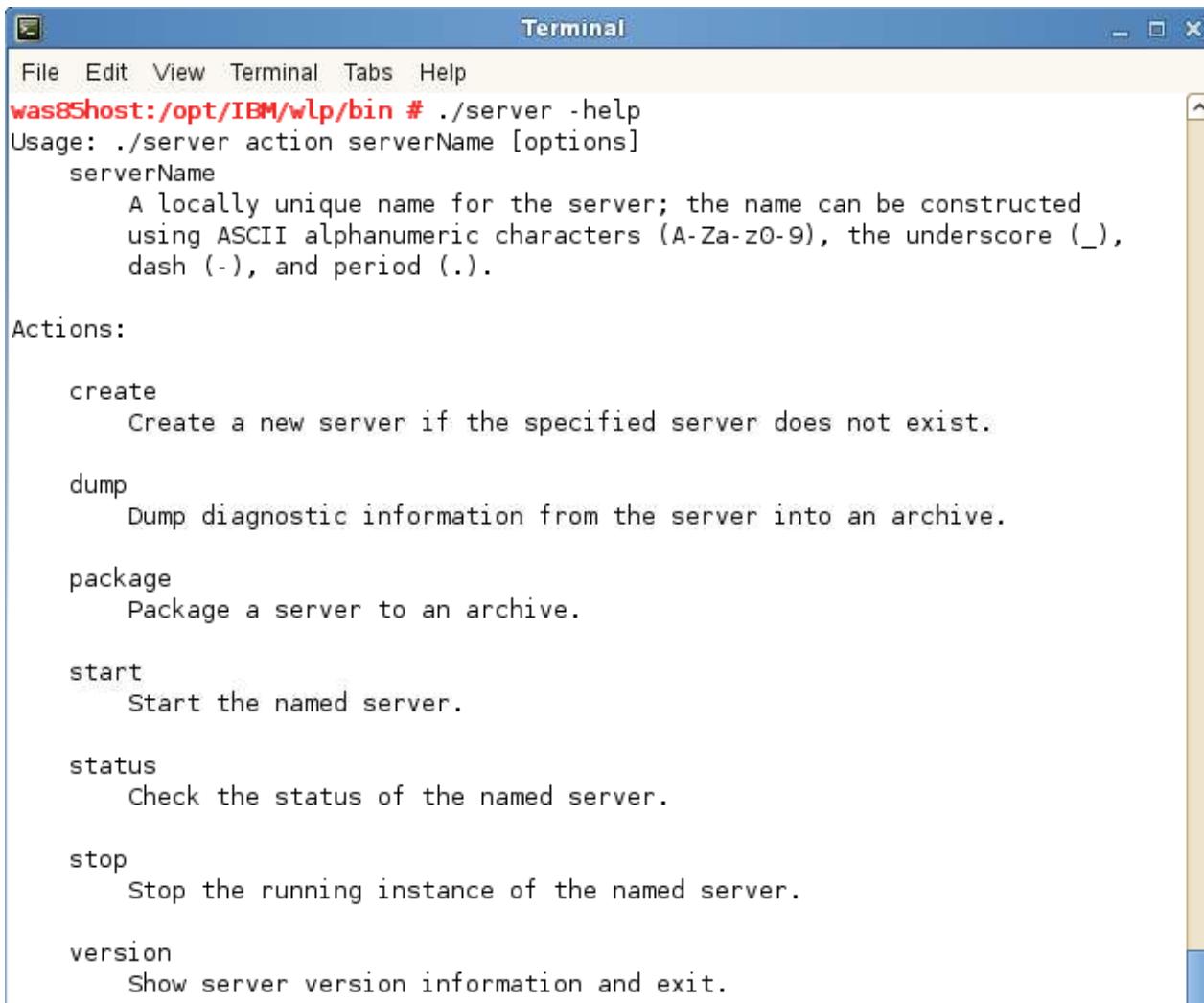
Important

This variable definition exists only in this terminal window. To define it in more terminal windows, you must either add the definition to your user startup scripts (for example: `.profile` or `.bashrc`) or use the same command in each new terminal window.

```
export JAVA_HOME=/opt/IBM/WebSphere/AppServer/java/
```

- __ f. Enter the following command to get the `server` command help:

```
./server help
```



```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/wlp/bin # ./server -help
Usage: ./server action serverName [options]
  serverName
    A locally unique name for the server; the name can be constructed
    using ASCII alphanumeric characters (A-Za-z0-9), the underscore (_),
    dash (-), and period (.).

Actions:
  create
    Create a new server if the specified server does not exist.

  dump
    Dump diagnostic information from the server into an archive.

  package
    Package a server to an archive.

  start
    Start the named server.

  status
    Check the status of the named server.

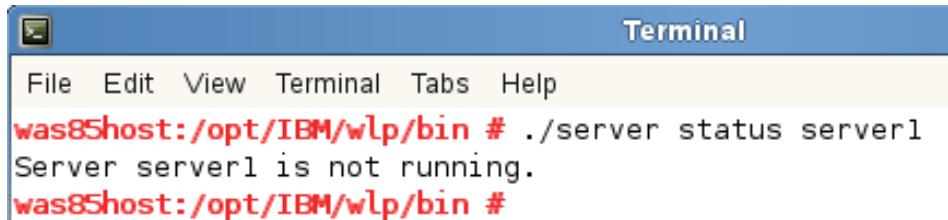
  stop
    Stop the running instance of the named server.

  version
    Show server version information and exit.
```

- __ 3. Explore the `server` command.

- __ a. Use the `server` command to check the status of `server1` with the following command:

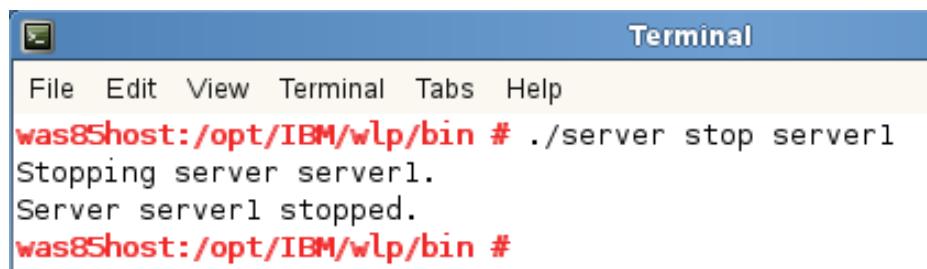
```
./server status server1
```



```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/wlp/bin # ./server status server1
Server server1 is not running.
was85host:/opt/IBM/wlp/bin #
```

- ___ b. If your server is running, use the following command to stop server1:

```
./server stop server1
```

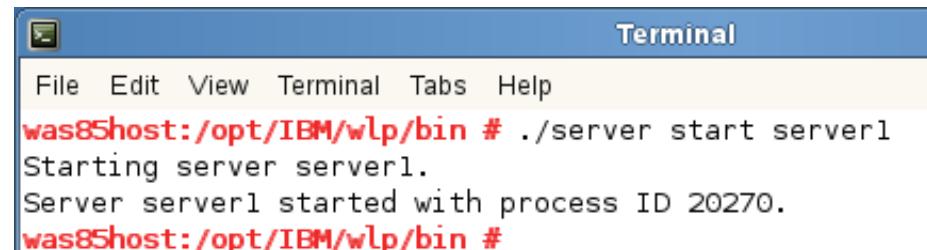


Terminal

```
File Edit View Terminal Tabs Help
was85host:/opt/IBM/wlp/bin # ./server stop server1
Stopping server server1.
Server server1 stopped.
was85host:/opt/IBM/wlp/bin #
```

- ___ c. Next, start server1 again with the following command:

```
./server start server1
```



Terminal

```
File Edit View Terminal Tabs Help
was85host:/opt/IBM/wlp/bin # ./server start server1
Starting server server1.
Server server1 started with process ID 20270.
was85host:/opt/IBM/wlp/bin #
```

- ___ d. Check the version of the runtime with the following command:

```
./server version
```



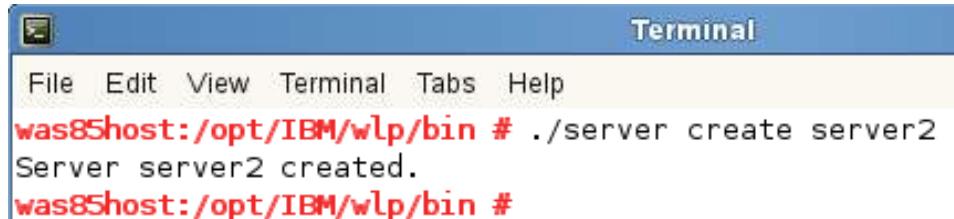
Terminal

```
File Edit View Terminal Tabs Help
was85host:/opt/IBM/wlp/bin # ./server version
WebSphere Application Server 8.5.5.0 (1.0.3.20130524-0951) on IBM J9 VM, version
pxi3260_26sr5fp1ifix-20130408_02 (SR5 FP1) (en_US)
was85host:/opt/IBM/wlp/bin #
```

- ___ 4. Using the command line, create another server instance.

- ___ a. In the same terminal window, use the following command to create server2:

```
./server create server2
```



Terminal

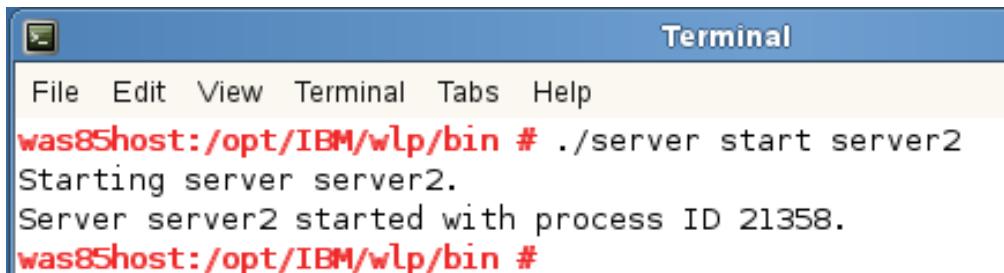
```
File Edit View Terminal Tabs Help
was85host:/opt/IBM/wlp/bin # ./server create server2
Server server2 created.
was85host:/opt/IBM/wlp/bin #
```

**Important**

During the creation of server2, default ports (9080 and 9443) are assigned to server2. This default action can cause problems, since these ports are the same ports that server1 uses.

It is up to the administrator to modify the server ports appropriately to ensure that they are unique if they are intended to be started at the same time. The `server.xml` file for server2 can be directly edited to change the ports for http and https.

- __ b. Now that server2 exists, go ahead and **start** it.



A screenshot of a terminal window titled "Terminal". The menu bar includes "File", "Edit", "View", "Terminal", "Tabs", and "Help". The command entered is `./server start server2`. The output shows the server starting and being assigned a process ID of 21358.

```
File Edit View Terminal Tabs Help
was85host:/opt/IBM/wlp/bin # ./server start server2
Starting server server2.
Server server2 started with process ID 21358.
was85host:/opt/IBM/wlp/bin #
```

**Important**

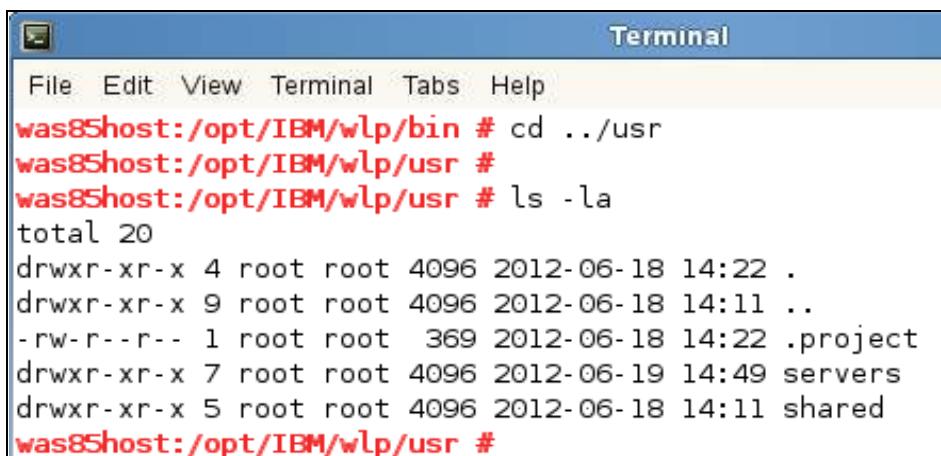
Although server2 seems to start correctly, if you look in the `console.log` file (`/opt/IBM/wlp/usr/servers/server2/logs/console.log`) you see a port conflict error:

```
Launching server2 (wlp-1.0.0.20120428-1251/websphere-kernel_1.0.0) on
IBM J9 VM, version pxi3260_26sr2ifx-20120419_02 (SR2) (en_US)
[AUDIT    ] CWWKE0001I: The server server2 has been launched.
[ERROR    ] CWWKO0221E: TCP Channel defaultHttpEndpoint initialization
did not succeed. The socket bind did not succeed for host localhost
and port 9080. The port might already be in use.
[AUDIT    ] CWWKZ0058I: Monitoring dropins for applications.
[AUDIT    ] CWWKF0011I: The server server2 is ready to run a smarter
planet.
```

- __ c. **Stop** server2 with the command: `./server stop server2`

5. Explore the configuration files.

- a. Change to the /opt/IBM/wlp/usr directory and get a directory listing.

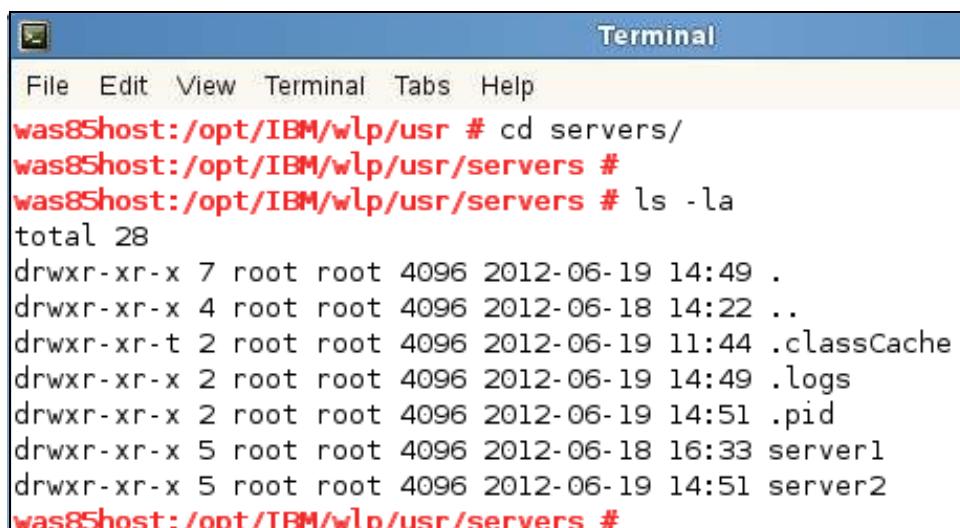


A terminal window titled "Terminal" showing a directory listing. The command "cd .." was run to move up one level from the current directory. Then, "ls -la" was run to list all files and directories in the "/opt/IBM/wlp/usr" directory. The output shows several files and directories, including ".project", "servers", and "shared".

```
File Edit View Terminal Tabs Help  
was85host:/opt/IBM/wlp/bin # cd ../usr  
was85host:/opt/IBM/wlp/usr #  
was85host:/opt/IBM/wlp/usr # ls -la  
total 20  
drwxr-xr-x 4 root root 4096 2012-06-18 14:22 .  
drwxr-xr-x 9 root root 4096 2012-06-18 14:11 ..  
-rw-r--r-- 1 root root 369 2012-06-18 14:22 .project  
drwxr-xr-x 7 root root 4096 2012-06-19 14:49 servers  
drwxr-xr-x 5 root root 4096 2012-06-18 14:11 shared  
was85host:/opt/IBM/wlp/usr #
```

Notice the `servers` directory. This directory is where the configuration files for each of the servers are stored.

- b. Change directory to the `servers` folder and get another directory listing.

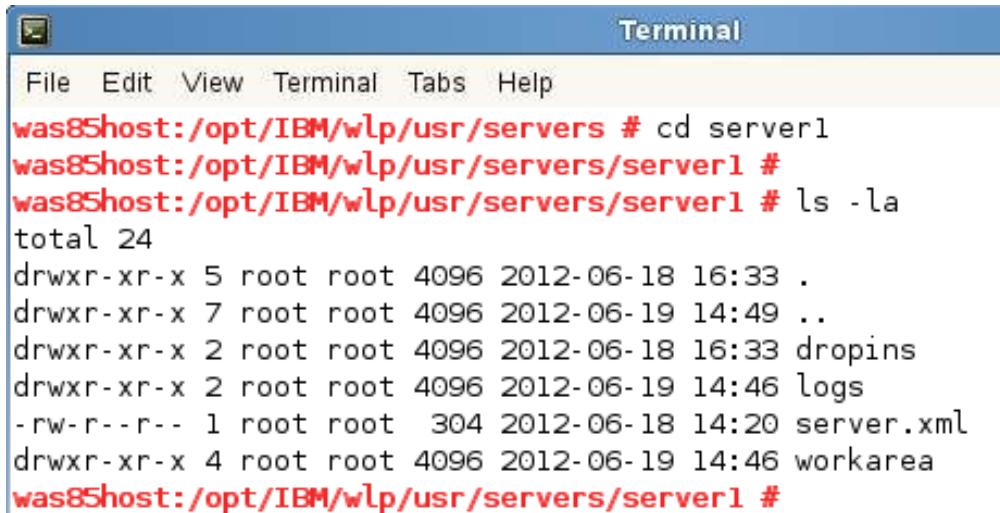


A terminal window titled "Terminal" showing a directory listing. The user first changes to the "servers" directory using "cd servers/". Then, "ls -la" is run to list the contents of the "servers" directory. The output shows sub-directories for "server1" and "server2".

```
File Edit View Terminal Tabs Help  
was85host:/opt/IBM/wlp/usr # cd servers/  
was85host:/opt/IBM/wlp/usr/servers #  
was85host:/opt/IBM/wlp/usr/servers # ls -la  
total 28  
drwxr-xr-x 7 root root 4096 2012-06-19 14:49 .  
drwxr-xr-x 4 root root 4096 2012-06-18 14:22 ..  
drwxr-xr-t 2 root root 4096 2012-06-19 11:44 .classCache  
drwxr-xr-x 2 root root 4096 2012-06-19 14:49 .logs  
drwxr-xr-x 2 root root 4096 2012-06-19 14:51 .pid  
drwxr-xr-x 5 root root 4096 2012-06-18 16:33 server1  
drwxr-xr-x 5 root root 4096 2012-06-19 14:51 server2  
was85host:/opt/IBM/wlp/usr/servers #
```

Notice that there are directories for both server1 and server2.

- ___ c. Change directory to the `server1` directory and get another directory listing.



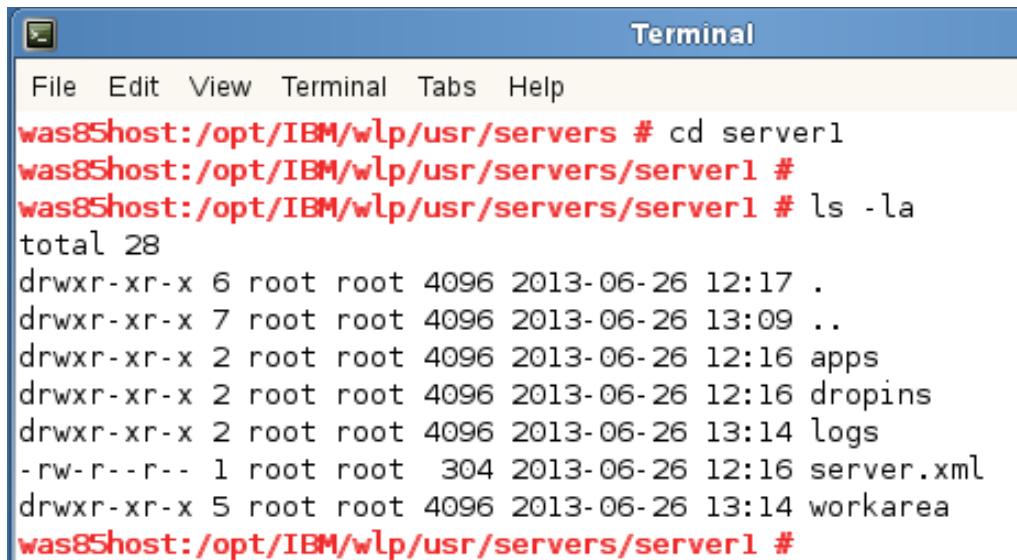
A screenshot of a terminal window titled "Terminal". The menu bar includes "File", "Edit", "View", "Terminal", "Tabs", and "Help". The command line shows the user navigating to the `server1` directory and listing its contents with `ls -la`. The output shows a total of 24 files and directories, including `dropins`, `logs`, and `server.xml`.

```
was85host:/opt/IBM/wlp/usr/servers # cd server1
was85host:/opt/IBM/wlp/usr/servers/server1 #
was85host:/opt/IBM/wlp/usr/servers/server1 # ls -la
total 24
drwxr-xr-x 5 root root 4096 2012-06-18 16:33 .
drwxr-xr-x 7 root root 4096 2012-06-19 14:49 ..
drwxr-xr-x 2 root root 4096 2012-06-18 16:33 dropins
drwxr-xr-x 2 root root 4096 2012-06-19 14:46 logs
-rw-r--r-- 1 root root 304 2012-06-18 14:20 server.xml
drwxr-xr-x 4 root root 4096 2012-06-19 14:46 workarea
was85host:/opt/IBM/wlp/usr/servers/server1 #
```

The `dropins` directory is used for application installs. The `logs` directory contains the application server logs. Feel free to explore those directories.

- ___ d. Use the following command to display the contents of the `server.xml` file:

```
more server.xml
```



A screenshot of a terminal window titled "Terminal". The menu bar includes "File", "Edit", "View", "Terminal", "Tabs", and "Help". The command line shows the user navigating to the `server1` directory and displaying the contents of `server.xml` with `more`. The output shows the XML configuration for the server.

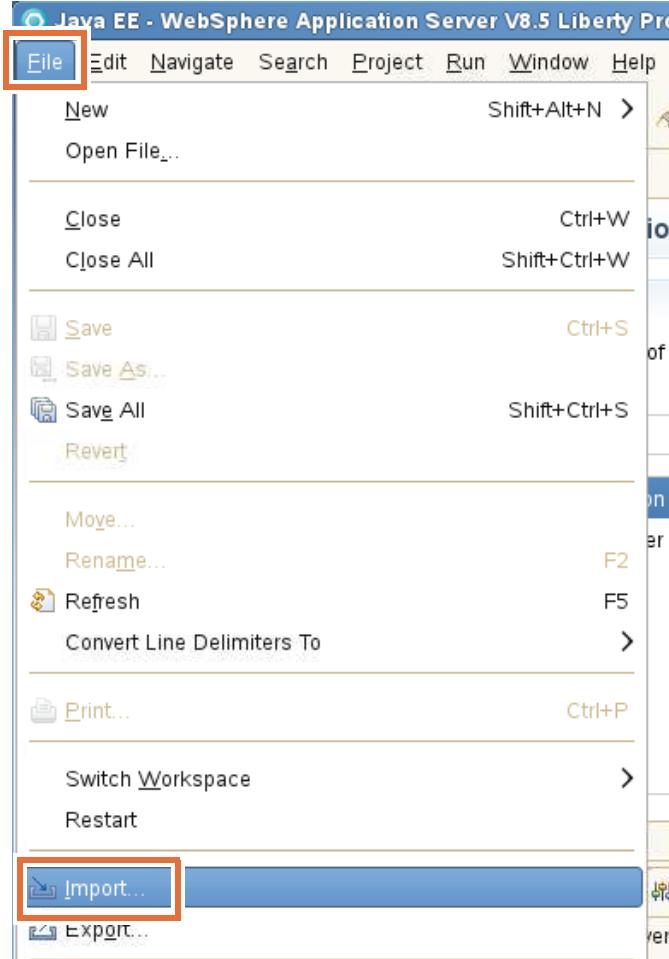
```
was85host:/opt/IBM/wlp/usr/servers # cd server1
was85host:/opt/IBM/wlp/usr/servers/server1 #
was85host:/opt/IBM/wlp/usr/servers/server1 # ls -la
total 28
drwxr-xr-x 6 root root 4096 2013-06-26 12:17 .
drwxr-xr-x 7 root root 4096 2013-06-26 13:09 ..
drwxr-xr-x 2 root root 4096 2013-06-26 12:16 apps
drwxr-xr-x 2 root root 4096 2013-06-26 12:16 dropins
drwxr-xr-x 2 root root 4096 2013-06-26 13:14 logs
-rw-r--r-- 1 root root 304 2013-06-26 12:16 server.xml
drwxr-xr-x 5 root root 4096 2013-06-26 13:14 workarea
was85host:/opt/IBM/wlp/usr/servers/server1 #
```

- ___ e. Notice that the `server.xml` file initially contains minimal configuration, a single JSP feature, and the http endpoint configuration.

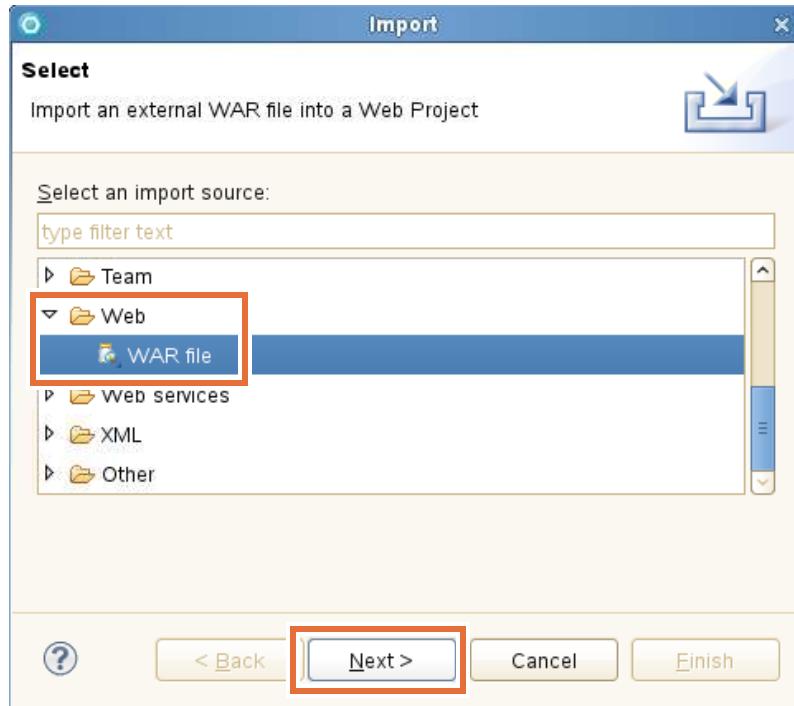
Section 4: Use IBM Assembly and Deploy Tools to deploy an application

In this section, you use IBM Assembly and Deploy Tools to deploy a simple Hello World application.

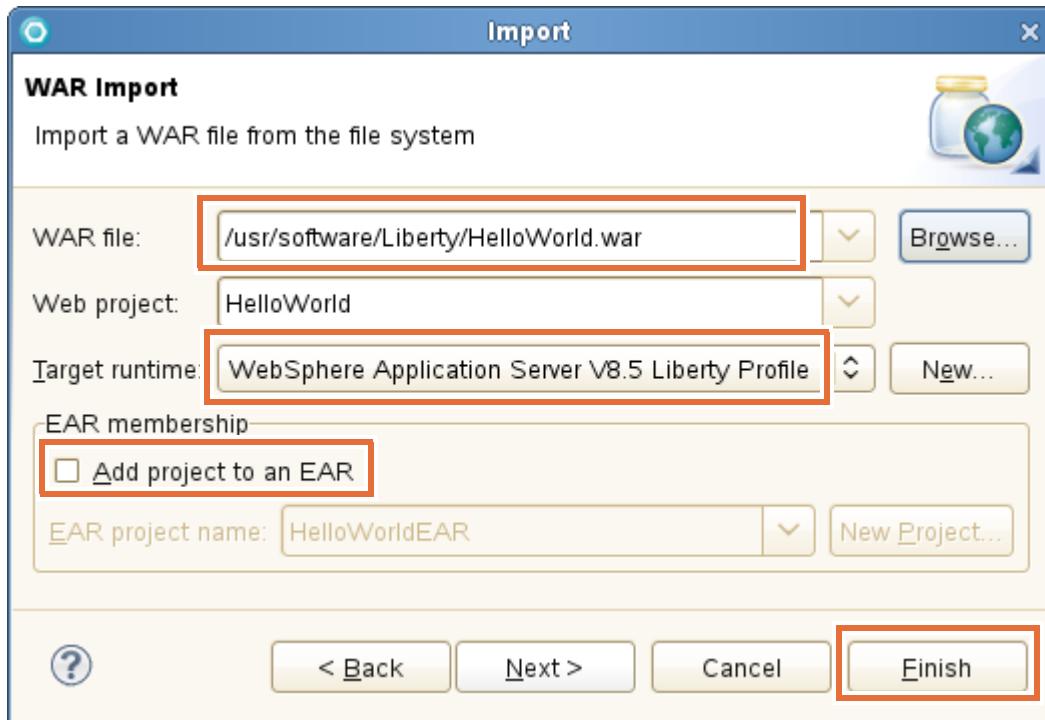
- 1. Use the developer tool to import the Hello World application.
 - a. In the IBM Assembly and Deploy Tools window, click **File > Import**.



- __ b. Click **Web > WAR file** and click **Next**.



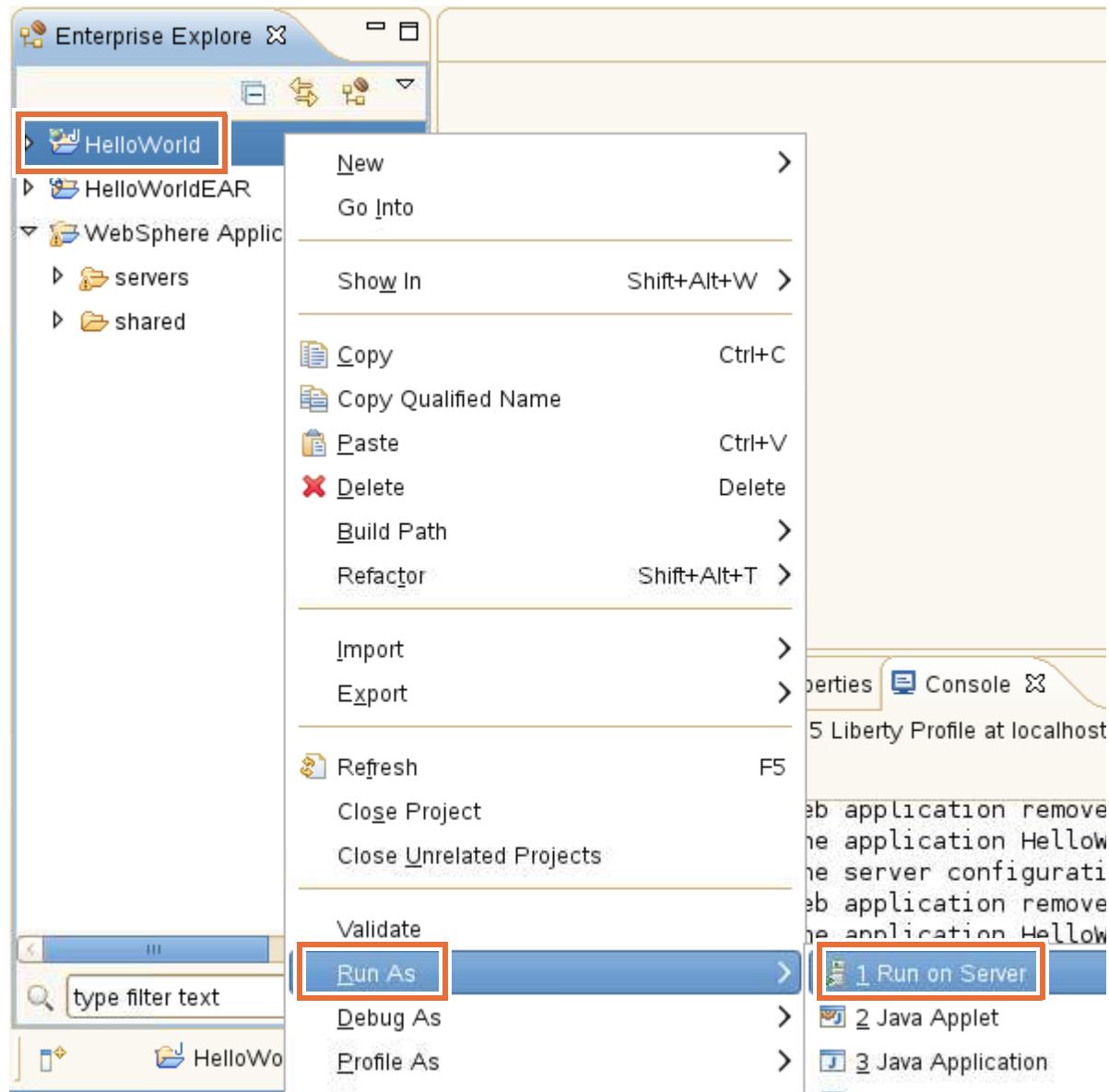
- __ c. Browse to the WAR file `/usr/software/Liberty/HelloWorld.war` and click **OK**. Ensure that the **Target runtime** is set to **WebSphere Application Server V8.5 Liberty Profile**, clear the box for **Add projects to an EAR**, and click **Finish**.



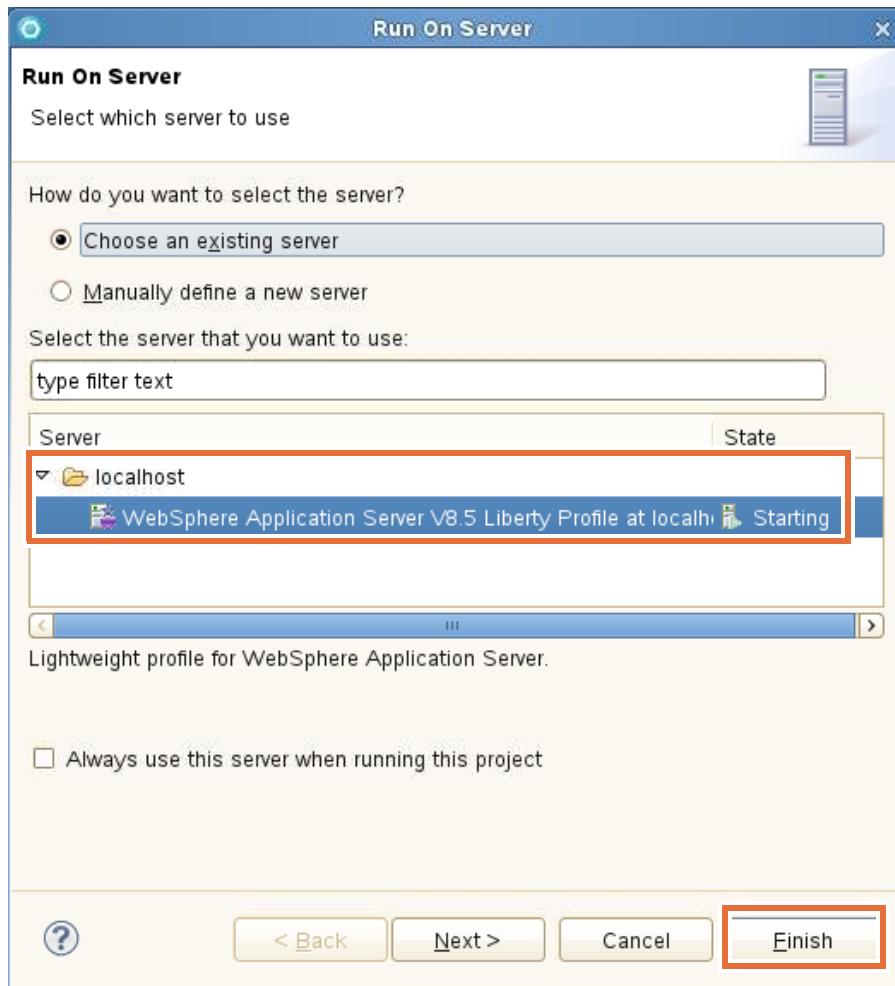
- __ d. If asked to Save changes, click **Yes**.
__ e. If asked to change to the **Web perspective**, click **Yes**.

2. Run the application in the server.

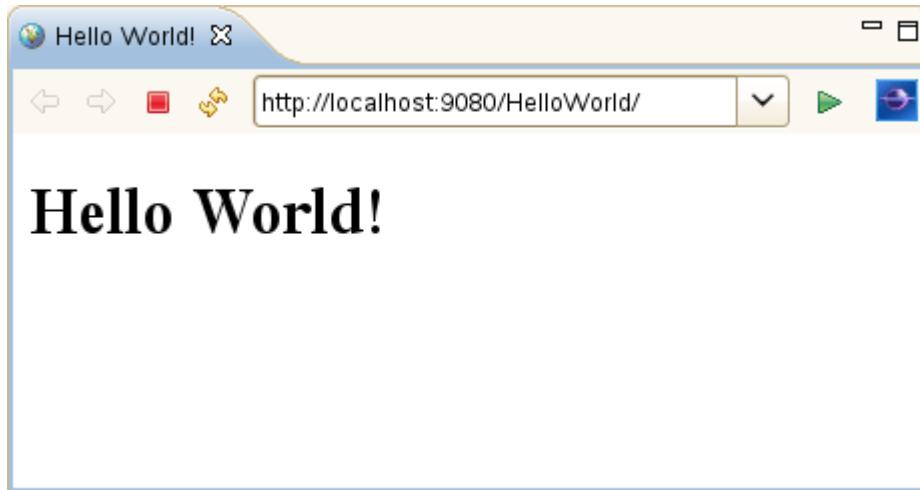
a. Notice that HelloWorld now is listed in the Enterprise Explorer. Right-click it, and select **Run As > Run on Server**.



- __ b. In the dialog box, ensure that the **Liberty Profile** is selected and click **Finish**.



- __ c. An internal browser window opens in the developer tool and displays the Hello World page at URL:
<http://localhost:9080>HelloWorld/>



- ___ d. Look at the **Console** tab, and notice the log messages that indicate that the application is started. Also, notice that the log messages include a hyperlink to the Hello World application.

```
Application Server V8.5 Liberty Profile at localhost [server1] (Jun 19, 2012 11:11:10 PM)
CwWKG0017I: The server configuration was successfully updated in 0.4 seconds.
CwKG0016I: Starting server configuration update.
CwKG0017I: The server configuration was successfully updated in 0.3 seconds.
CwKG0016I: Starting server configuration update.
CwKG0017I: Web application available (default_host): http://localhost:9080/HelloWorld/\*
CwKG0017I: Application HelloWorld started in 0.5 seconds.
CwKG0017I: The server configuration was successfully updated in 0.52 seconds.
```

- ___ e. **Save** the configuration by entering **Ctrl-S**.
- ___ 3. Access the Hello World application from an external browser.
- ___ a. Open a Firefox browser and browse to the following URL:

`http://localhost:9080/HelloWorld/`



Hello World!

Section 5: Deploy an application through the `dropins` directory

This section shows you how to deploy an application through the `dropins` directory. This approach allows developers to copy a WAR or EAR file into a monitored directory, and the application is automatically deployed.

- ___ 1. Copy the `HelloWorldDropin.war` file into the `dropins` directory.
- ___ a. In a terminal window, use the `cp` command to copy `/usr/software/Liberty/HelloWorldDropin.war` into the `/opt/IBM/wlp/usr/servers/server1/dropins` directory.

```
File Edit View Terminal Tabs Help
was85host:/opt/IBM/wlp/usr/servers/server1 # cp /usr/software/Liberty/HelloWorldDropin.war /opt/IBM/wlp/usr/servers/server1/dropins/
was85host:/opt/IBM/wlp/usr/servers/server1 #
```

- __ 2. View the console messages that show the application is deployed.
- __ a. Examine the log messages in the console view. Notice the entries for the **HelloWorldDropin** application.



The screenshot shows the WebSphere Application Server V8.5 Liberty Profile interface. The 'Console' tab is selected. The log output is as follows:

```
CWKG0016I: Starting server configuration update.  
CWWKT0016I: Web application available (default_host): http://localhost:9080>HelloWorld/\*  
CWWKZ0001I: Application HelloWorld started in 0.5 seconds.  
CWWKG0017I: The server configuration was successfully updated in 0.52 seconds.  
CWWKT0016I: Web application available (default_host) : http://localhost:9080>HelloWorldDropin/\*  
CWWKZ0001I: Application HelloWorldDropin started in 0.0 seconds.
```

- __ 3. Access the new application through both the internal and external browsers.
- __ a. Click the link in the console view for:
- `http://localhost:9080>HelloWorldDropin/*`
- __ b. An internal browser is displayed.



- __ c. Open a Firefox browser and browse to the following URL:

`http://localhost:9080>HelloWorldDropin/`

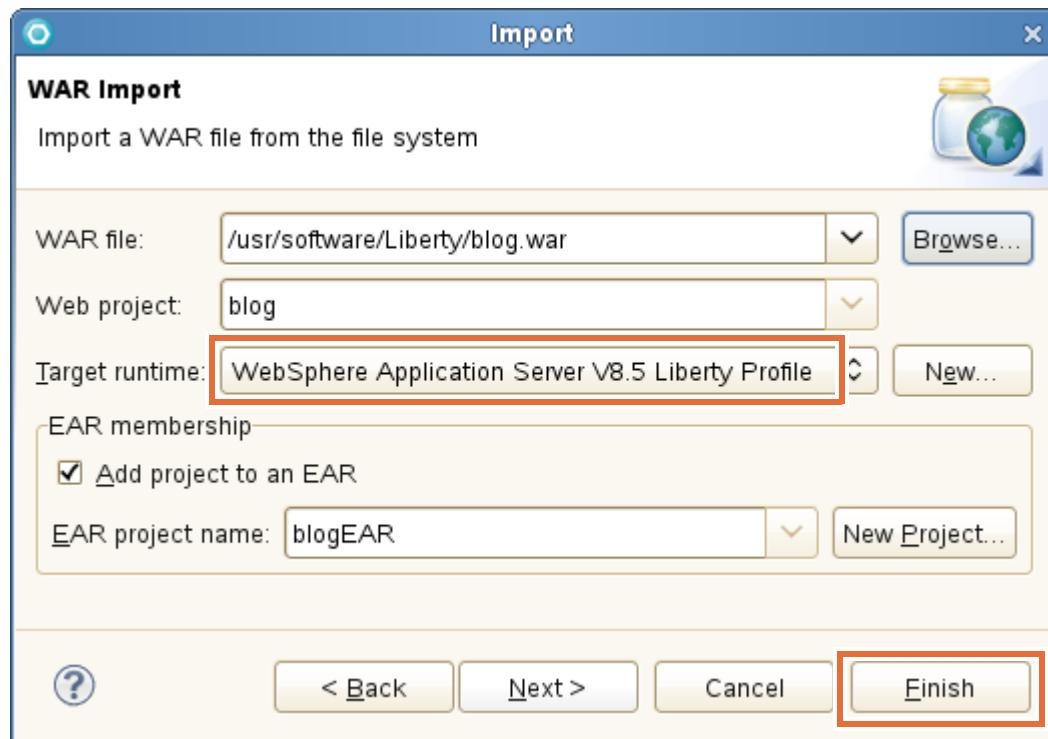


- __ d. Close any Firefox browser windows and any IBM Assembly and Deploy Tools internal browsers.

Section 6: Deploy the blog application with a data source

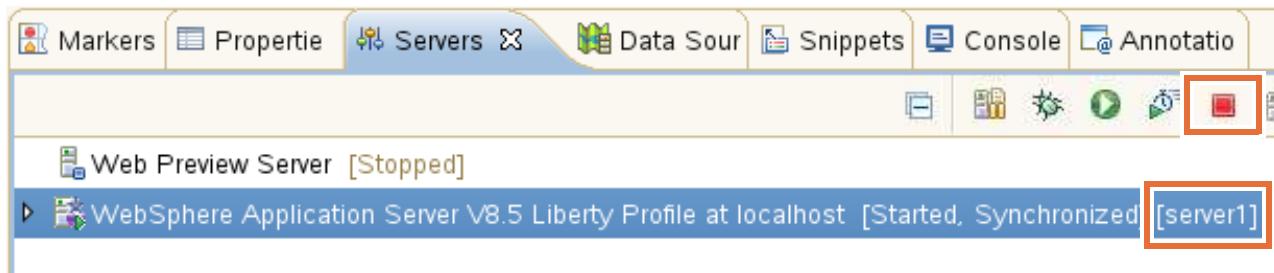
In this section, you deploy an application that requires a data source. You learn how to use IBM Assembly and Deploy Tools to add the features and configuration that support access to a Derby database.

- 1. Import the blog application.
 - a. Click **File > Import**.
 - b. In the Import wizard, click **Web > WAR file** and click **Next**.
 - c. For the WAR file, browse to `/usr/software/Liberty/blog.war` and click **OK**.
 - d. Ensure that the **Target runtime** is **WebSphere Application Server V8.5 Liberty Profile** and accept the defaults for the other fields.

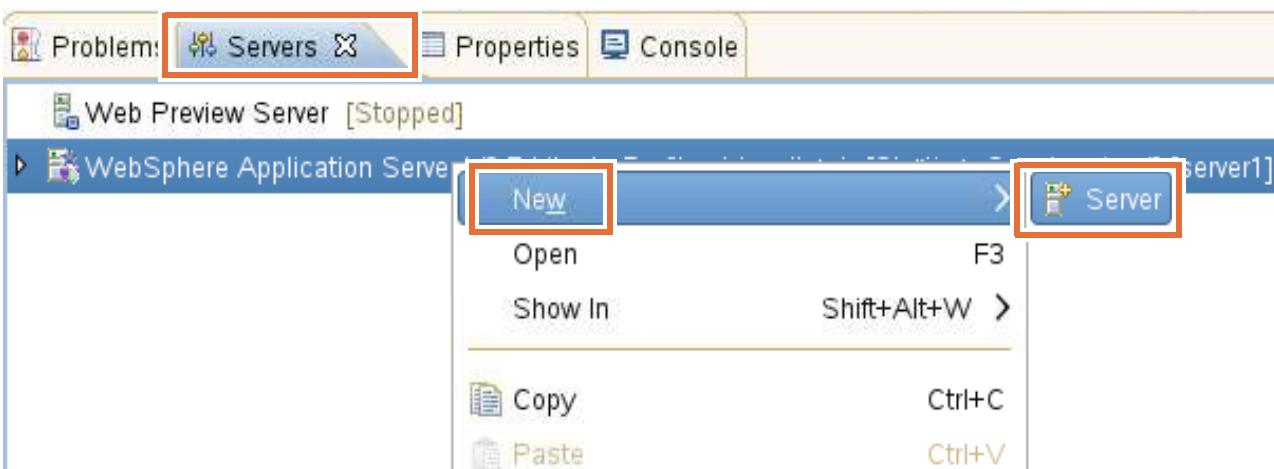


- e. Click **Finish** to complete the import. Change to the Web perspective when you are prompted.

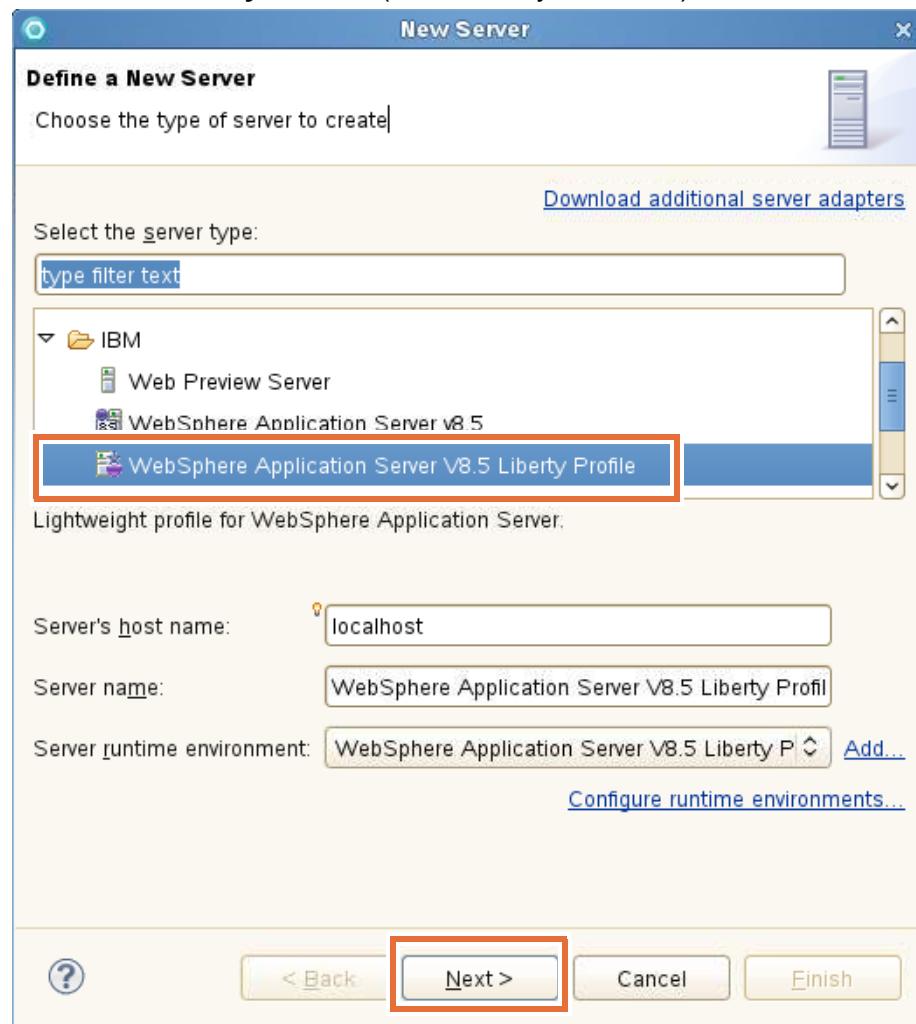
- __ 2. Create a server for running the blog application. To avoid port conflict issues, make sure server1 is stopped.
- __ a. From the **Servers** view, select **server1** and click the **Stop** icon (red square).



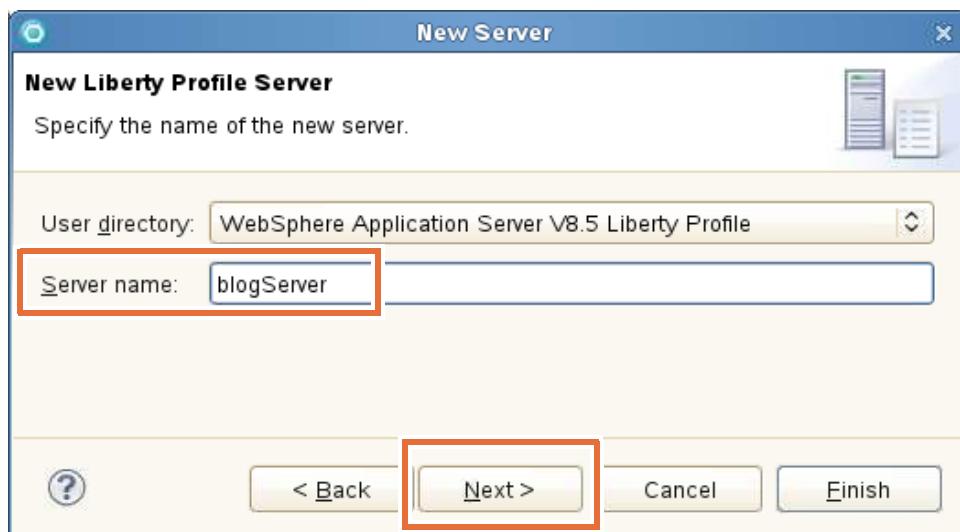
- __ 3. Create a server called blogServer.
- __ a. From the **Servers** view, right-click the **Liberty Profile** and select **New > Server**.



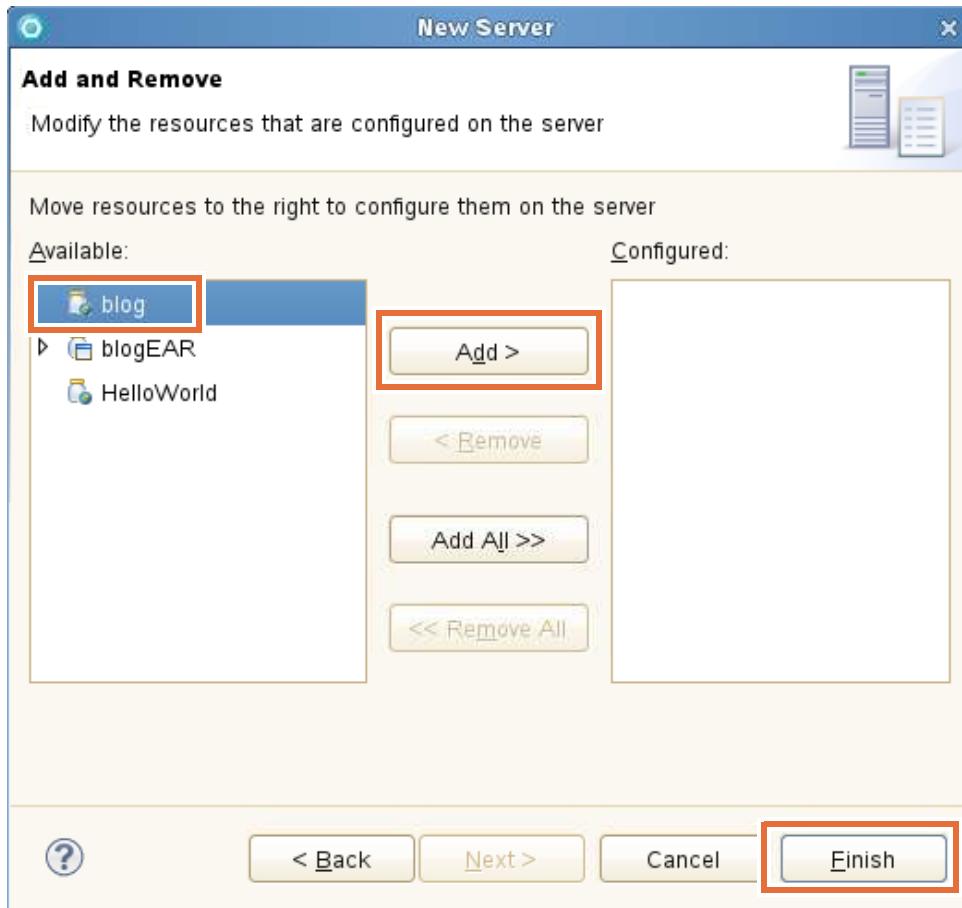
- b. In the **Define a New Server** wizard, select **IBM > WebSphere Application Server V8.5 Liberty Profile** (it is already selected). Click **Next**.



- c. On the next screen, enter `blogServer` for the **Server name** and click **Next**.



- __ d. On the next screen, select **blog** from the **Available** list and click **Add** to move it to the **Configured** list. Click **Finish**.



- __ 4. Change the display name for the blogServer.
- __ a. In the Servers view, notice that the server name is **WebSphere Application Server V8.5 Liberty Profile at localhost (2) ... [blogServer]**.



- __ b. Double-click the server name and replace the string for the **Server name** with the simpler name: blogServer

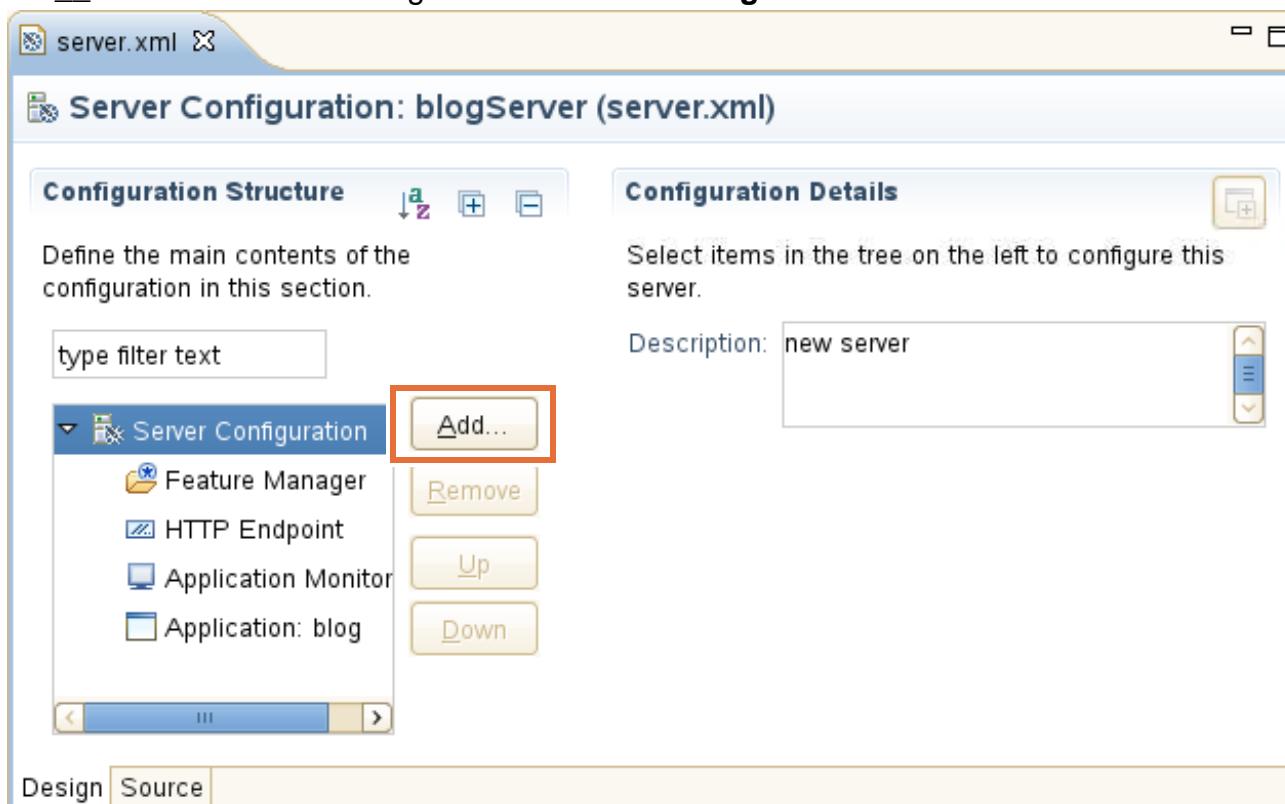


- __ c. Enter **Ctrl-S** to save the configuration. This action also changes the name of the blogServer in the Servers view.
 __ d. Expand the **blogServer** entry and double-click **Server Configuration [server.xml]**.

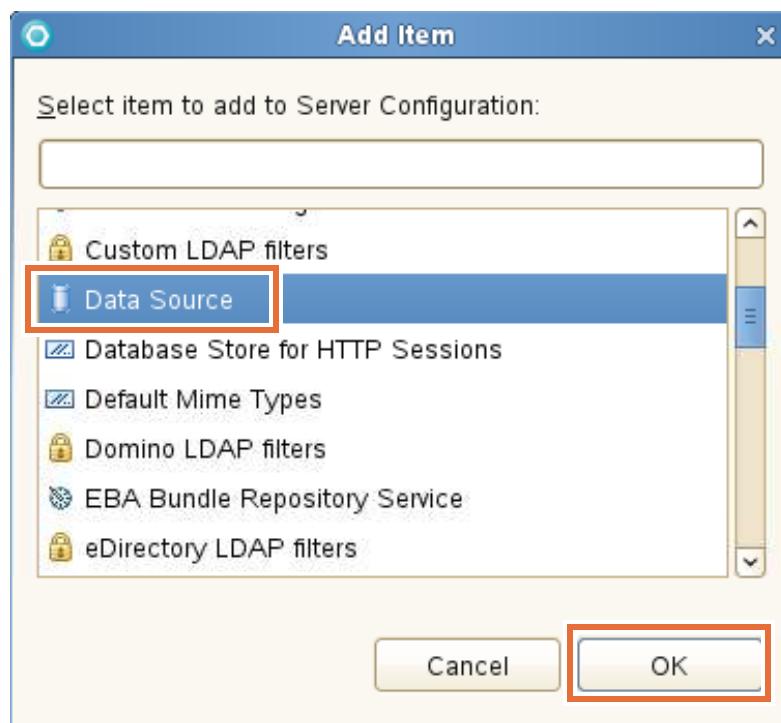


- __ e. The `server.xml` editor opens for blogServer. The editor has two tabs at the bottom, Design and Source. Click the **Design tab** to access Server Configuration.

- ___ 5. Add a data source to the server for the blog application.
___ a. Click **Add** to the right of the **Server Configuration** list.



- ___ b. Start entering data source; from the **Add Item** list, select **Data Source**, and click **OK**.



- ___ c. Under the Data Source Details section, enter blogDS for the **Id** and jdbc/blogDS for the **JNDI name**. Click **New** for the **JDBC driver**.

Data Source Details

| | |
|-----------------------------------|------------------------------------|
| Id: | blogDS |
| JNDI name*: | jdbc/blogDS |
| JDBC driver reference: | <input type="button" value="New"/> |
| Connection manager reference: | <input type="button" value="New"/> |
| Type: | <input type="button" value="New"/> |
| Connection matching: | MatchOriginalRequest |
| Transaction isolation level: | <input type="button" value="New"/> |
| Cached statements per connection: | 10 |

- ___ d. Under the JDBC Driver Details section, click **New** for the **Shared libraries**.

JDBC Driver Details

| | |
|--|------------------------------------|
| Shared library reference: | <input type="button" value="New"/> |
| XADatasource class: | |
| ConnectionPoolDataSource implementation class: | |
| DataSource implementation class: | |

- ___ e. Under the Shared Library Details section, click **New** for the **Fileset reference**.

Shared Library Details

| | |
|--------------------|--|
| Name: | |
| Description: | |
| Fileset reference: | <input type="button" value="Add..."/>
<input type="button" value="New"/>
<input type="button" value="Remove"/> |
| Nested count: 0 | |
| Allowed API types: | spec, ibm-api, api |

- __ f. Under the Fileset Service Details section, for the **Base directory**, select **Absolute Path** from the list next to **Browse**.

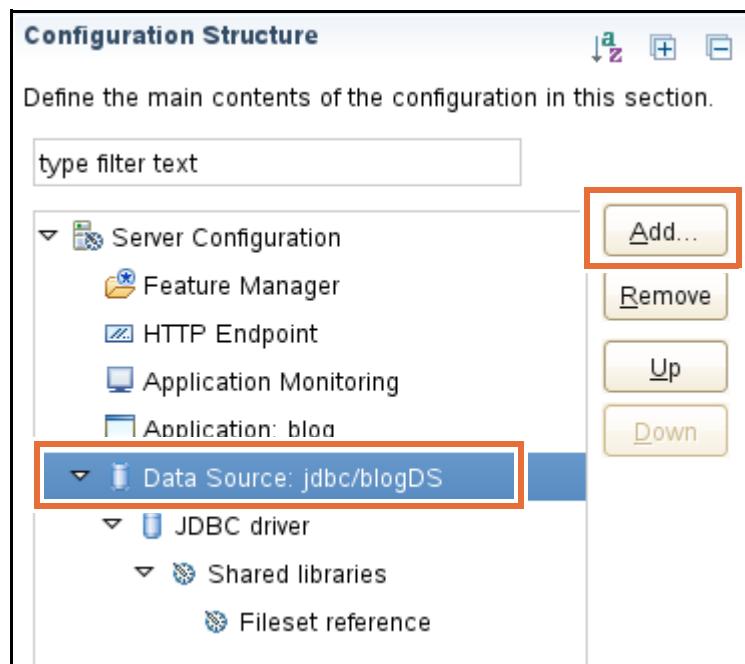


- __ g. Browse to the /usr/software/Liberty folder and click **OK**.
__ h. This action returns you to the **Fileset Service Details**. Click **Browse** next to **Includes pattern**. Select derby.jar and click **OK**.

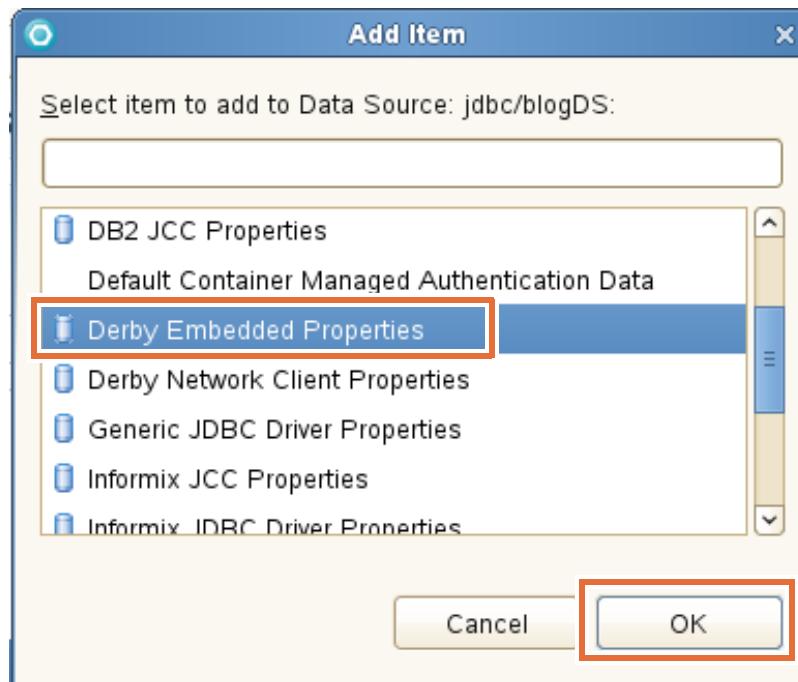


- __ i. Enter **Ctrl-S** to save the configuration.

6. Add the data source properties for Derby.
 - a. Under the Server Configuration, select **Data Source: jdbc/blogDS** and click **Add**.



- b. From the Add Item list, select **Derby Embedded Properties** and click **OK**.



- __ c. For the **Properties for Derby Embedded Details**, enter blogDB for the **database name** and select **create** from the **Create database** list.

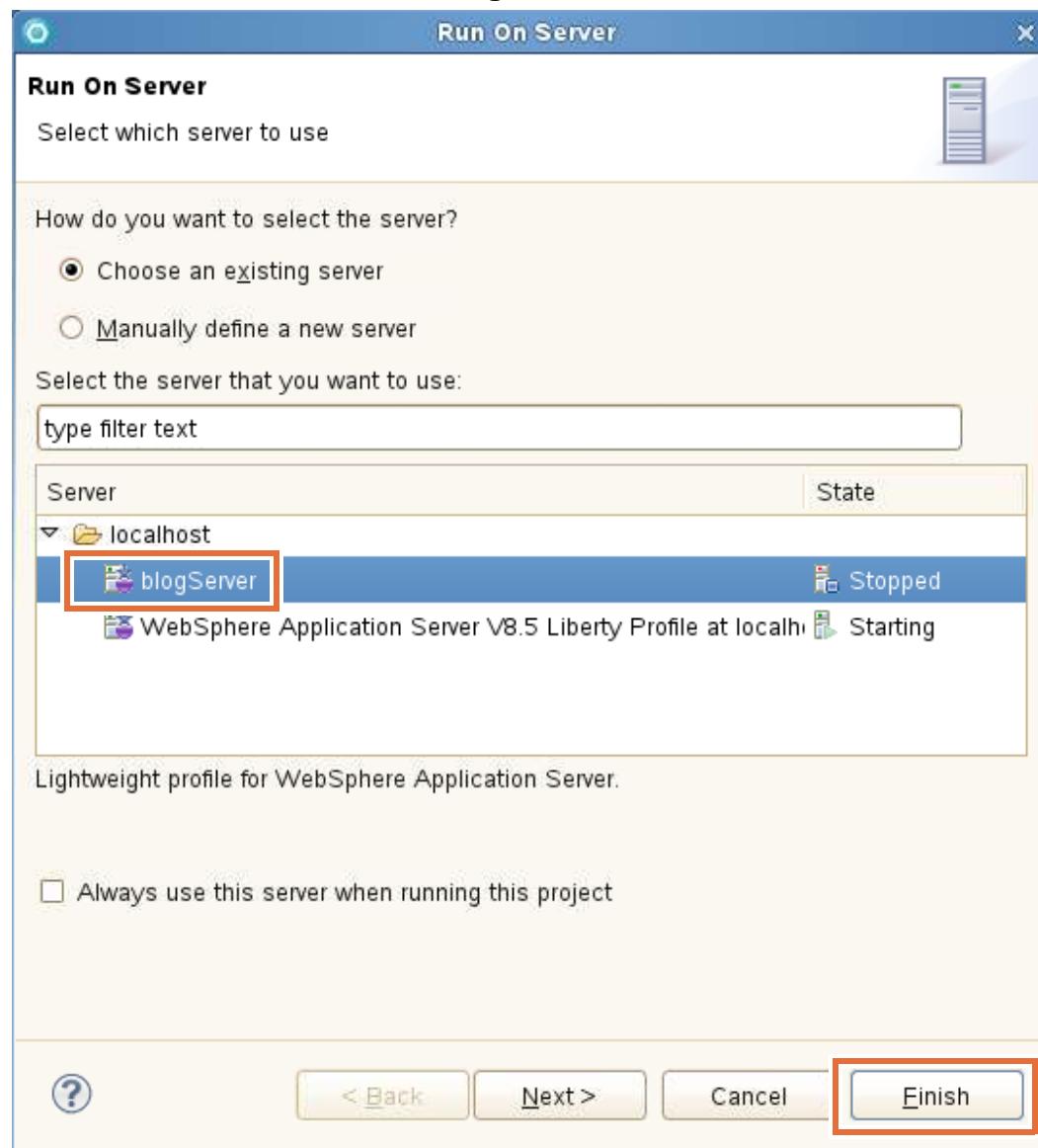


- __ d. Enter **Ctrl-S** to save the configuration changes.
- __ 7. Examine the source XML for the `server.xml` file.
- __ a. Click the **Source** tab for the `server.xml` file. Notice the entry for the data source.

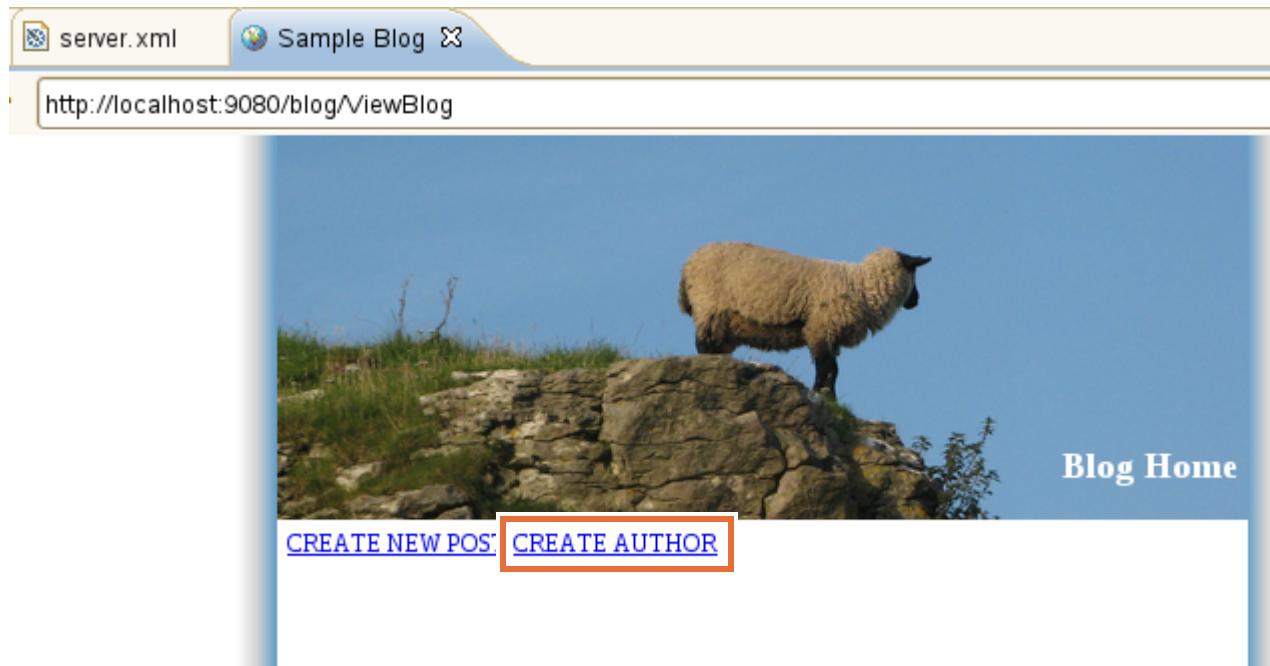
```
<application id="blog" location="blog.war" name="blog" type="war"/>
<dataSource jndiName="jdbc/blogDS" id="blogDS">
    <jdbcDriver id="derbyEmbedded">
        <library id="derbyLib">
            <fileset id="derbyFileset" dir="/usr/software/Liberty" includes="derby.jar"/>
        </library>
    </jdbcDriver>
    <properties.derby.embedded createDatabase="create" databaseName="blogDB"/>
</dataSource>
```

All the actions in the last step created the `dataSource` entry in the `server.xml` file.

8. Run the blog application on the blogServer.
- a. In the Enterprise Explorer view, right-click the **blog** application and select **Run As > Run on server**. Select **blogServer** and click **Finish**.



- __ b. A new tab (internal browser) called Sample Blog opens and loads the blog application. The page might take several seconds to show up in the browser.



- __ c. At your option, you can create an author by clicking the **CREATE AUTHOR** link.

The screenshot shows a "CREATE AUTHOR" form. It includes fields for Name (filled with "Joe Student"), Nickname (filled with "Joe"), Email (filled with "joe@student.com"), Date of Birth (filled with "9/11/1986"), and Bio (filled with "Student for life"). A "Submit" button is at the bottom, highlighted with a red box.

Name	Joe Student
Nickname	Joe
Email	joe@student.com
Date of Birth	9/11/1986
Bio	Student for life

Submit

- __ d. Click **Submit**.

- ___ e. Click **Blog Home** near the upper right corner of the screen to return to the home page. Create a post by clicking the **CREATE NEW POST** link.



- ___ f. Click **Submit**.
- ___ 9. Examine the console message for the application URL.

- ___ a. Select the **Console** tab.

```

Application Server V8.5 Liberty Profile [blogServer] (Jun 21, 2012 9:21:39 AM)
logServer (wlp-1.0.0.20120428-1251/websphere-kernel_1.0.0) on IBM J9 VM, version pxi
CwKE0001I: The server blogServer has been launched.
J2CA8004I: The dataSource blogDS is available as jdbc/blogDS.
J2CA8000I: The jdbcDriver derbyEmbedded is available.
CwKZ0058I: Monitoring dropins for applications.
CwKT0016I: Web application available (default_host) http://localhost:9080/blog/
CwKZ0001I: Application blog started in 1.100 seconds.
CwKF0011I: The server blogServer is ready to run a smarter planet.
  
```

- ___ b. Notice the active hyperlink for the application. This link can be used to open the application directly by clicking it. Click the hyperlink to verify that the application starts in another internal browser.
- ___ c. Close the new internal browser, but leave the original Sample Blog tab.

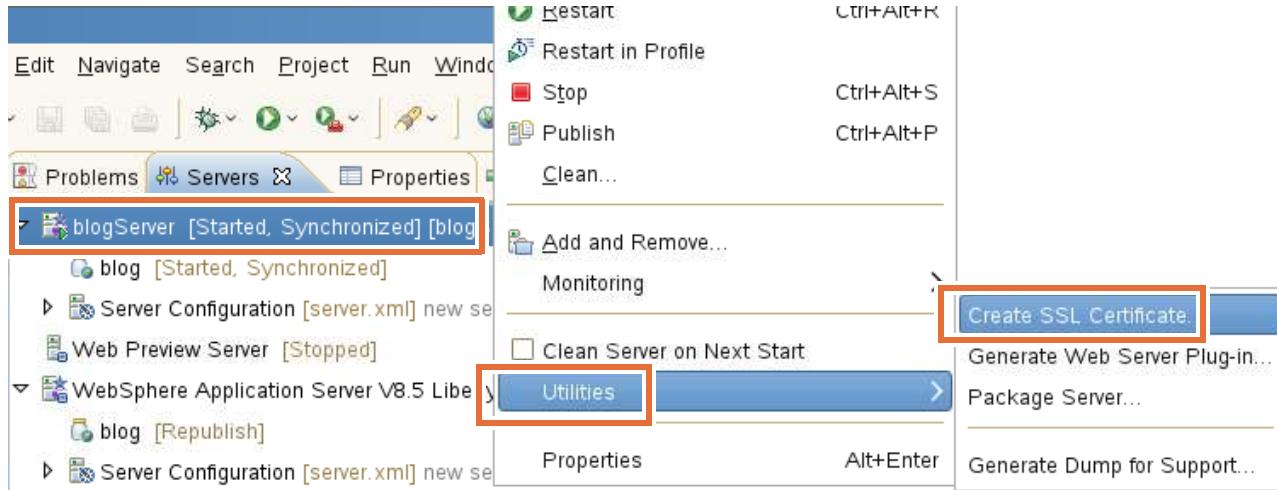
Section 7: Configure SSL and application security

This section looks at the security configurations for the Liberty profile.

- ___ 1. Configure the SSL port for the blogServer.

Notice that the SSL port 9443 is initially configured during server creation. But the port is not active until it is configured.

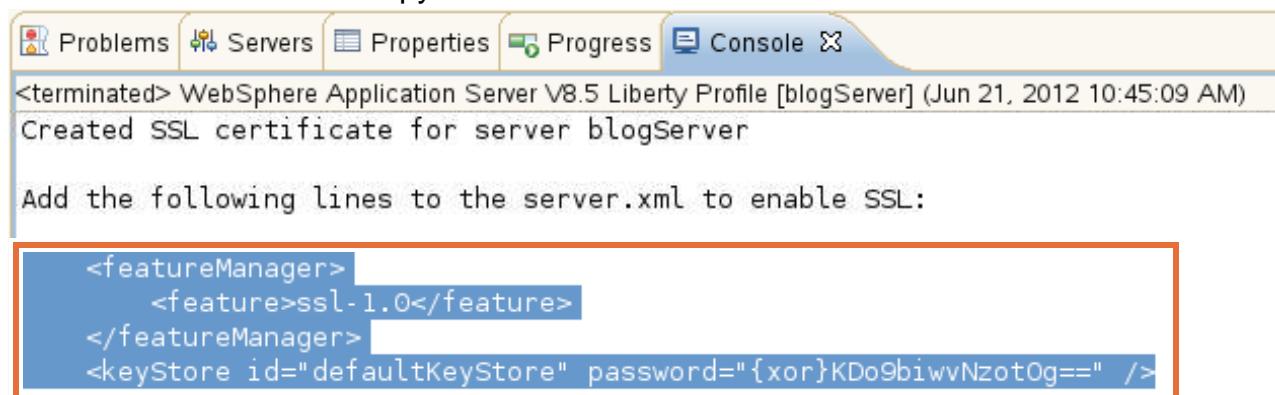
- ___ a. From the Servers tab, right-click **blogServer** and select **Utilities > Create SSL Certificate**.



- ___ b. In the dialog box, enter `websphere` for the **Keystore password**. Select both check boxes, and click **Finish**.



- ___ c. Return to the Console view, and notice the instructions that are provided for adding configuration lines to your `server.xml` file. Highlight the XML lines and enter **Ctrl-C** to copy.

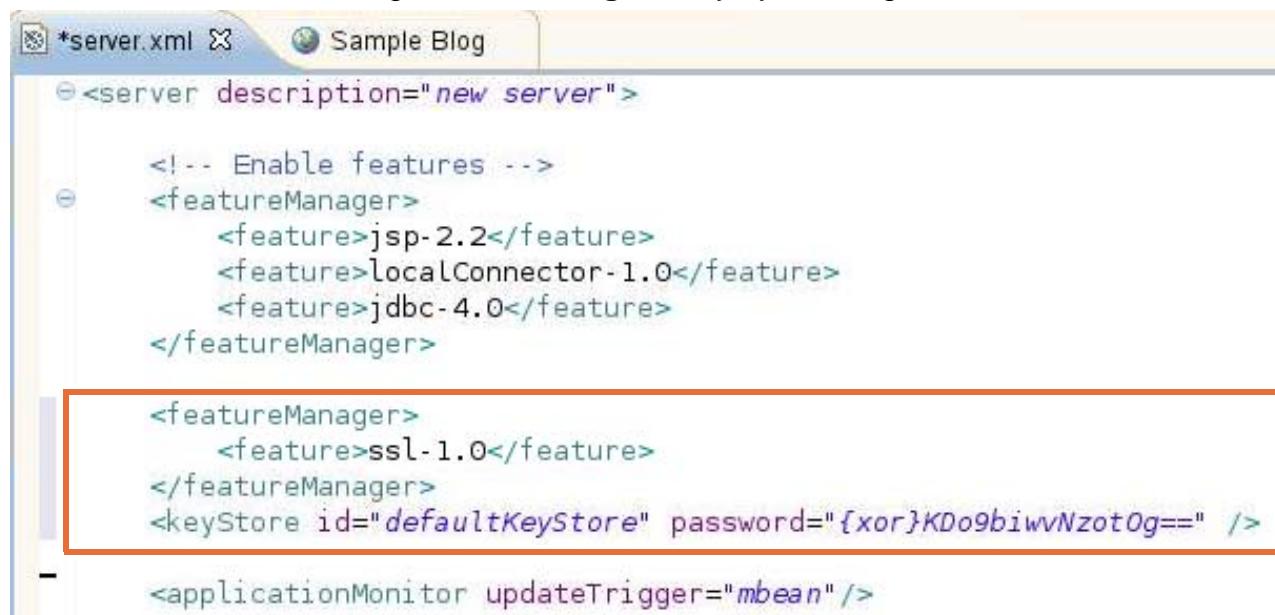


```
<terminated> WebSphere Application Server V8.5 Liberty Profile [blogServer] (Jun 21, 2012 10:45:09 AM)
Created SSL certificate for server blogServer

Add the following lines to the server.xml to enable SSL:

<featureManager>
    <feature>ssl-1.0</feature>
</featureManager>
<keyStore id="defaultKeyStore" password="{xor}KDo9biwvNzotOg==" />
```

- ___ d. Go to the `server.xml` editor, select the **Source** tab, and paste the XML lines below the existing **featureManager** entry by entering **Ctrl-V**.



```
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.2</feature>
        <feature>localConnector-1.0</feature>
        <feature>jdbc-4.0</feature>
    </featureManager>

    <featureManager>
        <feature>ssl-1.0</feature>
    </featureManager>
    <keyStore id="defaultKeyStore" password="{xor}KDo9biwvNzotOg==" />

    <applicationMonitor updateTrigger="mbean" />
```

- ___ e. Enter **Ctrl-S** to save the changes.

- ___ f. Return to the **Sample Blog** tab, and change the URL to the following (changes include protocol **https** and port **9443**):

`https://localhost:9443/blog/ViewBlog`

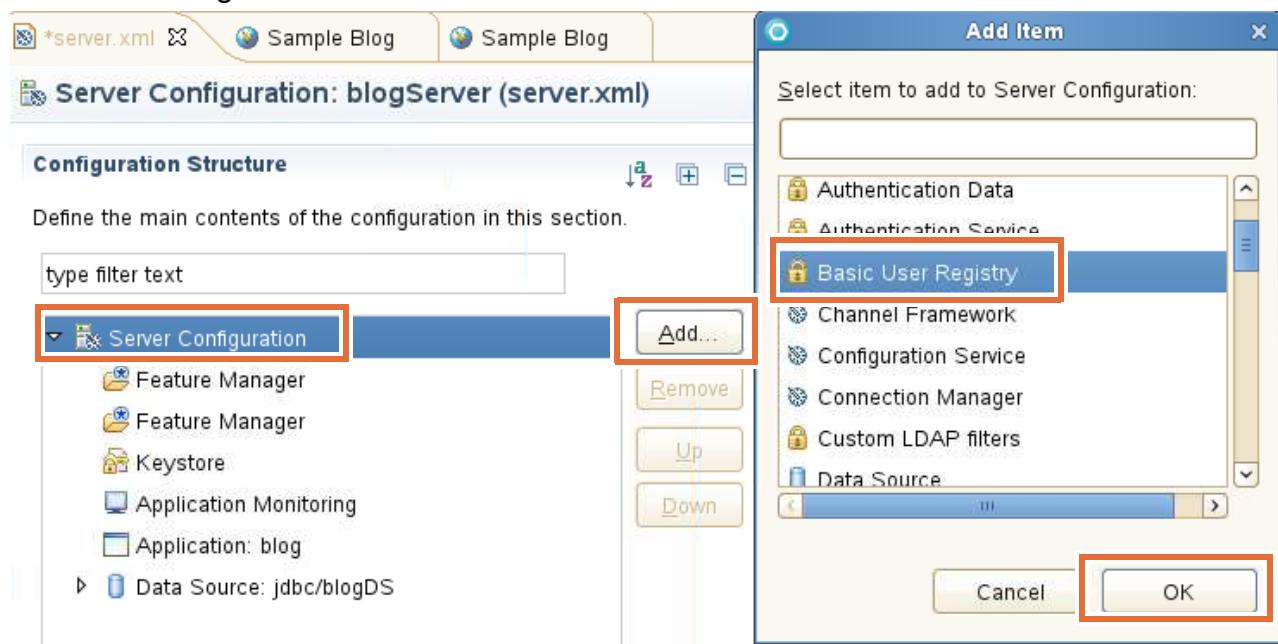


- ___ g. Notice the Invalid Certificate dialog box that is displayed. It is warning you of two issues: a host mismatch and an untrusted certificate authority. Click **OK**. Click **OK** again on the Security Warning.

These warnings are expected. The host mismatch is because the certificate is issued to host was85host, whereas the URL entered is directed to host localhost. The second warning is because the certificate is not signed by a known certificate authority.

2. Configure a user registry.

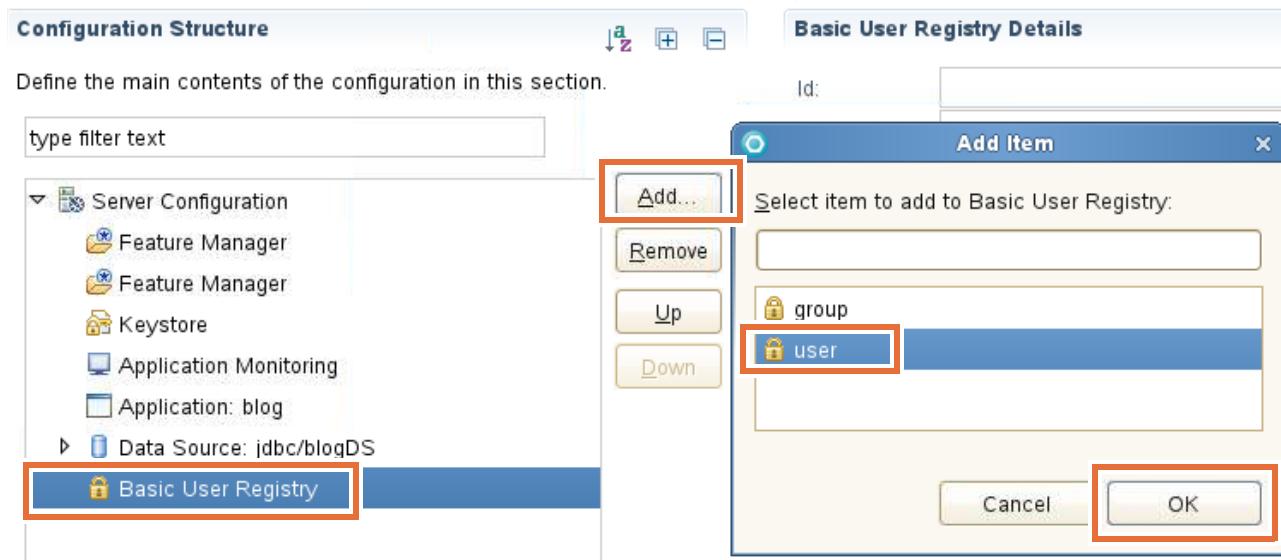
- __ a. In the `server.xml` editor, from the Design view, select the **Server Configuration** and click **Add**. Select **Basic User Registry** from the **Add Item** dialog box and click **OK**.



- __ b. Enter `BasicRegistry` as the **Id**.



- __ c. Add a user to the registry. Select **Basic User Registry** from under **Server Configuration** and click **Add**. From the dialog box, select **User** and click **OK**.



- ___ d. Under User Details, enter a **user name** of: bob
 Click **Edit** to enter the **Password**. In the dialog box, enter `websphere` in both password fields and click **OK**.

User Details

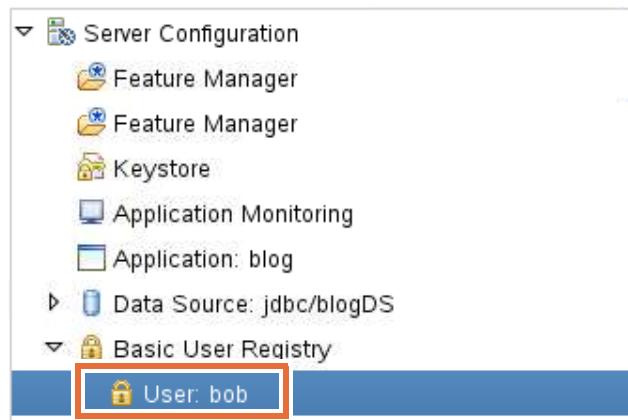
Id:

User name*: bob

Password*: {xor}KDo9biwvNzotOg==

Edit... **Clear**

- ___ e. Notice that user **bob** is listed under **Server Configuration**.



- ___ f. Enter **Ctrl-S** to save the configurations.
 ___ g. Click the **Source** tab and examine the basicRegistry entry in the `server.xml` file.

```
<basicRegistry id="BasicRegistry">
  <user password="{xor}KDo9biwvNzotOg==" name="bob" />
</basicRegistry>
</server>
```

Information

Supported user registries:

The Liberty profile supports basic and LDAP user registries. The basic user registry stores user names and encoded passwords in the `server.xml` file.

An LDAP user registry can be configured that allows the server to store authentication in any of the supported LDAP directory servers, such as IBM Tivoli Directory Server and Microsoft Active Directory.

You can add an LDAP user registry to the server configuration just as you added the basic user registry. However, the LDAP user registry details are more extensive as shown in the following screen capture.

LDAP User Registry Details

Id:	
Host*:	
Port*:	
Base distinguished name (DN)*:	
LDAP server type*:	IBM Lotus Domino
Realm name:	LdapRegistry
Bind distinguished name (DN):	
Bind password:	
<input type="button" value="Edit..."/> <input type="button" value="Clear"/>	
<input checked="" type="checkbox"/> Ignore case for authorization <input type="checkbox"/> Perform a nested group search <input type="checkbox"/> Reuse connection <input type="checkbox"/> LDAP SSL enabled	
SSL configuration:	
Search timeout:	2m

- h. Restart the blogServer.



Note

Since the configuration changes are dynamically loaded, in general it is not necessary to restart the Liberty server. The restarts in the exercise are intended to demonstrate that the server starts without issue when using the new configuration.

- 3. Configure application security. This exercise does not have you complete the required configuration for applying application security for the blog application since much of the configuration is typically the job of the application developer rather than the administrator. The following information block describes the steps that are involved.



Information

Configuring application security:

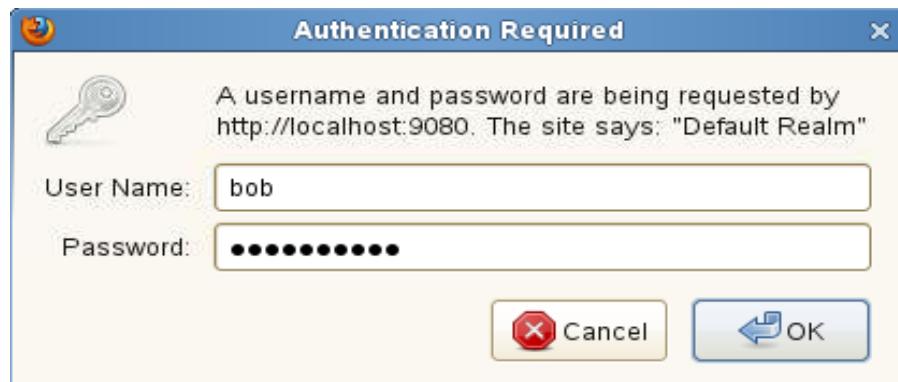
Define security roles and security constraints within the application. For example, the `web.xml` file for the blog application can be edited to include a `blogUser` security role and a `blogConstraint` security constraint. The security constraint would list application resources, the permitted HTTP methods, and transport guarantee. Here is an example of the XML configuration.

```
<security-role>
    <role-name>blogUser</role-name>
</security-role>
<security-constraint>
    <display-name>blogConstraint</display-name>
    <web-resource-collection>
        <web-resource-name>ViewBlog</web-resource-name>
        <url-pattern>/ViewBlog</url-pattern>
        <http-method>PUT</http-method>
        <http-method>POST</http-method>
        <http-method>GET</http-method>
    </web-resource-collection>
    <auth-constraint>
        <role-name>blogUser</role-name>
    </auth-constraint>
    <user-data-constraint>
        <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
</security-constraint>
</web-app>
```

This configuration is an example that shows how one web resource, the `ViewBlog` servlet, can be secured. For a real-world application, it would be necessary to secure other application resources as well.

The application security server feature, `appSecurity-1.0`, must be added to the server configuration. Also, an application binding must be configured that tells the server how to respond to the `blogUser` security role and the members of that role.

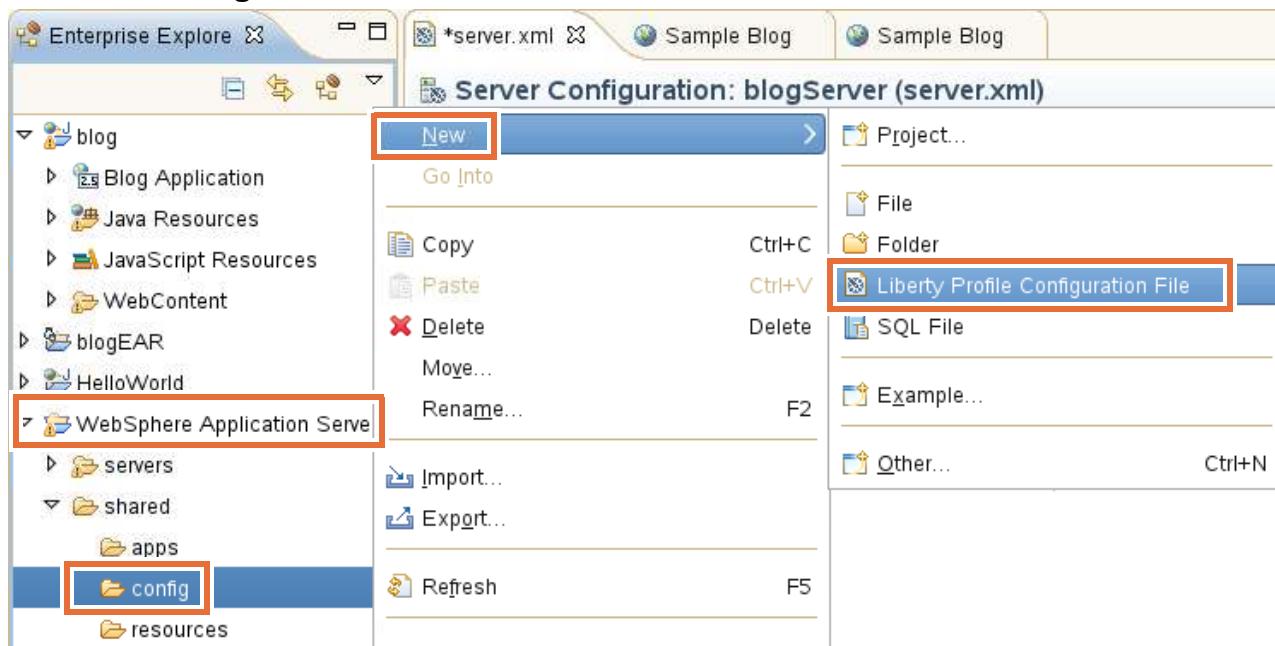
After this configuration is complete, when users try to access the web address `http://localhost:9080/blog/ViewBlog`, they are prompted to authenticate.



Section 8: Explore the use of shared folders and include statements

This section explores the concept of using shared folders and the include statement. These techniques are powerful in the Liberty profile because they allow the administrator to share configurations among numerous servers.

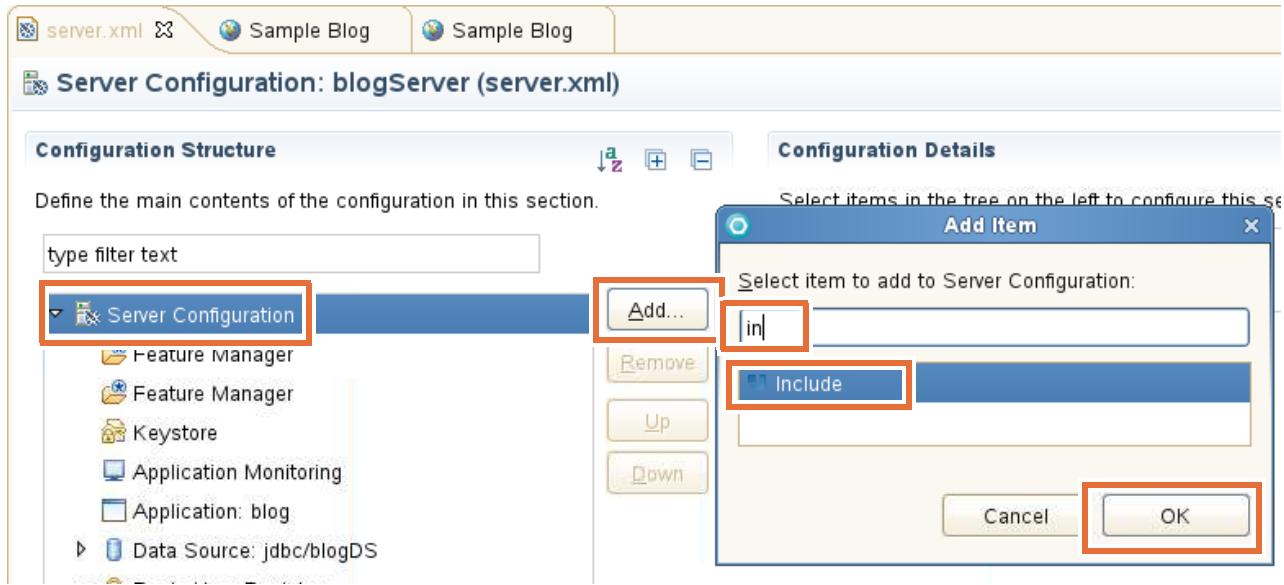
- 1. Create a new shared configuration file.
 - a. In the Enterprise Explorer view, expand **WebSphere Application Server V8 Liberty Profile > shared**. Right-click **config** and select **New > Liberty Profile Configuration File**.



- __ b. In the dialog box, scroll down to **shared > config** and enter `global.xml` for the **File name**. Click **Finish**.



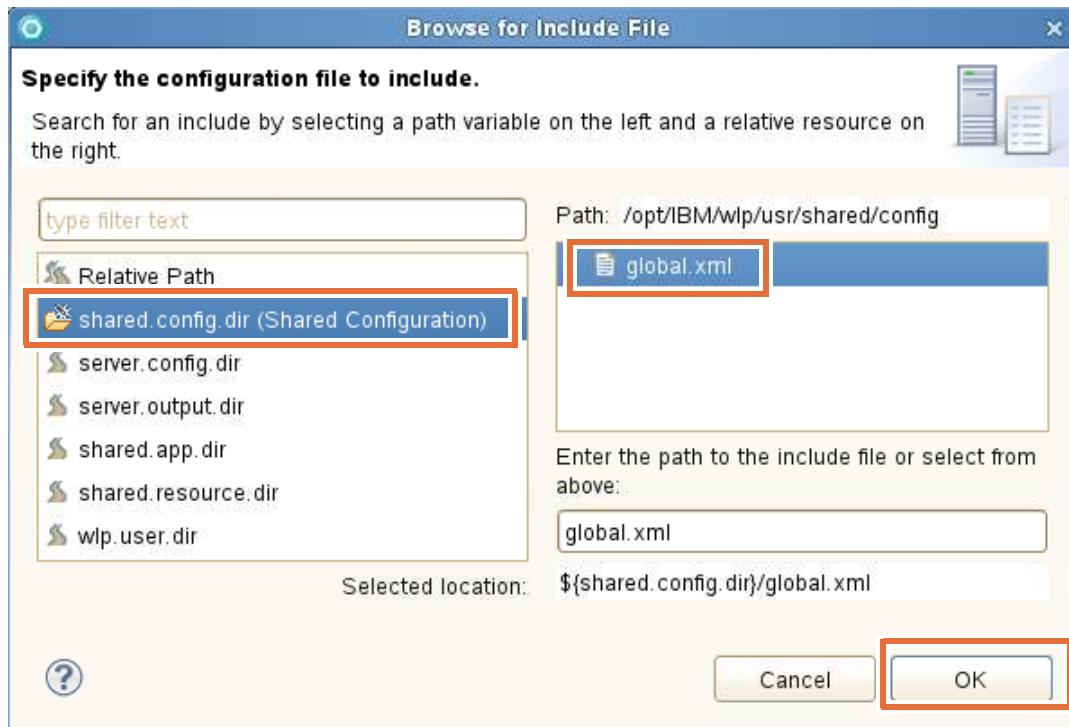
2. Add an include statement to the `server.xml` file and point it at the `global.xml` file.
- a. In the `server.xml` editor, on the **Design** tab select the **Server Configuration** and click **Add**. In the dialog box, start typing the word `include` and notice that this string acts as a filter. Select **Include** and click **OK**.



- b. Click **Browse** in the **Include Details** dialog.



- ___ c. In the dialog box, select **shared.config.dir (Shared Configuration)** on the left, and select **global.xml** on the right. Click **OK**.



- ___ d. Notice that the value of the variable `shared.config.dir` is `/opt/IBM/wlp/usr/shared/config`.
 ___ e. Examine the `server.xml` file. Click the **Source** tab and look for the `include` directive.

```

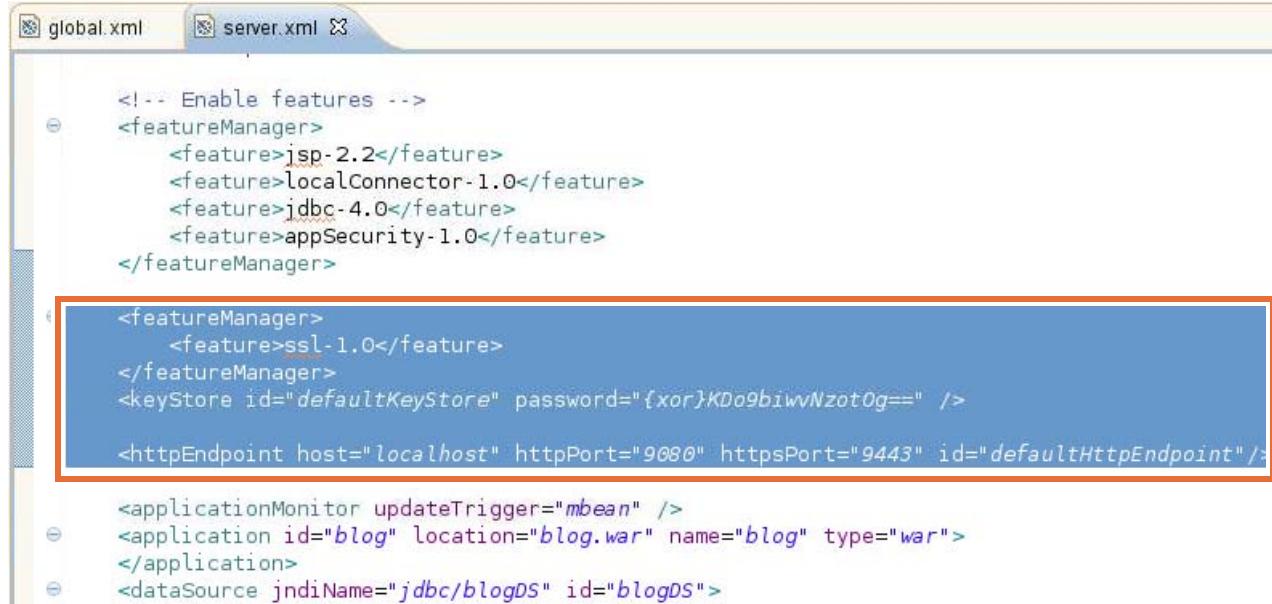
<basicRegistry>
  <user password="{xor}KDo9biwvNzot0g==" name="bob" />
</basicRegistry>

<include location="${shared.config.dir}/global.xml" />

```

Currently, there is nothing in `global.xml`, but it is already included. In the next step, some of the server configuration is moved to `global.xml`.

- ___ 3. Copy the SSL and endpoint configurations to the `global.xml` file.
- ___ a. Open `server.xml` in the Source tab. Select both the SSL feature and endpoint configuration lines (if these lines are not next to each other, do the pasting in two steps). Enter **Ctrl-X** to cut these lines.



```

<!-- Enable features -->
<featureManager>
    <feature>jsp-2.2</feature>
    <feature>localConnector-1.0</feature>
    <feature>jdbc-4.0</feature>
    <feature>appSecurity-1.0</feature>
</featureManager>

<featureManager>
    <feature>ssl-1.0</feature>
</featureManager>
<keyStore id="defaultKeyStore" password="{xor}KDo9biwvNzot0g==" />

<httpEndpoint host="localhost" httpPort="9080" httpsPort="9443" id="defaultHttpEndpoint"/>

<applicationMonitor updateTrigger="mbean" />
<application id="blog" location="blog.war" name="blog" type="war">
</application>
<dataSource jndiName="jdbc/blogDS" id="blogDS">

```

- ___ b. Enter **Ctrl-S** to save the changes.
- ___ c. Switch to the `global.xml` editor and select the **Source** tab. Put the cursor between the `<server>` and `</server>` tags, and press return several times. This action creates some blank space.



- ___ d. Put the cursor in the middle of the blank space and enter **Ctrl-V** to paste the SSL and endpoint configurations.



```

<server></server>

<featureManager>
    <feature>ssl-1.0</feature>
</featureManager>
<keyStore id="defaultKeyStore" password="{xor}KDo9biwvNzot0g==" />

<httpEndpoint host="localhost" httpPort="9080" httpsPort="9443" id="defaultHttpEndpoint"/>

```

- ___ e. Enter **Ctrl-S** to save the configuration changes.
- ___ f. Restart the blogServer and notice that everything works, just as it did before.



Information

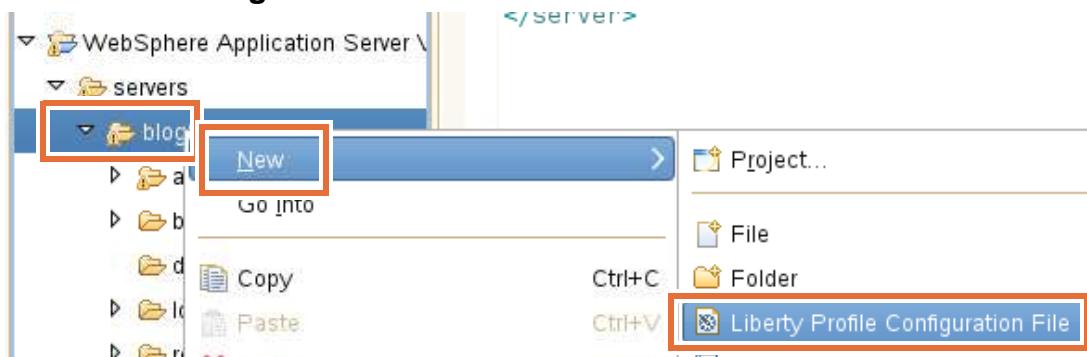
The `global.xml` file can be shared among multiple servers. Consider the case where many Liberty profile servers can share common configuration. Storing that configuration in the `global.xml` file can greatly simplify the management of the servers and the deployment of applications.

- 4. Use variables to factor out the port numbers.

To further help configure the servers, you can use variables in your configurations.

- a. Create a file in the root of the blogServer called: `ports.xml`

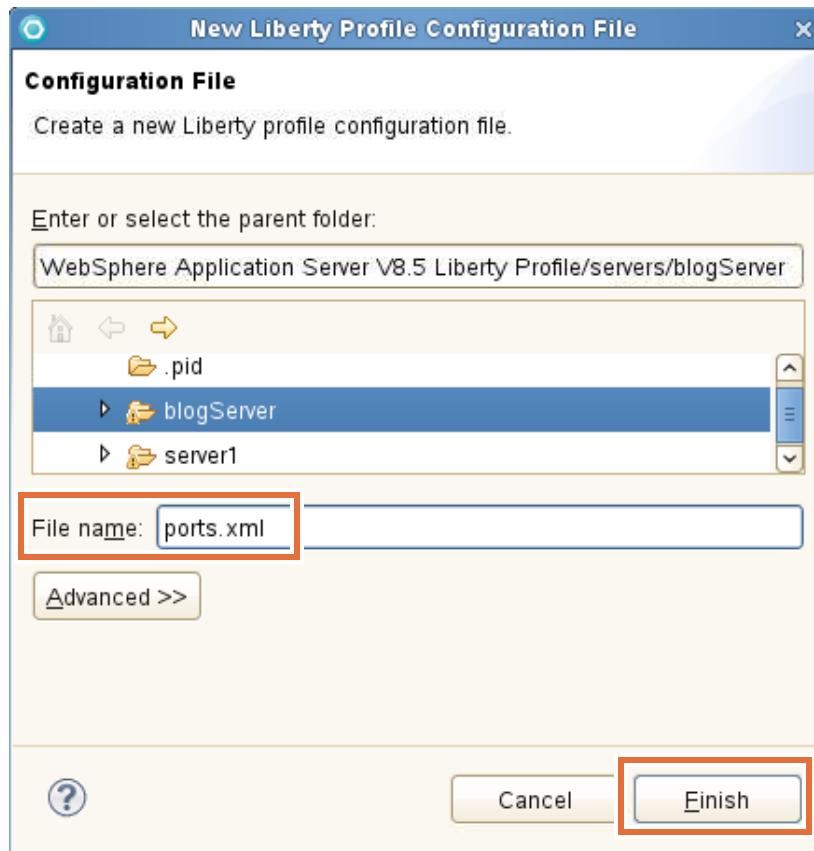
In the Enterprise Explorer, expand **WebSphere Application Server V8.5 Liberty Profile > servers**, right-click **blogServer**, and select **New > Liberty Profile Configuration File**.



Note

Notice that this file is created as part of the blogServer. The previous `global.xml` file was created in the share configuration directory and is therefore available to all servers. This file is going to be visible only to the blogServer.

- __ b. Enter ports.xml for the **File name** and click **Finish**.



- __ c. Open the **ports.xml** tab (which is opened for you automatically when you create the file). Add the following between the `<server>` tags:

```
<variable name="server.http.port" value="9080"/>
<variable name="server.https.port" value="9443"/>
```



- __ d. Enter **Ctrl-S** to save the changes.

- __ e. Select the **global.xml** tab. Add the following line before the `httpEndpoint` tag:

```
<include location="${server.config.dir}/ports.xml" />
```

- __ f. In the `httpEndpoint` line, change the port values as follows:

- From **9080** to `${server.http.port}`
- From **9443** to `${server.https.port}`

```

global.xml ✎ server.xml ✎ ports.xml ✎ Sample Blog
└ <server>
    └ <featureManager>
        <feature>ssl-1.0</feature>
    </featureManager>
    <keyStore id="defaultKeyStore" password="{xor}KDo9biwvNzot0g==" />
    <include location="${server.config.dir}/ports.xml" />
    └ <httpEndpoint
        host="localhost"
        httpPort=" ${server.http.port}"
        httpsPort=" ${server.https.port}"
        id="defaultHttpEndpoint" />
</server>

```

- __ g. Enter **Ctrl-S** to save the changes. Make sure that there are no errors in either the `global.xml` or the `ports.xml` file. If there are errors, the blogServer does not restart successfully.
- __ h. Restart the blogServer and verify that everything works, just as it did before.
- __ i. Edit the values in `ports.xml`. Change the values to: **9081** and **9444**
- __ j. Enter **Ctrl-S** to save the changes. Notice that the server configuration is dynamically updated, and it is not necessary to restart the server.
- __ k. Return to the **Console** view, click the link to access the application on port **9081**, and verify that everything works just as it did before.

```

properties ✎ Console ✎
V8.5 Liberty Profile [blogServer] (Sep 18, 2012 0:26:31 PM)
Processing included configuration resource: file:/opt/IBM/wlp/usr/shared/co
Processing included configuration resource: file:/opt/IBM/wlp/usr/servers/b
The server configuration was successfully updated in 0.3 seconds
Web application available (default_host): http://localhost:9081/blog/*
Web application available (default_host): https://localhost:9444/blog/*

```

- __ l. Notice that if you click the `https://localhost:9444/blog/*` link, you see the same certificate warnings as you did before.



Information

The SSL and endpoint configurations can be shared among all the servers. But adding server-specific `ports.xml` files means that each server can specify its own port numbers, even though the `global.xml` file is shared.

Section 9: Configure the server to generate a web server plug-in configuration file

In this section, you configure the server to generate a plug-in configuration file.

- 1. Configure the server to generate a plug-in configuration file (`plugin-cfg.xml`).
 - a. In the `server.xml` editor, select the **Design** tab. Select **Server Configuration** and click **Add**.
 - b. In the dialog box, start typing the word `generate`. Select **Generate Plugin** and click **OK**.



- __ c. On the right, under the Generate Plugin Details, enter appropriate data for the web server information. The information includes the **HTTP** (80) and **HTTPS** (443) ports, the **SSL keyring** (/opt/IBM/keys/keyring.kdb), **stashfile** (/opt/IBM/keys/stashfile.sth) locations, and the **SSL cert label** (plugin-cert).

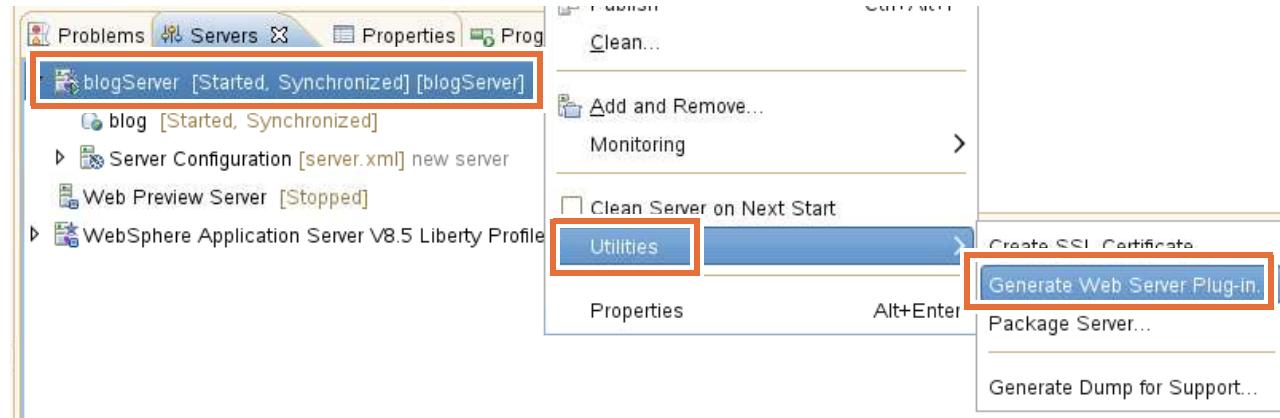
Generate Plugin Details	
Web container plugin installation position in file system:	.
Web server HTTP port:	80
Web server HTTPS port:	443
Location of SSL keyring:	/opt/IBM/keys/keyring.kdb
Location of SSL stashfile:	/opt/IBM/keys/stashfile.sth
SSL cert label:	plugin-cert



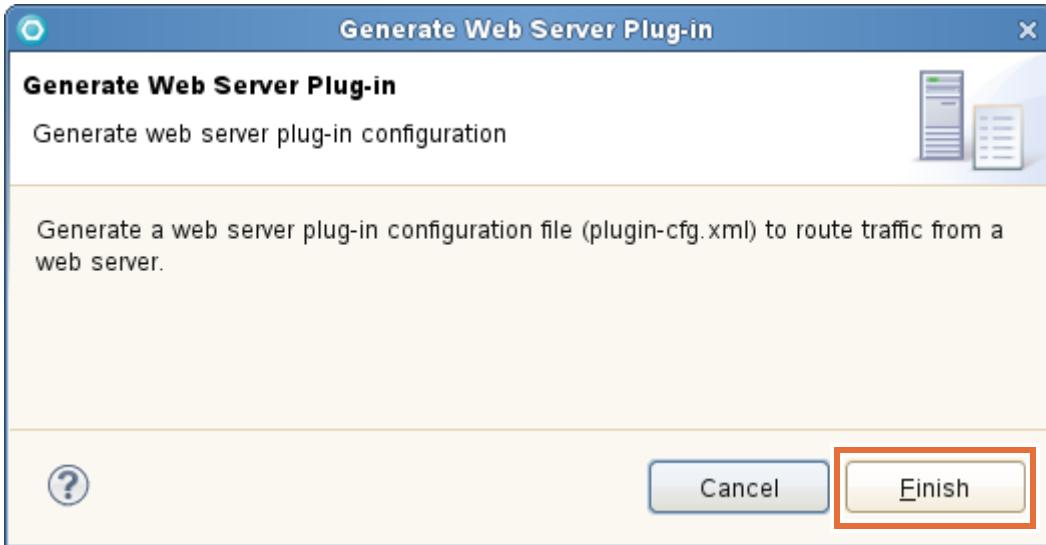
Note

If you do not see the entry fields on the right, expand the size of the window.

- __ d. Enter **Ctrl-S** to save the changes.
- __ e. Generate the plug-in by right-clicking the **blogServer** in the Servers view and selecting **Utilities > Generate Web Server Plug-in**.



- __ f. Click **Finish** on the dialog box that is displayed.



- __ g. In a terminal window, browse to /opt/IBM/wlp/usr/servers/blogServer. Use the ls command to get a directory listing.

```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/wlp/usr/servers/blogServer # ls
apps derbyDB dropins plugin-ctg.xml resources tranlog
blogDB derby.log logs ports.xml server.xml workarea
was85host:/opt/IBM/wlp/usr/servers/blogServer #
```

- __ h. Use gedit with the following command to look at the contents of the plug-in file:

```
gedit plugin-cfg.xml &
```

- __ i. Explore the contents of the configuration file. Close the gedit window when you are done.

End of exercise

Exercise review and wrap-up

This exercise goes through installing the Liberty profile runtime environment. After an application server is created, both the command line and the developer tool are used to start and stop the server. Applications are also deployed both through the developer tool and through the `dropins` directory. You also explored configuring various features in the Liberty profile application server. These features included data source, security, shared configurations, and shared libraries.

Exercise 16.Using the performance monitoring tools

What this exercise is about

In this exercise, you use the performance tools available in WebSphere Application Server to monitor various application and server resources in real time, and generate tuning advice for performance.

What you should be able to do

At the end of this exercise, you should be able to:

- Enable various levels of Performance Monitoring Infrastructure (PMI) statistics for an application server
- Monitor an application server by using Tivoli Performance Viewer
- Configure user settings for Tivoli Performance Viewer
- Examine summary reports and performance modules in Tivoli Performance Viewer
- View performance messages from the Tivoli Performance Viewer Advisor
- Enable and configure the Request Metrics tool
- View Request Metrics messages in the standard logs of an application server
- Configure IBM Tivoli Composite Application Manager for WebSphere Application Server collector for an application server
- View IBM Tivoli Composite Application Manager application performance statistics by using Tivoli Performance Viewer

Introduction

WebSphere Application Server offers a collection of tools to monitor and help tune the runtime environment, including Tivoli Performance Viewer, the Tivoli Performance Viewer advisor, and Request Metrics.

The Tivoli Performance Viewer is the user interface for monitoring the performance of application servers, servlets, and other resources in the environment. It is integrated into the administrative console and can be used for a number of tasks, including viewing real-time

performance data, gauging the load on servers over time, and evaluating the efficiency of resource allocations.

The Tivoli Performance Viewer advisor provides advice to help tune systems for optimal performance and recommendations to remedy inefficient server resource settings. It generates advice that is based on data that the Performance Monitoring Infrastructure (PMI) collects.

Request metrics allow you to track the response time of the individual components that a transaction traverses, providing you with an in-depth understanding of the application flow that satisfies the user request.

Requirements

This exercise requires at least one application server that runs the PlantsByWebSphere application and the Default application.

Exercise instructions

Preface

In the first part of this exercise, you use Tivoli Performance Viewer to monitor applications that run on WebSphere Application Server.

When the Performance Viewer is running in a Network Deployment environment, the data is collected at each of the nodes and stored in memory at the node agent. Data is then viewed from the deployment manager. With this architecture, the monitoring is distributed among the nodes.

Section 1: Resetting the WebSphere environment



Note

To reset your WebSphere environment, read **Appendix A** for instructions on how to complete this procedure.

Section 2: Verifying the environment

If you changed the maximum heap size for the deployment manager to a lower value to conserve system resources, you must restore the maximum heap size to the default setting. More memory is required because the Tivoli Performance Viewer runs inside the deployment manager.

- 1. Verify that the deployment manager, the node agent, and WebSphere Application Server server1 in profile1 are all running.
- 2. Start an administrative console and make sure that you are logged in as: wasadmin



Information

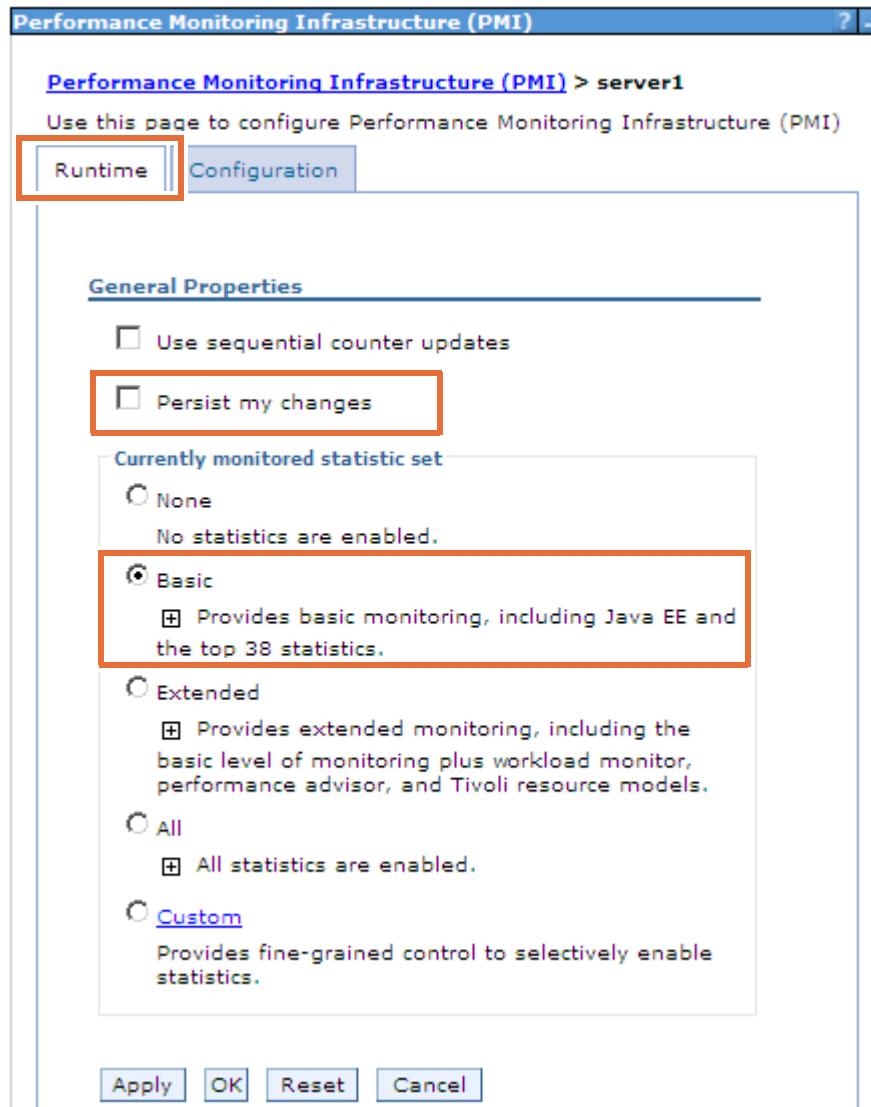
You can stop server2 and its node agent to free up physical memory if the class environment has limited memory. As a rule of thumb, if your computer has less than 2 GB of memory, it can be beneficial to run only server1 for this lab.

Section 3: Enabling performance monitoring and setting user preferences

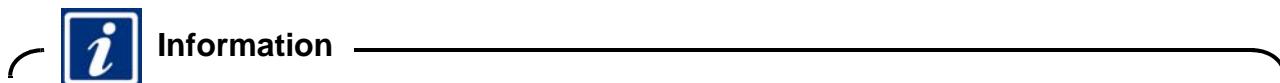
Before the Tivoli Performance Viewer can begin monitoring data, the performance monitoring service must be started. The monitoring service is turned on by default on the application server.

- 1. Verify that the Performance Monitoring Infrastructure (PMI) is turned on for server1.

- __ a. In the administrative console navigation tree, click **Monitoring and Tuning > Performance Monitoring Infrastructure (PMI)**. In the workspace area on the right pane of the administrative console, click **server1**.
- __ b. On the Configuration tab, make sure that **Enable Performance Monitoring Infrastructure (PMI)** is already selected.
- __ c. In the Currently monitored statistic set area, notice that the **Basic** level is selected. Basic is the default monitoring level setting.
- __ d. Click the **Runtime** tab and verify that **Basic** is selected (default statistic set).



- __ e. Click **Cancel**.



The **Runtime** tab allows you to change the monitoring settings without restarting the server. The new settings are applied immediately after clicking **OK**. If the **Persist my changes** checkbox is checked, the changes are saved to disk.

changes check box is selected, the runtime settings are saved and are shown in the **Configuration** tab. This option allows you to change the PMI settings, and persist these changes for the next time the server is started.

- 2. To get more frequent data collections in the Tivoli Performance Viewer, change the **Monitoring Refresh Rate** to 20 seconds.
 - a. In the administrative console, under Monitoring and Tuning, expand **Performance Viewer** and click **Current activity**.



- b. On the right pane, select the check box for **server1** and start the monitoring process for this server by clicking **Start Monitoring**.

Tivoli Performance Viewer

Tivoli Performance Viewer

Specifies the server to monitor with Tivoli Performance Viewer. Select the check box for the servers that you want Start Monitoring. Click the name of the server to display the activity page.

Preferences

	<input type="button" value="Start Monitoring"/>	<input type="button" value="Stop Monitoring"/>			
<input type="checkbox"/>					
Select	Server	Node	Host Name	Version	Collection Status
<input type="checkbox"/>	nodeagent	was85hostNode01	was85host	ND 8.5.5.0	Available
<input type="checkbox"/>	nodeagent	was85hostNode02	was85host	ND 8.5.5.0	Unavailable, server st
<input checked="" type="checkbox"/>	server1	was85hostNode01	was85host	ND 8.5.5.0	Available
<input type="checkbox"/>	server2	was85hostNode02	was85host	ND 8.5.5.0	Unavailable, nodeage
Total 4					

- c. Click the **server1** link to view its current activity. If the tree for server1 is collapsed, click **[+]** next to server1 to expand it. Expand **Settings** and click **User**.

- __ d. In the User Settings panel, change the Data Collection **Refresh Rate** to **20** seconds. Click **Apply**.

The screenshot shows the Tivoli Performance Viewer interface. At the top, it says "Tivoli Performance Viewer > server1". Below that, there's a brief description: "Use this page to view and refresh performance data for the selected server, change user and log settings, and view summary reports and information on specific performance modules." On the left, there's a navigation tree for "server1" with nodes like Advisor, Settings (which has User and Log), Summary Reports, and Performance Modules. There are "Refresh" and "View Module(s)" buttons at the top left. On the right, there's a "User Settings" section with a "Data Collection" sub-section. The "Refresh Rate" field is set to "20 seconds" and is highlighted with a red box. Below it are "Buffer Size" (set to 40) and "Data points". Under "View Data As", "Raw" is selected. At the bottom right are "Apply" and "Cancel" buttons, with "Apply" also highlighted with a red box.

Section 4: Viewing servlet and web applications module data

In this section, you use the Tivoli Performance Viewer to generate and view performance metrics.

- __ 1. Open a new browser and start the **Snoop** servlet by entering the web address:

`http://was85host:9080/snoop`

If application security is on, you are prompted for a user ID and password. Log in as `wasadmin` if necessary.

This URL runs the Snoop servlet, which is part of the Default application, and shows a page with various information about the servlet. A servlet must be loaded in order for data collection to take place. Leave the browser window open, as you are going to return here soon.

- __ 2. Using the administrative console, go to the Tivoli Performance Viewer and monitor server1.

- __ a. Click **Monitoring and Tuning > Performance Viewer > Current activity > server1**.
- __ b. Expand **Summary Reports** and select **Servlets**. In the Servlets Summary Report pane, locate the entry for the **Snoop servlet**, which is part of the DefaultWebApplication.war file.



Information

You might need to go to the next page of the report to find the listing for the Snoop servlet. Use the arrows at the bottom of the page to go to the different pages of the report. As an alternative, you can use filters to reduce the amount of information that is shown, or sort by application name or another attribute.

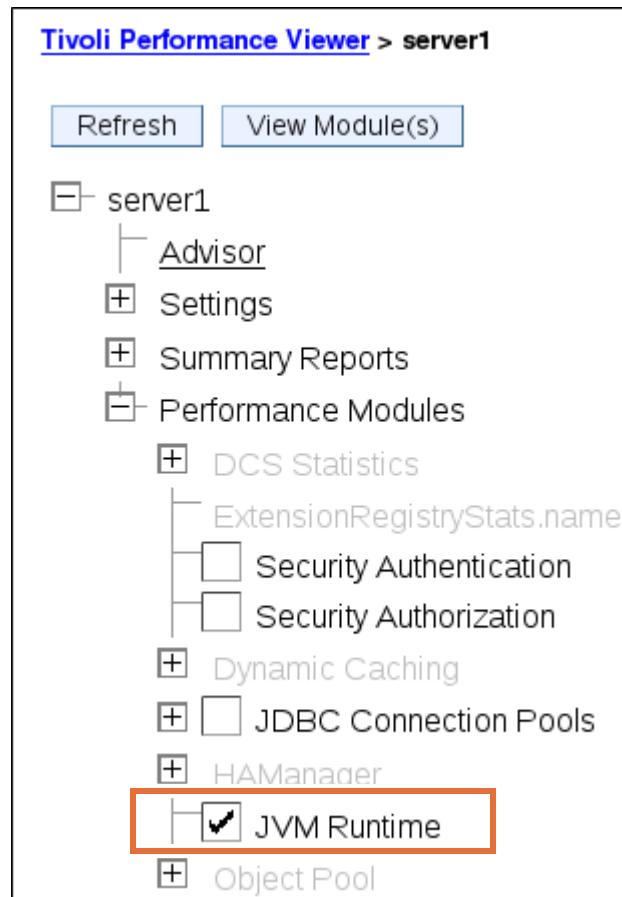
Servlets Summary Report						
More information about this page						
<input type="button" value="Start Logging"/> 						
Name	Application	Total Requests	Avg Resp Time (ms)	Total Time (ms)	Time	
Hello Pervasive Servlet	DefaultWebApplication.war	0	0	0	4:52:11 PM	
Hit Count Servlet	DefaultWebApplication.war	0	0	0	4:52:11 PM	
Snoop Servlet	DefaultWebApplication.war	1	24	24	4:52:11 PM	
rsp servlet	ibmasyncrsp.war	0	0	0	4:52:11 PM	
Total 4						

- __ c. Notice that there is one request for the Snoop servlet and an average response time in milliseconds. Record the average response time here: _____
 - __ d. Go back to the browser and reload the page several times.
 - __ e. Under the **Summary Reports**, click **Servlets** again to refresh the view. Look at the Servlets report again. What is the average response time now? _____
Is the response time longer or shorter after several requests are processed?
(The response time is now shorter because of caching.)
- __ 3. Open a new browser and enter the web address for PlantsByWebSphere:

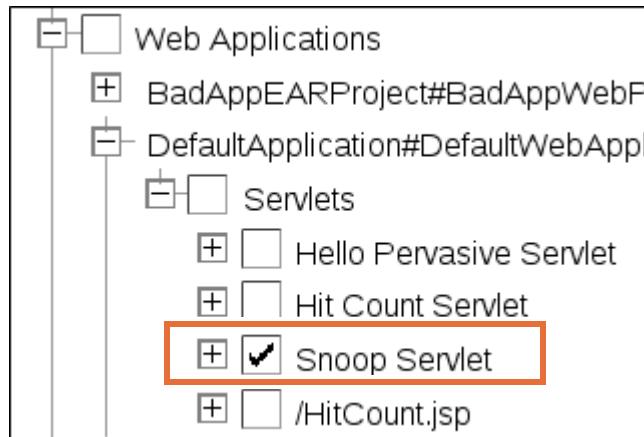
`http://was85host:9080/PlantsByWebSphere`

Click the **Flowers** tab and the link for **Lily**.

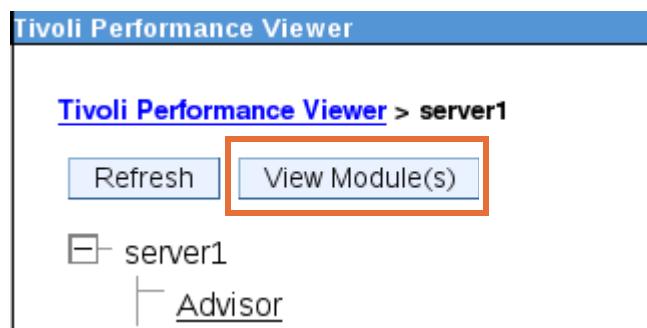
- a. Explore more summary reports:
- Are there any Enterprise JavaBeans being monitored? Click **EJBs** under **Summary Reports**. The EJBs Summary Report lists all enterprise beans that are currently running on this server. It also shows the amount of time that is spent in their methods, the number of EJB invocations, and the total time that is spent in each enterprise bean.
 - Click **Connection Pool**. The Connection Pool Summary Report lists all data source connections that are defined in the application server and shows their usage over time. The performance data is shown in graph form.
 - Click **Thread Pool**. The Thread Pool Summary Report shows the usage of all thread pools in the application server over time.
- 4. Inside Tivoli Performance Viewer, view the available performance counters for the Snoop servlet and the JVM Runtime module.
- a. In Tivoli Performance Viewer, expand **Performance Modules** and select the **JVM Runtime** check box.



- __ b. Expand **Web Applications > DefaultApplication#DefaultWebApplication.war > Servlets** and select **Snoop Servlet**.

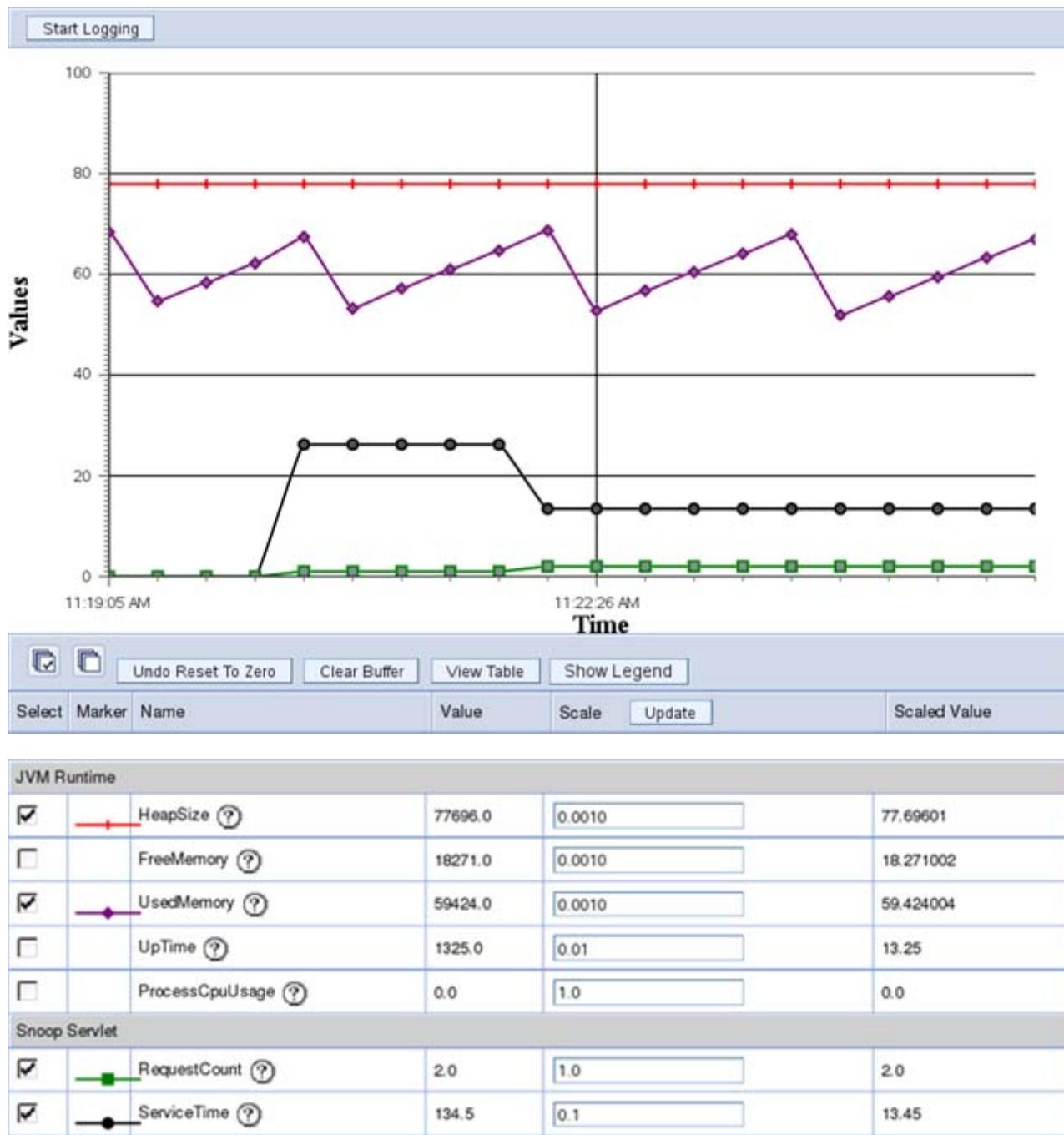


- __ c. Click **View Module** at the top of the viewer (if necessary, scroll in your browser to see this screen). You see a table or a graph that shows the monitored data.



- __ d. Open a new browser and start the **Snoop** servlet by entering the web address:
<http://was85host:9080/snoop>

- e. Use the browser to **reload the Snoop servlet** several times by clicking the browser refresh button. Review the changes in the console.



Information —————

You see changes in the metrics for the Snoop servlet. The request count increases, and the service time changes. Keep in mind that the JVM runtime counters change too. Notice in

the screen capture, the FreeMemory metric was cleared and it therefore is not plotted on the graph.



Warning

When viewing graphs and comparing lines, take note of the Scale value. The Performance viewer scales values such that all data points can fit on the graph.

- __ f. In the Performance viewer, click **View Table** to switch to a tabular view of the performance data. You can toggle back and forth between the table and graph views by selecting **View Table** or **View Graph**.

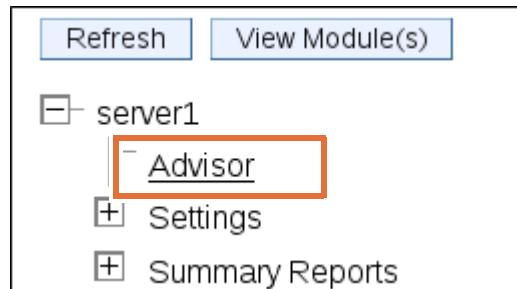


- __ g. Reload the Snoop servlet several times and observe the servlet metrics.

Section 5: Using the Tivoli Performance Viewer performance advisor

WebSphere Application Server includes a performance advisor, the Tivoli Performance Viewer advisor, which is accessed from inside Tivoli Performance Viewer. The Tivoli Performance Viewer advisor provides helpful tuning advice for various resources, cache size, JVM heap size, and more. The Tivoli Performance Viewer advisor also provides recommendations to address inefficient settings.

- 1. To access Tivoli Performance Viewer advisor messages in the administrative console, select **Monitoring and Tuning > Performance Viewer > Current Activity > server1**. In the Tivoli Performance Viewer navigation pane, click **Advisor** under **server1**.



- a. In the pane on the right, read the provided advice messages.
Are there some Alert messages? _____
Is there a configuration advice message? _____
If more than one page is available, view the messages on subsequent pages.

<input type="button" value="Refresh All Advice"/> <input type="button" value="Remove Selected Advice"/>			
Select	Severity	Message	Status
<input type="checkbox"/>	Config	TUNE5012W: The size of the minimum ...	Unread
<input type="checkbox"/>	Config	TUNE5042W: Enable servlet caching f...	Unread
<input type="checkbox"/>	Warning	TUNE0303W: Number of threads workin...	Unread
<input type="checkbox"/>	Warning	TUNE0303W: Number of threads workin...	Unread
<input type="checkbox"/>	Warning	TUNE0303W: Number of threads workin...	Unread

Page: 1 of 3 Total 12

- ___ b. Click the link for the **TUNE5042W: Enable servlet caching for better performance** message to see the advice details.

General Properties	
Message	TUNE5042W: Enable servlet caching for better performance.
Severity	Config
Description	Servlet caching is not enabled.
User Action	To enable servlet caching in the administrative console, click Servers > Application servers > server_name > Web container settings > Web container and select Enable servlet caching under the Configuration tab. Click Apply or OK. You must restart your Application Server.
Detail	Currently, servlet caching is disabled.
Back	



Information

In your lab environment, typically the processor utilization is low, and so you do not see much relevant performance advice. However, you see the configuration advice to turn on servlet caching with descriptions as to how to use the console to turn on this feature.

It is a good practice to run a representative workload against your application. Look at the performance advisor after the workload is run.

You can use the IBM Rational Performance Tester to create load tests and run them against your applications. You can read more about Rational Performance Tester and download a trial version at the following website:

<http://www.ibm.com/software/awdtools/tester/performance>

Section 6: Using request metrics

Request metrics log the time that is spent at major components of the application server, such as the web server plug-in, web container, EJB container, and more.

The request metric architecture differs from the Performance Monitoring Infrastructure (PMI). PMI provides information about average system resource usage, with no correlation between the data across different WebSphere components.

The request metrics tool tracks each individual transaction within WebSphere Application Server, recording the response time of the major components. Some of these response times include time in the web server or in the Enterprise JavaBeans (EJB) container. The collected information can be saved to log files or forwarded to an Application Response Measurement (ARM) agent.

- 1. Using the administrative console, turn on request metrics by selecting **Monitoring and Tuning > Request metrics**.
 - a. Under **General Properties**, select **Prepare Servers for request metrics collection**.
 - b. Under **Components to be instrumented**, select **ALL**.
 - c. Set **Trace level to Debug**.



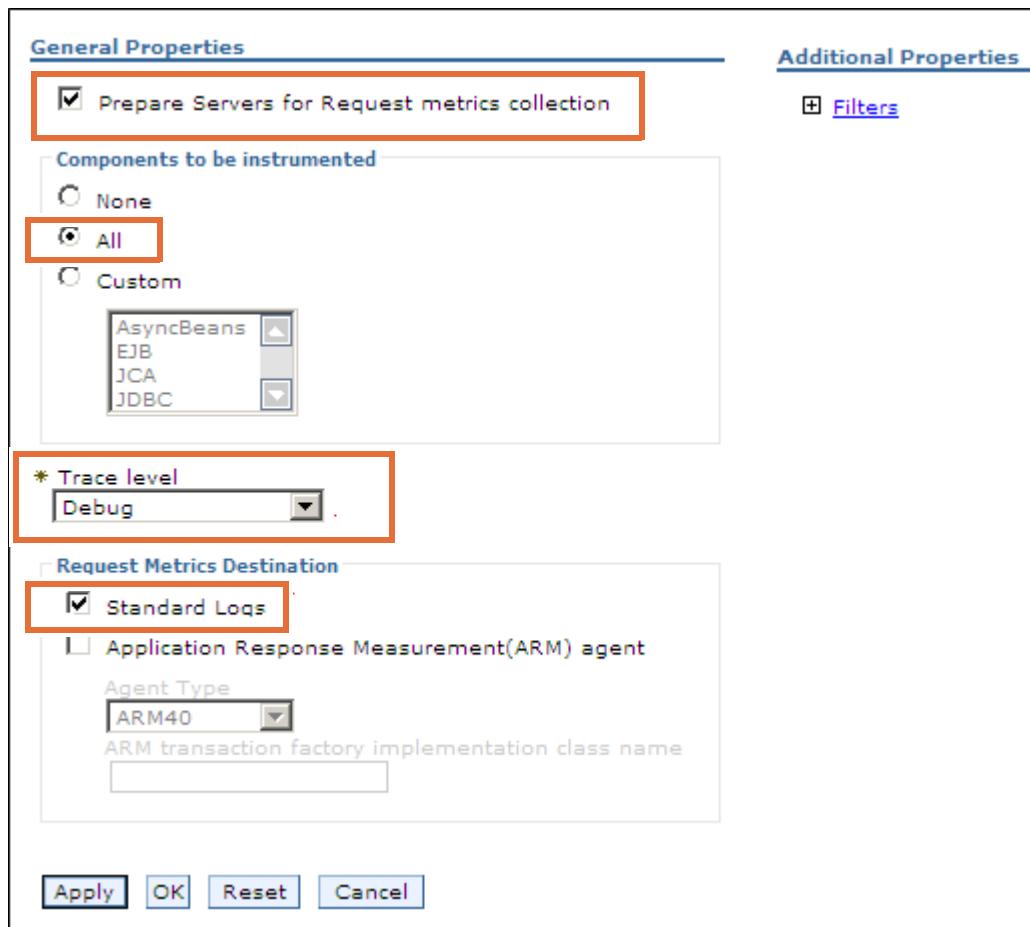
Information

Setting the trace level to **Debug** provides detailed instrumentation data, including response times for all intra-process servlet and Enterprise JavaBeans (EJB) calls. This trace level provides a fine level of detail on each method call.

Setting the trace level to **Hops** generates instrumentation information about process boundaries only (for example, a servlet request that comes from a browser or a web server, and a JDBC request that goes to a database).

You can also filter requests so that only specific incoming requests result in request metrics that are logged. Such filtering can keep the logs from being overloaded with request metrics for every request.

- __ d. Under Request Metrics Destination, select **Standard Logs**.



- __ e. Click **OK**.
- __ f. **Save** directly to the master configuration and log out of the administrative console.
- __ g. Restart **server1**.
- __ 2. Open a web browser and run the PlantsByWebSphere application by entering the following address:
`http://localhost:9080/PlantsByWebSphere`
- __ a. Go through the site, look at some plants, buy something, and check out.
- __ 3. Now view the standard JVM logs for server1. If HPEL is not turned on for server1, you can go to the Problem determination lab exercise for the steps on how to turn on HPEL. Alternatively, you can open the `SystemOut.log` file with a text editor and search for the PMRM codes. The remaining steps assume that HPEL is turned on for server1.
- __ a. In the administrative console, go to **Troubleshooting > Logs and trace > server1**.

- __ b. On the Logging and tracing panel, scroll down and click the link **View HPEL logs and trace**.

Related Items

- [View HPEL logs and trace](#)
- [Change log detail levels](#)
- [Change log and trace mode](#)
- [Manage process logs](#)
- [NCSA access and HTTP error logging](#)

- __ c. You are now in the Log Viewer. Expand the **Content and Filtering Details** section.

Logging and tracing

Logging and tracing > server1 > Log Viewer

Use this page to view log data from the HPEL repository the customized view or full repository into a compressed

Content and Filtering Details

- __ d. Scroll down to the **Filtering** box and type **PMRM*** in the **Message contents** field.

View Contents

System out
 System err
 Logs and trace

Minimum level:

Maximum level:

Filtering

Wild cards: *?% are allowed
Separate multiple entries by a ;
Include loggers:
Exclude loggers:
Message contents:

Event Timing

From: On:
Until: On:

- __ e. Click **Apply**.

- ___ f. Now only the request metrics messages are shown in the log view. Click any of the **PMRM0003I** codes to view details of the message.

Level	Message
INFO	PMRM0003I : parentver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 - currentver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 type=EJB detail=com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean.create elapsed=72
INFO	PMRM0003I : parentver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 - currentver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 type=EJB detail=com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean.getSize elapsed=3
INFO	PMRM0003I : parentver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 - currentver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 type=EJB detail=com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean.getSize elapsed=0
INFO	PMRM0003I : parentver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 - currentver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 type=EJB detail=com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean.getSize elapsed=1
INFO	PMRM0003I : parentver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 - currentver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 type=EJB detail=com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean.getSize elapsed=0
INFO	PMRM0003I : parentver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 - currentver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 type=EJB detail=com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean.getSize elapsed=0

- ___ g. Scroll through the request metrics records. In addition to the Shopping Cart bean, you also see metrics for PlantsByWebSphere (PBW) servlets and JSPs. Look at the different event types. You see `type=URI`, `type=EJB`, `type=JDBC`, and more, reflecting the request flow.



Information

The **time** and **pid** fields are the start time and ID of the application server process. **Type** and **detail** are the description of the type of operation that is timed and its name. Most important is the measured *elapsed* time in milliseconds, which includes all suboperations that this operation calls. The **reqid** is a unique ID assigned to the request by request metrics.

Two correlators are shown, a parent correlator and a current correlator, representing the upstream request and the current operation. A correlator consists of the comma-separated fields that follow the “parent” or “current” keyword. If the parent and current correlators are the same, as in this case where all field values are the same (including `reqid=1`), then the record represents an operation that occurs as it enters WebSphere Application Server.

4. Disable Request Metrics.

- ___ a. Select **Monitoring and Tuning > Request metrics**.
- ___ b. Clear **Prepare Servers for Request metrics collection**.
- ___ c. Click **OK**.
- ___ d. **Save** directly to the master configuration and log out of the administrative console.
- ___ e. Restart **server1**.

End of exercise

Exercise review and wrap-up

In this exercise, you learned how to turn on different levels of PMI statistics for an application server, monitor an application server with Tivoli Performance Viewer, and configure user settings. You generated some load on applications and examined summary reports and performance modules in Tivoli Performance Viewer. Additionally, you viewed performance messages from the Tivoli Performance Viewer advisor.

You learned how to turn on and configure the Request Metrics tool, and viewed Request Metrics messages in the standard logs of an application server.

Appendix A. Resetting the WebSphere environment

To complete some lab exercises, specific lab exercises must first be completed. However, there might be occasions where you have a problematic configuration or want to skip labs. For these cases, a reset function is provided to set your environment to an appropriate state.

The reset scripts are initiated from a desktop icon and allow users to choose which state they want to restore. The reset scripts can take some time to run, depending on what software is already installed on the lab computer. For example, if none of the exercises are completed, it might be necessary for the reset scripts to install numerous pieces of software; this action would take 5–10 minutes. However, if all of the software installation is completed, it might be necessary for the reset scripts to restore only the profiles directories, and this action would take only 1–2 minutes.

- 1. Run the reset script.
 - a. From the desktop, locate and click the **ResetScripts** icon.



- b. The reset script interface lists the available states that are available.

```

Terminal
File Edit View Terminal Tabs Help

The following reset scripts are available:
=====
1) 1_Initial-state
2) 2_IIM-installed
3) 3_WAS-installed_with_profile1
4) 4_IHS-installed
5) 5_WAS-installed_with_profile1_plus_PlantsByWebSphere
6) 6_WAS-Federated_dmgr-profile1-profile2
7) 7_WAS-Federated_plus_PlantsCluster
8) X_Reset_Plants-DB

To execute a script, enter the script number <#>. To view details for a reset
script, enter d<#>.

Which exercise reset do you wish to execute (1-8, d1-d8, q) [q]:

```

- ___ c. There are a number of reset scripts available. Locate the name of the exercise that directed you here and select the associated reset script state. Running the script that is listed resets the lab machine to a state usable to start that exercise. For example, if you wanted to start **Exercise 6: Installing an application**, you would select the reset script **4_IHS-installed**.
- **Reset script: 1 Initial-state**
 - Exercise 1: Installing IBM Installation Manager
 - **Reset script: 2 IIM-installed**
 - Exercise 2: Installing WebSphere Application Server
 - **Reset script: 3 WAS-installed with profile1**
 - Exercise 3: Installing IBM HTTP Server
 - **Reset script: 4 IHS-installed**
 - Exercise 4: Exploring the administrative console
 - Exercise 5: Assembling an application
 - Exercise 6: Installing an application
 - **Reset script: 5 WAS-installed with profile1 plus PlantsByWebSphere**
 - Exercise 7: Problem determination
 - Exercise 8: Using wsadmin
 - Exercise 9: Creating a federated cell
 - **Reset script: 6 WAS-Federated dmgr-profile1-profile2**
 - Exercise 10: Clustering and workload management
 - **Reset script: 7 WAS-Federated plus PlantsCluster**
 - Exercise 11: Configuring the service integration bus
 - Exercise 12: Configuring WebSphere security
 - Exercise 13: Configuring application security
 - Exercise 14: Configuring SSL for WebSphere
 - Exercise 15: Working with the Liberty profile
 - Exercise 16: Using the performance monitoring tools
 - **Reset script: X Reset Plants-DB**
 - This script rebuilds the Plants database.



Information

If you would like more information about the specific reset scripts, type the letter **d** followed by the number for the reset script.

```

Terminal
File Edit View Terminal Tabs Help
which exercise reset do you wish to execute (1-8, d1-d8, q) [q]: d2
Details for: /usr/software/reset/reset_scripts/reset_2_IIM-installed_details.txt
=====
This reset script ensures that the image has IIM installed.

If you need other software (like WAS or IHS) to be uninstalled, those need
to be uninstalled manually.

The script first stops all java processes. It then does a silent
install of IIM.

=====
Press Enter to continue...

```

- __ d. Depending on how much work the reset script must do, the wait is several minutes. When the script finished, press **Enter** to close the window.

```

Terminal
File Edit View Terminal Tabs Help
To execute a script, enter the script number <#>. To view details for a reset
script, enter d<#>.

which exercise reset do you wish to execute (1-8, d1-d8, q) [q]: 7
Running script: /usr/software/reset/reset_scripts/reset_7_WAS-Federated_plus_Pl
antsCluster.sh
=====
This script will reset the profiles to a state expected at the
7_WAS-Federated_plus_PlantsCluster
Do you want to continue (y/n) [y]: 

```


Appendix B. List of Linux commands

alias	Creates an alias for a command
apropos	Provides a list of man pages relevant to a particular subject
cat	Type file out cat <file>
cd	Change directory to absolute or relative path cd /home/waslocal cd .. (change directory up one level)
chgrp	Change file group ownership
chmod	Change access mode bits on files chmod 744 <file> (change mode for user=all, group=rw, and other=rw) chmod g+rwx <file> (change mode for group to have all permissions) chmod a=x <file> (change mode and give all execute permission)
chown	Change file owner
clear	Clear the screen
cp	Copy files
date	Display or set date
df	Show free disk space df -k (shows the free space in KB) df -m (shows the free space in megabytes)
diff	Show differences between two files
du	Show disk usage
echo	Show output status text to the screen or a file echo \$DISPLAY (shows the value for the DISPLAY variable) echo \$PATH (shows the value for the PATH variable)
emacs	Start an emacs editing session
env	Display the list of current environment variables
exit	Exit a shell script
export	Set the value of one or more shell variables export DISPLAY=:0.0
find	Finds and locates files

	find / -name <file>
firefox	Launch a Firefox browser window firefox -p <profile-name> -no-remote
ftp	File transfer protocol
gedit	Start a gedit editing session, text editor for the GNOME desktop gedit <file>
gnome-terminal	Launch a GNOME terminal window
grep	Search files for text patterns grep <string> <file> grep localhost /etc/hosts
groups	Show the groups to which a user belongs
gzip	Compress and uncompress gzip <file>
head	Show the first few lines of a file
help	Access help for shell commands
history	Show the list of previous commands
hostname	Show the host name
ifconfig	View, enable, and disable a network interface, IP address, broadcast address, and subnet mask
jobs	List child processes of current process
kill	Terminate a running command or process kill <PID> kill 1423
ln -s	Create a symbolic link ln -s <file/directory> <link>
ls	List files or directories ls -l (long format listing) ls -la (long format list all files, including normally hidden files)
man	Use online reference manuals (man pages) to get information about a command man mkdir (displays the man page for the <code>mkdir</code> command)
mkdir	Create a directory

more	Type out a file one screen at a time more SystemOut.txt
mount	Instruct the operating system that a file system is ready to use, and associate it with a particular point in the file system hierarchy
mv	Move or rename files or directories mv <from> <to>
passwd	Set password
ps	Show processes ps -ef (show all processes and do a full listing) ps -ef grep java (search the list of all processes for the word java)
pwd	Print your present working directory
reboot	Reboot system
rm	Remove files rm /tmp/myfile
rmdir	Remove directories
sed	Stream editor; edit one or more files without user interaction
ssh	Secure Shell, a network protocol that allows data to be exchanged by using a secure channel between two networked devices
su	Become a substitute user su - db2inst1
shutdown	Allow a user to change system state, taking system down shutdown -h (halt after shutdown) shutdown -P (halt action is to turn off power)
source	Read and execute commands from a file in the current script
tail	Show the last few lines of a file tail SystemOut.txt (shows the last 10 lines of the file) tail -f SystemOut.txt (show output as it is appended to the file)
tar	Create or expand archive files tar cvf myfile.tar <directory> (create .tar file from directory) tar xvf myfile.tar . (expand .tar file here)
telnet	Connect to another system
time	Display and set time

top	Dynamically display process information
touch	For one or more files, update the access time and modification time to current date and time
vi	Visual text editor
wc	Print a character, word, and line count for files
whereis	Locate files
which	Used to identify the location of executables within the path
whoami	Print the effective user name
xterm	Launch an xterm window
~	HOME directory of the current user cd ~/temp
 	Pipe ps -ef grep java (pipes the ps command into the grep command)
<	Redirect
>>	Append date >> /tmp/mylog
!	Recall ! 34 (recalls the 34th command from the history list) !ps (recalls the last command that starts with ps)
./	Execute file in current working directory . ./startManager.sh

Glossary of abbreviations and acronyms

A

AB ApacheBench
AFS Andrew File System
AIX Advanced IBM UNIX
AMI asynchronous message interface
Ant Another Neat Tool
AOL America Online
APAR Authorized Program Analysis Report
API application programming interface
ARFM application request flow manager
ARM Application Response Measurement
ASCII American Standard Code for Information Interchange
AST Application Server Toolkit

B

BSF Bean Scripting Framework

C

CA certificate authority
CICS Customer Information Control System
CIM centralized installation manager
CIP custom installation package
CMP container-managed persistence
CMS Certificate Management System
CMT Configuration Migration Tool
CN common name
COBOL Common Business-Oriented Language
CORBA Common Object Request Broker Architecture
CP caching proxy
CPU central processing unit
CSlv2 Common Secure Interoperability Protocol Version 2
CSS Cascading Style Sheets

D

DB database
DC domain controller
DCS Distribution and Consistency Services
DD deployment descriptor

DHCP Dynamic Host Configuration Protocol
DLL Dynamic Link Library
DMgr deployment manager
DMZ demilitarized zone
DN distinguished name
DNS Domain Name System
DRS data replication service
DTD document type definition
DVD digital versatile disc

E

EAR enterprise archive
EE Enterprise Edition
EIS enterprise information system
EJB Enterprise JavaBean
EJS Enterprise Java Services
EL Expression Language
ENC Enterprise Naming Context
ERP enterprise resource planning
ESB Enterprise service bus
ESI Edge Side Include

F

FIPS Federal Information Processing Standard
FQDN fully qualified domain name
FTP File transfer protocol

G

GA generally available
GB gigabyte
GC garbage collection
GCD greatest common divisor
GCMV Garbage Collection and Memory Visualizer
GIF Graphics Interchange Format
GMT Greenwich Mean Time
GPS global positioning system
GSS Generic Security Services
GTK GNU GUI Tool Kit
GUI graphical user interface

H

HA high availability or highly available
HACMP High-Availability Cluster Multi-Processing
HAM high availability manager
HFS Hierarchical File System
HPEL High Performance Extensible Logging
HPUX Hewlett Packard UNIX

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

HTTPD HTTP Daemon

HTTPS HTTP over SSL

I

IADT IBM Assembly and Deploy Tools

IBM International Business Machines Corporation

IDE integrated development environment

IIOP Internet Inter-ORB Protocol

IMCL Installation Manager command-line

IMS Information Management System

I/O input/output

IP Internet Protocol

ISC Integrated Solutions Console

IPSEC IP Security

ISMP InstallShield MultiPlatform

IT information technology

IVT Installation Verification Tool

J

J2C Java 2 Connector

J2EE Java 2 Enterprise Edition

J2SE Java 2 Platform Standard Edition

JAAS Java Authentication and Authorization Service

JACC Java Authorization Contract for Containers

Jacl Java Command Language

JAF Java Activation Framework

JAR Java archive

JASPI Java Authentication Service Provider Interface for Containers

Java EE Java Platform, Enterprise Edition

JAXB Java Architecture for XML Binding

JAXP Java API for XML Processing

JAXR Java API for XML Registries

JAX-RPC Java API for XML Remote Procedure Calls

JAX-RS Java API for XML-based Remote Procedure Calls

JAX-WS Java API for XML Web Services

JCA Java EE Connector Architecture

JCE Java Cryptology Extension

JDBC Java Database Connectivity

JDE JD Edwards

JDK Java Development Kit

JIT just-in-time

JMS Java Message Service

JMX Java Management Extensions

JNDI Java Naming and Directory Interface

JPA Java Persistence API

JPG Graphics file type or extension (lossy compressed 24 bit color image storage format developed by the Joint Photographic Experts Group)

JRE Java Runtime Environment

JSF JavaServer Faces

JSP JavaServer Pages

JSR Java Specification Request

JSTL JavaScript Tag Library

JTA Java Transaction API

JVM Java virtual machine

K

L

LAN Local area network

LB Load Balancer

LDAP Lightweight Directory Access Protocol

LSD Location service daemon

LTPA Lightweight Third Party Authentication

M

MAC message authentication code or media access control

MAPs message addressing properties

MDB message-driven bean

ME messaging engine

MQ Message Queue

MQI Message Queue Interface

MVS Multiple Virtual System

N

NAS network attached storage

NAT network address translation

NIC network interface card

NIM Network Installment Management

NTP Network Time Protocol

O

ODR on demand router

OLTP online transaction processing

OOM OutOfMemory

ORB Object Request Broker

OS operating system

OVF Open Virtualization Format

P

PAR Parchive Index File
PBW PlantsByWebSphere
PCT Plug-ins Configuration Tool
PD problem determination
PFBC properties file based configuration
PGP Pretty Good Privacy
PHD portable heap dump
PHP personal home page
PID process identifier
PKI public key infrastructure
PMAT Pattern Modeling and Analysis tool
PME programming model extensions
PMI Performance Monitoring Infrastructure
PMR problem management record
PMT Program Management Tool
POJO plain old Java object
PWB PlantsByWebSphere

Q

QA quality assurance
QoS quality of service

R

RA registration authority
RAM random access memory
RAR resource archive
RACF Resource Access Control Facility
RAS reliability, availability, and serviceability
RC return code
RDBMS relational database management system
RDN relative distinguished name
REST Representational State Transfer
RM request metrics
RMI Remote Method Invocation
RMI/IOP Remote Method Invocation Over Internet Inter-ORB Protocol
RUP Rational Unified Process
RXA Remote Execution and Access

S

SAAJ SOAP with Attachments API for Java
SAF System Authorization Facility
SAM Security Access Manager
SAR SIP application resource
SAS Secure Association Service
SBDT Smart Business Development and Test

SCA Service Component Architecture

SCADA supervisory control and data acquisition

SCM source code management

SDO Service Data Objects

SDK software development kit

SDLC systems development lifecycle

SFSB stateful session bean

SIB or **SIBus** service integration bus

SIP Session Initiation Protocol

SMTP Simple Mail Transfer Protocol

SNMP Simple Network Management Protocol

SOA service-oriented architecture

SOAP A lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment. Usage note: SOAP is not an acronym; it is a word in itself (formerly an acronym for Simple Object Access Protocol)

SPI service provider interface

SPNEGO Simple and Protected GSS-API Negotiation Mechanism

SPOF single point of failure

SQL Structured Query Language

SSB stateful session bean

SSL Secure Sockets Layer

SSO single sign-on

StAX Streaming API for XML

SUSE Software und System Entwicklung (German: Software and Systems Development)

SVC supervisor call

SVG Scalable Vector Graphics

SWAM Simple WebSphere Authentication Mechanism

T

TCL Tool Command Language

TCP/IP Transmission Control Protocol

TCP/IP Transmission Control Protocol/Internet Protocol

TKCARS1 Toolkit for Custom and Reusable Solution Information

TLS Transport Layer Security

TMDA Thread and Monitor Dump Analyzer

TP Trade Processor (application in lab exercises)

TT Trade Ticker (application in lab exercises)

U

UDDI Universal Description, Discovery, and Integration

UI user interface

UNC Universal Naming Convention

UNIX Uniplexed Information and Computing System

UOW unit of work

URI Uniform Resource Identifier

URL Uniform Resource Locator

UTC Universal Test Client

V

VMM virtual member manager

VPN virtual private network

W

WAR web archive

WLM workload management

WS web services

WS-AT web services atomic transaction

WS-BA web services business activity

WS-COOR web services coordination

WSDL Web Services Description Language

WS-I Web Services Interoperability

WSIF Web Services Invocation Framework

X

XA Extended Architecture

XCT Cross-Component Trace

xJCL XML Job Control Language

XML Extensible Markup Language

XTP extreme transaction processing

Y

Z

zMMT z/OS Migration Management Tool

z/OS zSeries operating system

zPMT z/OS Profile Management Tool

IBM
®