

*Administration of WebSphere
Application Server V6*

(Course Code WF381)

Instructor Exercises Guide

ERC 6.0

IBM Certified Course Material

Trademarks

IBM® is a registered trademark of International Business Machines Corporation.

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX®	Cloudscape™	DB2®
developerWorks®	Domino®	e-business on demand™
HACMP™	Passport Advantage®	Rational®
Tivoli®	WebSphere®	z/OS®

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Intel is a trademark of Intel 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.

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

Other company, product and service names may be trademarks or service marks of others.

March 2005 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.

© Copyright International Business Machines Corporation 2002, 2005. All rights reserved.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Trademarks	v
Instructor exercises overview	vii
Exercises description	xi
Exercise 1. WebSphere Application Server installation	1-1
Exercise 2. WebSphere Information Center installation (optional)	2-1
Exercise 3. Explore WebSphere Application Server through the administrative console	3-1
Exercise 4. Using AST to assemble the Trade Application	4-1
Exercise 5. Installing the Trade Application	5-1
Exercise 6. Problem determination (optional)	6-1
Exercise 7. Experimenting with wsadmin	7-1
Exercise 8. Use WebSphere Rapid Deployment to install the QuoteWS Application	8-1
Exercise 9. Federating a cell	9-1
Exercise 10. IBM Http Server installation	10-1
Exercise 11. Clustering	11-1
Exercise 12. Install Trade Listener Application	12-1
Exercise 13. Configuring SSL (optional)	13-1
Exercise 14. Enabling WebSphere and J2EE security	14-1
Exercise 15. Integrated Performance Viewer (optional)	15-1

Appendix A. General UNIX commands and information	A-1
Appendix B. General Windows commands and information.....	B-1
Appendix C. General AIX commands and information.....	C-1
Appendix D. General Linux commands and information.....	D-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® is a registered trademark of International Business Machines Corporation.

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX®
developerWorks®
HACMP™
Tivoli®

Cloudscape™
Domino®
Passport Advantage®
WebSphere®

DB2®
e-business on demand™
Rational®
z/OS®

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Intel is a trademark of Intel 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.

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

Other company, product and service names may be trademarks or service marks of others.

Instructor exercises overview

The purpose of the exercises in this course is to give the students hands-on experience in working with WebSphere Application Server V6 and its components.

The exercises support Windows 2000 and various UNIX platforms. The exercise instructions are Windows-centric, as are the screen captures. UNIX specific instructions are contained in UNIX information blocks inline with the rest of the exercise steps. Other platform specific information, such as program file location folders, is found in the appendix.

Most of the exercises revolve around setting up the Trade family of applications. The students will first install WebSphere Application Server and create a single server "base" profile, then assemble and install the various parts of the Trade application.

The Trade family of applications is made up of three ear files: TradeApplication, QuoteWS and TradeListener. A complete description and composition of these enterprise application archives can be found in the lecture Introduction to the Trade family of applications.

Later in the class, additional profiles are created to make up a working cell and cluster. The installed applications are then re-deployed to take advantage of single cell administration and workload management.

Exercise 1: Students install WebSphere Application Server V6 Network Deployment on the lab machine and create a single server "base" profile. After installation, verification steps will be performed including starting the default server and testing the sample applications using the First Steps console. In addition, students use the administrative console to examine the default installed applications.

Note: DB2 and other software required by the case study will be preloaded on the lab machines.

Exercise 2: This exercise is optional. Students will install WebSphere Application Server V6 Help System and libraries and learn how to navigate and search the documentation.

Exercise 3: Students explore the WebSphere Application Server V6 Administrative Console and the underlying configuration files that store the node and cell configuration.

Exercise 4: Students will explore the capabilities of the Application Assembly Tool (AST). The major topics are: learn to navigate AST,

import JARs and WARs, define data source at application scope, define authentication alias at application scope, export an enhanced EAR file ready for deployment. They also explore the various module types that make up the Trade application.

Exercise 5: Students install the TradeApplication.ear enterprise application in WebSphere Application Server V6 using the administrative console. After installation they make sure the application works as expected by thoroughly testing it.

Exercise 6: Students will examine server and application log files for the purposes of troubleshooting. They use the administrative console and other tools to do problem determination such as: locating and viewing important log files, using the Log Analyzer and AST tools to view the service log, using the administrative console to enable tracing of an application server

Exercise 7: Students explore the new capabilities of using the Jython scripting language in wsadmin. They also use Jacl in a parallel exercise. They learn how to use wsadmin interactively, start wsadmin using Jacl and Jython, use wsadmin with a single command or script, configure wsadmin to use a default profile, configure wsadmin's default scripting language. Lots of examples are supplied with this lab, from simple one liners to complete scripts to install and configure an application.

Exercise 8: Students use WebSphere Rapid Deployment (WRD) to install the QuoteWS component of the Trade application and test the application using database supplied quotes (instead of Internet quotes)

Exercise 9: Students create a WebSphere cell through the generation of a Deployment Manager profile followed by the federation of application server profiles. They also make backup copies of the configuration.

Exercise 10: Students install the IBM HTTP Server. Both the IBM HTTP Server V6 and its plug-ins are installed and enabled to work with WebSphere Application Server V6. The DefaultApplication is mapped to the Web server and the plug-in configuration file is regenerated. After installation, testing takes place to ensure the IBM HTTP Server was installed successfully and the Web server is working in concert with the application server. Students create an unmanaged node and include the Web server in the node to see how IHS can now be managed from the administrative console.

Exercise 11: Students create a cluster for the purpose of implementing workload management. While creating the cluster two nodes are added. After the cluster is created the Trade application is configured

to run in the cluster. The application is tested to ensure workload management and failover work as expected.

A replication domain is setup to use memory-to-memory replication and again the application is tested to ensure that session failover works as expected

Exercise 12: Students install the third module of the Trade Application. The Listener module contains a Message Driven Bean (MDB) which listens on a JMS queue for transactions made by users of the Trade Application. For this module to work the WebSphere Application Server needs to be configured to handle asynchronous messaging.

Aside from installing the Trade Listener module, which is very straightforward, this module deals with the complexities of setting up the SIBus, Messaging Engine (ME), and JMS queues.

Exercise 13: Students secure the communications between a browser and the Web server by configuring the IBM Http Server to use https. They create a Keyring with a self-signed key, and then they define a virtual host within the httpd.conf to use that keyring for https communications to the IBM Http Server.

Lastly, the students test connectivity to the IBM Http Server using https.

Exercise 14: Students secure the WebSphere Application Server V6 Administrative Console and verify that access is limited according to the role of the logged in user.

This exercise also secures the administration module of the Trade application. The application is tested with Global security enabled and an explanation, through exploration of configuring J2EE security in Application Server Toolkit (AST), is presented.

Exercise 15: Students use the integrated Tivoli Performance Viewer to monitor various application server resources in real time. Some of the activities of this exercise include: start and stop Monitoring Process, monitor application server resources, use the features and functions supplied with the Tivoli Performance Viewer (Performance Advisor), work with Request Metrics Data.

Exercises description

The exercises in this course provide hands-on experience in working with WebSphere Application Server V6 and its components.

The exercises support Windows 2000 and various UNIX platforms. The exercise instructions are Windows-centric, as are the screen captures. UNIX specific instructions are contained in UNIX information blocks inline with the rest of the exercise steps. Other platform specific information, such as program file location folders, is found in the appendix.

In general the exercises should be performed in sequence and the required parts of exercises should always be performed. Some exercises are optional and some exercises contain optional sections. The optional exercises can be skipped if desired.

Exercise 1. WebSphere Application Server installation

What this exercise is about

This lab covers the installation of WebSphere Application Server Network Deployment Version 6.0. In Version 6.0, installing WebSphere Application Server Network Deployment is a two-step process. The first step is using the installation wizard to install a set of shared product binaries. The second step is using the Profile creation wizard to create a profile.

After installation, you will test the product to ensure the WebSphere Application Server was installed successfully. You will also create an application server profile, profile1.

What you should be able to do

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

- Describe the installation process for WebSphere Application Server V6
- Verify the installation of WebSphere Application Server V6
- Create a WebSphere profile
- Create configuration backups
- Describe the directories and configuration files for WebSphere Application Server with profiles
- Describe how to use WebSphere Application Server command utilities

Introduction

In this exercise you will install WebSphere Application Server Network Deployment V6.0.

WebSphere Application Server relies on TCP/IP networking, so you must have TCP/IP correctly configured, and it is important that the machine host name remains unchanged.

Your lab machine should be configured appropriately. Make sure you know your machine's host name because you will need it in the exercises.

A Windows administrator user ID has been created for you. You should use this ID to log in to Windows and to configure services and database access.

- Userid: **userid**
- Password: **was1edu**

Note: Microsoft Windows 2000 passwords are case sensitive.

UNIX: In the UNIX environments, you will be using:

- Userid: **root**
- Password: **was1edu**

Information: The exercise screen captures are Windows-centric. The majority of the instructions are applicable across all WebSphere compatible platforms, though. In cases where there are differences in the Windows instructions versus the UNIX platform instructions, a **UNIX:** note will be added. There will be references in places to added appendices noted by the **[Appendix]** marker. Refer to your platform's appendix for appropriate information.

The standard classroom lab machines have a host name of **was6hostXX** (where **XX** is 01, 02, and so forth). If you are performing the labs on a machine with a different host name, substitute that name where appropriate.

On Windows, most paths are listed as appropriate variables. When a **<variable>** is found, use the table below to determine the correct value for this variable.

<variable>	Value
<software_dir>	C:\Software
<software_cds>	C:\Software_CDs
<was_root>	C:\Program Files\IBM\WebSphere\AppServer
<profile_root>	C:\Program Files\IBM\WebSphere\AppServer\profiles
<ihs_root>	C:\Program Files\IBM HTTP Server
<plugin_root>	C:\Program Files\IBM\WebSphere\Plugins
<db2>	C:\Program Files\IBM\SQLLIB

Instructor exercise overview

In this exercise the student will learn how to install WebSphere Application Server (WAS) Network Deployment Version 6.0. Note and point out to your students the places where special **UNIX** variables are given and the place where the **[Appendix]** is listed so that the students will be aware of the UNIX variables.

The first step is to log in to Windows 2000, using the userid **userid**, and password **was1edu**. This ID should have administrator group membership. If you are on a UNIX system, the userid is **root** and the password is **was1edu**.

The student will then install WebSphere Application Server 6.0. The WebSphere Application Server 6.0 product installation file has already been downloaded and unzipped onto the student's machine, in directory: **<software_cds>\WAS6**. This is the directory where the student will start their WebSphere Application Server installation process.

The students need to select the node name and their host name during installation. In the labs we have used the convention **was6hostXX** (where **XX** is 01, 02, etc). If your machines have a different convention, remind students to use the correct names. Verify that machines can connect to each other using those names.

The Fedora image will not appear to meet the system prerequisites. Instead, the user is prompted with two options: "Click Cancel to stop the installation and to install a supported operating system" or "Click Next to continue the installation." Students should click Next and continue with the installation. Installation will complete successfully.

After installation, the students learn to start the WebSphere Application Server server using command line tools and use the Installation Verification Test and the snoop servlet to test the application server installation.

The instructor should become familiar with the different product components and what they do. Also the instructor should be familiar with the tools such as the administrative console. Students are sure to discover and have questions beyond the scope of this lab.

After Exercise 1, WebSphere Application Server Network Deployment will have been installed and a profile, profile1, will be created.

Exercise instructions

Logging in

When you boot up your machine, you will be prompted for a userid and password. At this prompt, enter:

- Userid: **userid**
- Password: **was1edu**

Be sure to use lower case, for the Windows login fields are case-sensitive.

If you are already logged in, but not as **userid**, then log off your current ID and log in as **userid**.

- a. Press Ctrl-Alt-Delete for the Windows Security window.
- b. Click the **Log Off...** button. Are you sure? Click **Yes**.
- c. At the login window, type in **userid** and **was1edu** into the correct fields.

UNIX:

- Userid: **root**
- Password: **was1edu**

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

- a. In the desktop environment tool menus (such as the CDE Task Bar or FrontPanel), click the Exit icon.
- b. Click **Ok** to continue the logout.

At the login window, type in **root** and **was1edu** into the correct fields.

Install WebSphere Application Server Network Deployment, V6

The WebSphere Application Server Network Deployment V6 installation files have already been downloaded and unzipped in a software directory on your machine (See your platform appropriate [\[Appendix\]](#) for the directory path).

Now, install the WebSphere Application Server Network Deployment V6.

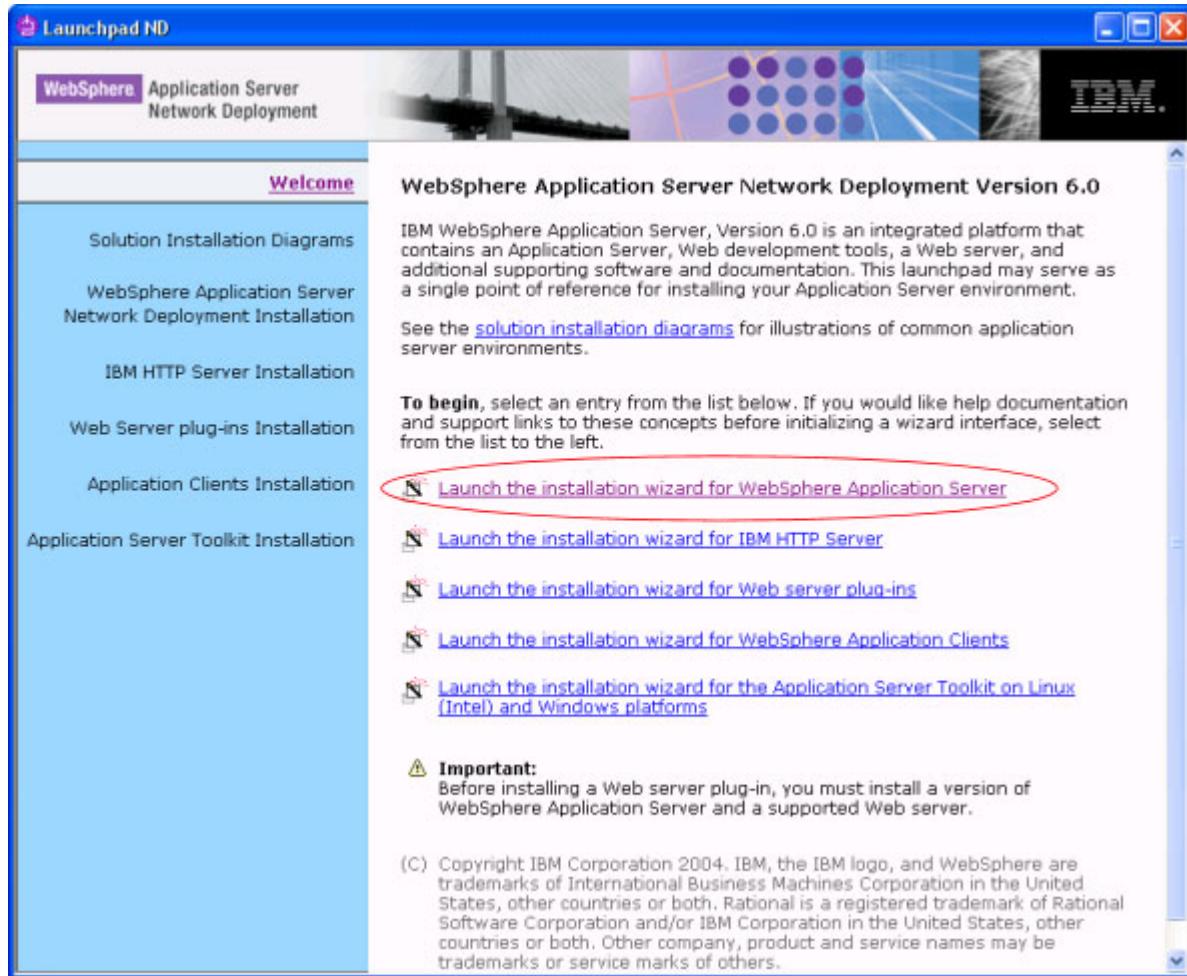
- 1. Start the WebSphere Application Server Launchpad.
- a. Bring up Windows Explorer (right-click **My Computer** —> **Explore** or right-click **Start** —> **Explore**).
- b. Navigate the tree to <**software_cds**>\WAS6\.

__ c. Run (Double-click) **launchpad**

UNIX:

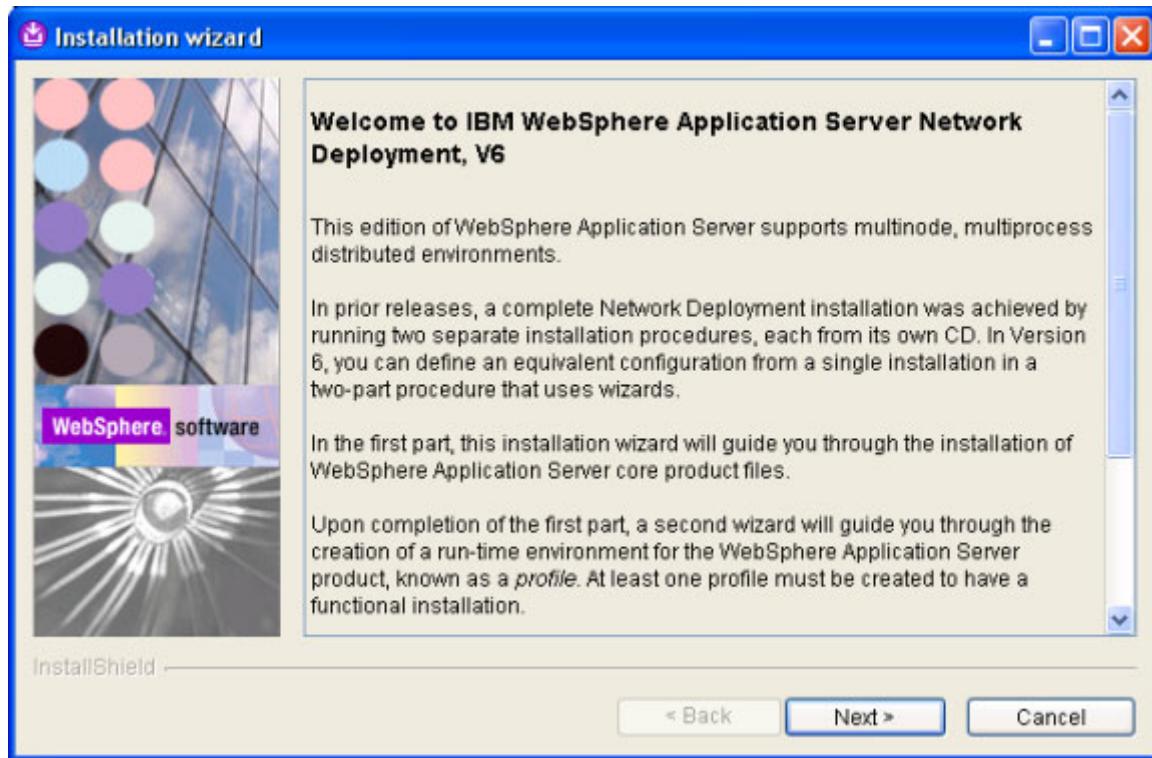
- __ a. Open a terminal window and navigate to <software_cds>/WAS6/.
- __ b. Invoke `./launchpad.sh`.

- __ 2. The Launchpad will be displayed. On the Welcome panel there are links that launch the installer programs for all of the installable components..

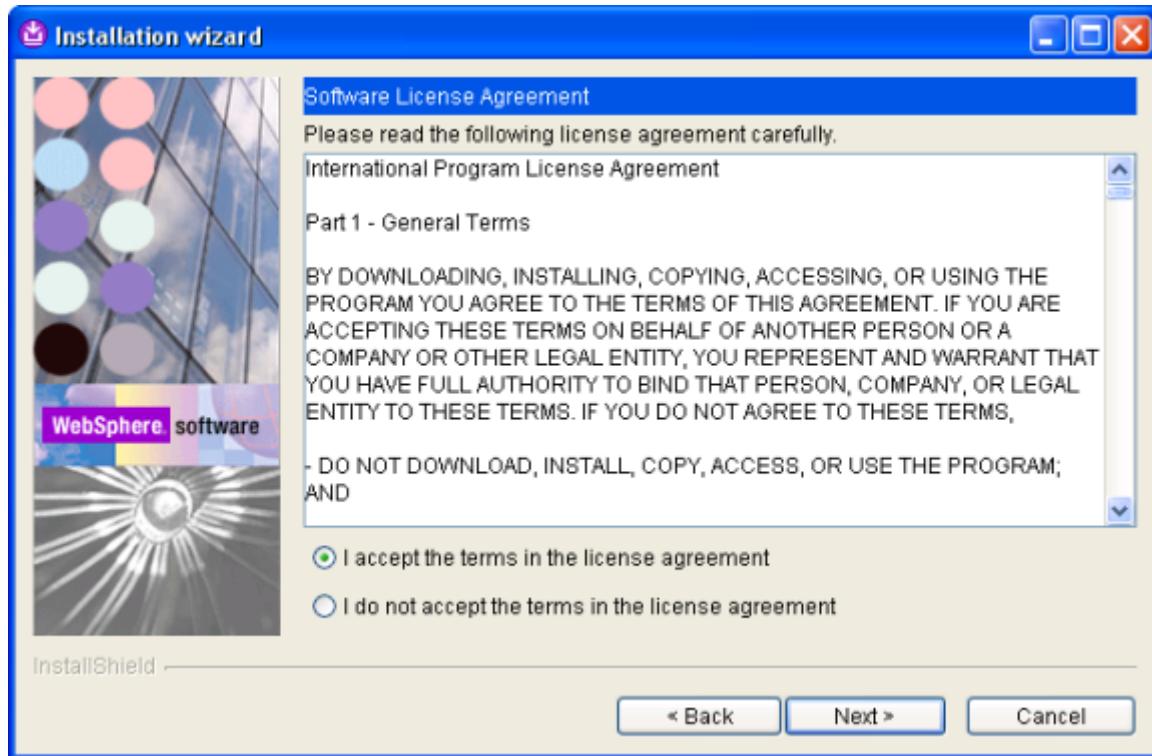


- __ 3. Click **Launch the installation wizard for WebSphere Application Server** to launch the installation wizard for WebSphere Application Server.

- __ a. The Welcome screen will appear and looks like the image below. Click **Next** to continue the installation.



- __ b. Accept the license agreement and click **Next**.

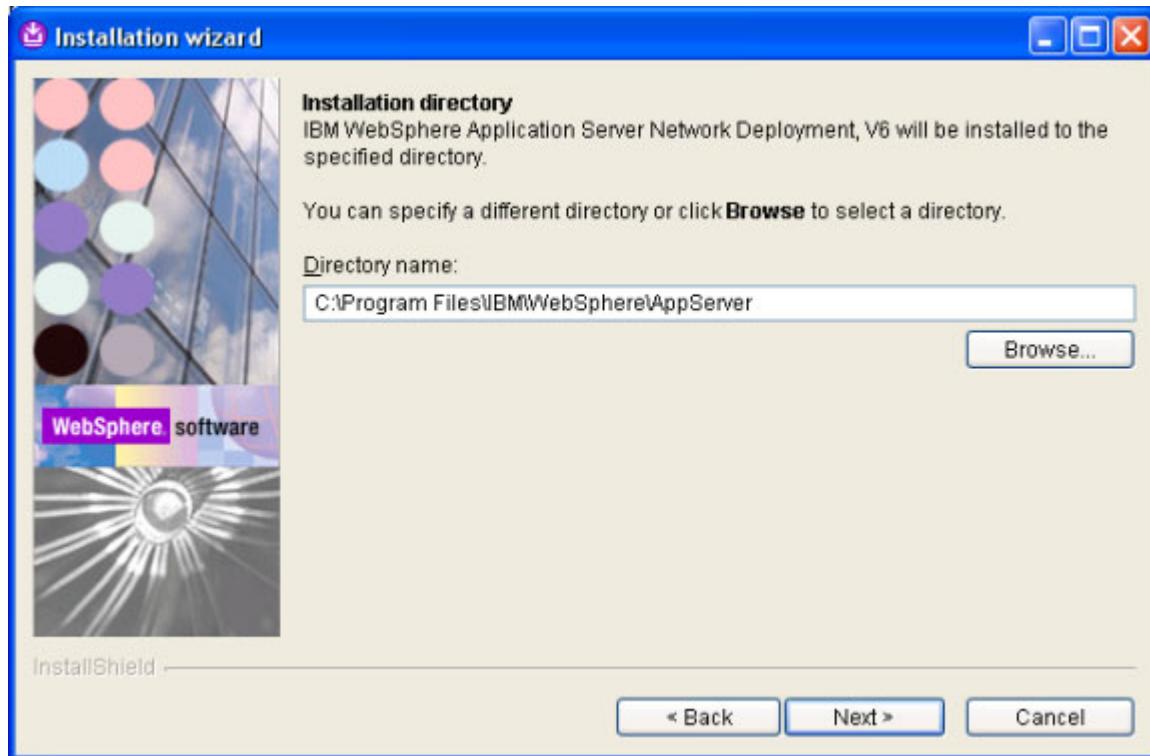


- ___ c. The installation wizard will check system prerequisites. If it detects any incorrect prerequisites a warning page will be displayed. If it detects an existing installation of WebSphere Application Server you will be given the opportunity to add features to the existing copy, install a new copy or perform an upgrade of a trial installation to a full product.

The lab system should meet the prerequisites and should not have any version of WebSphere Application Server already installed. Click **Next** to continue the installation.

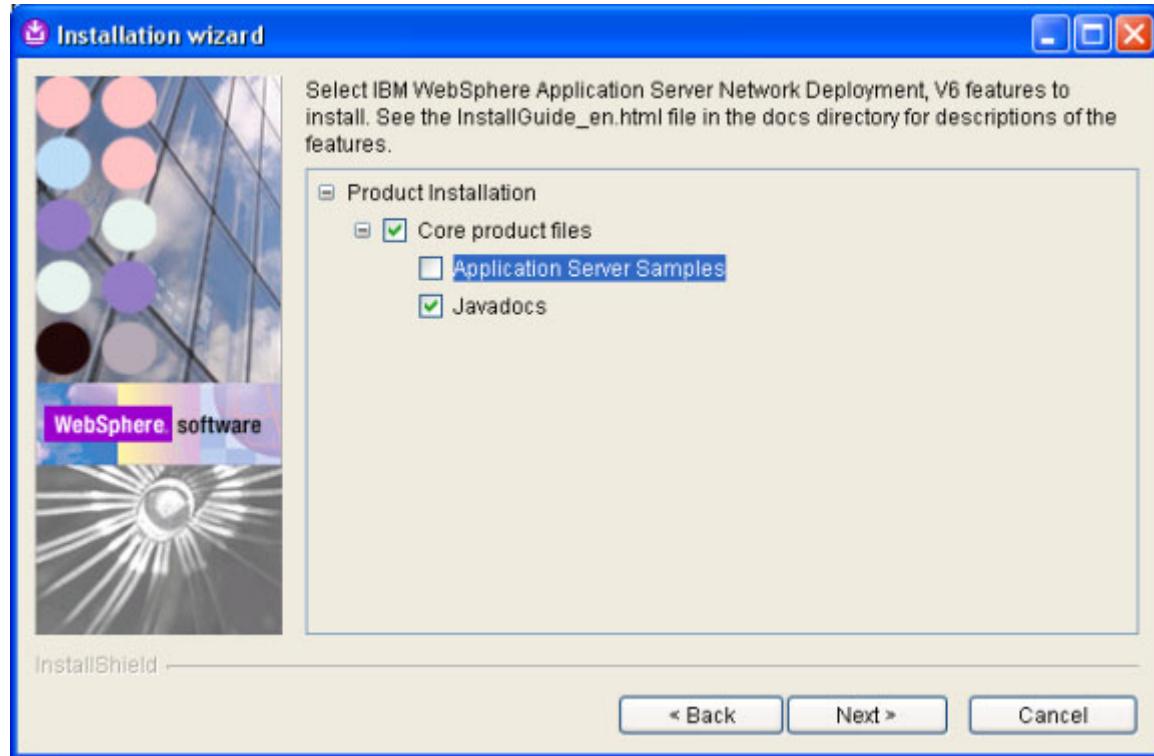
Information: If the lab system does not meet the prerequisites check with your instructor before continuing with the installation.

- ___ d. The next page lets you change the installation directory for WebSphere Application Server. Note the default installation directory location but do not change this unless there is insufficient space on the **C:** drive to install all of the components. Click **Next**.



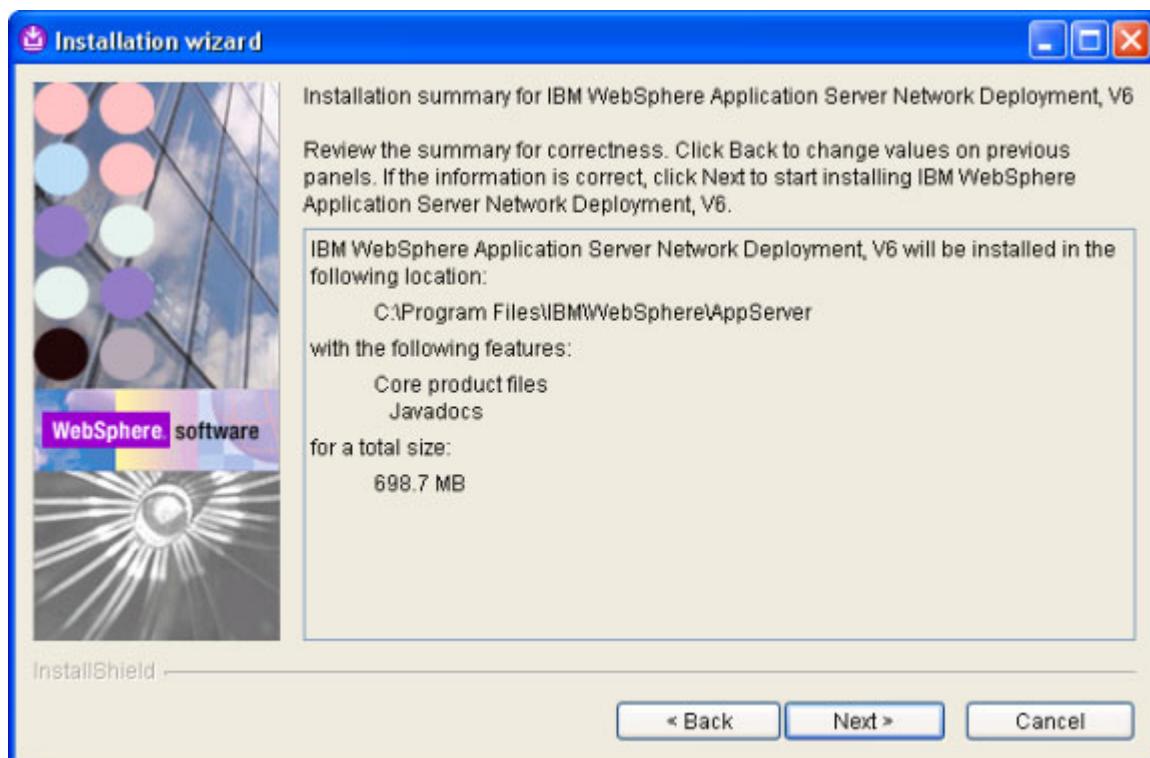
UNIX: In UNIX based systems, the directory structures will be different. Refer to the appropriate [\[Appendix\]](#) for the appropriate directory structures. The install wizard gives the appropriate default values for these directories.

- ___ e. The next page lists the features to install. All components are selected by default. For these exercises you do not want the samples to be installed. Make sure the following component is **not** selected:
- Application Server samples

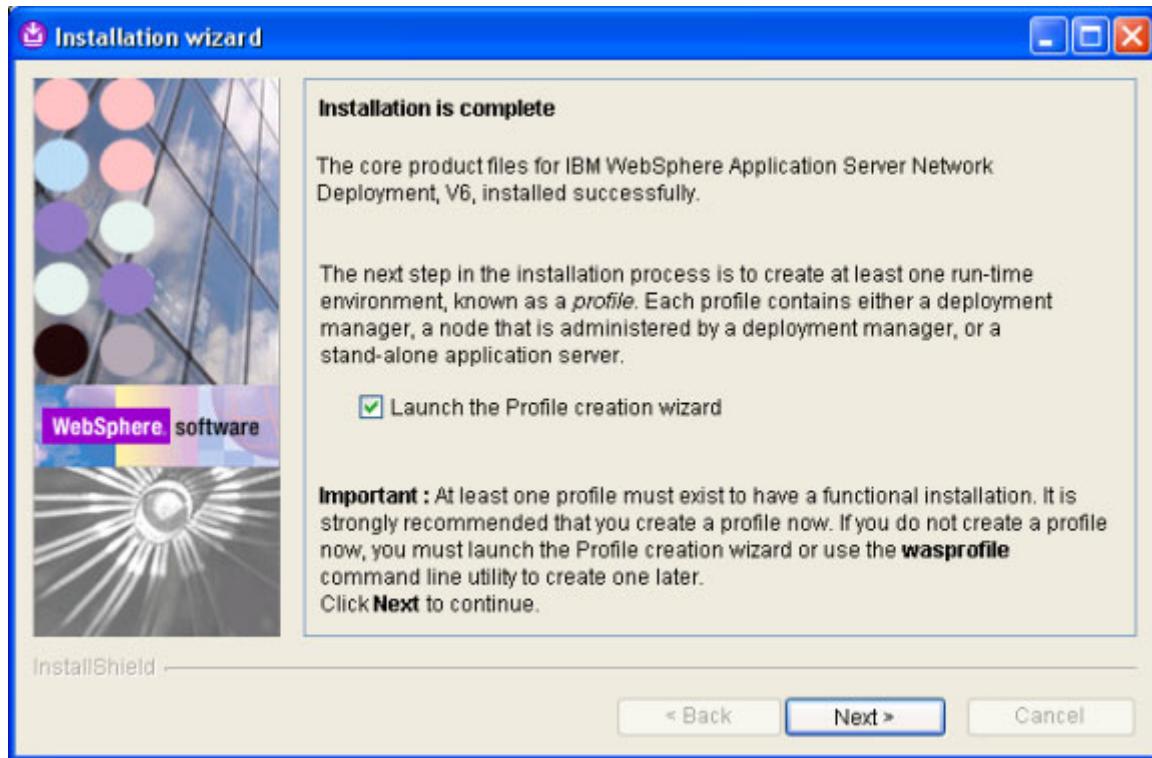


Click **Next** to continue the installation.

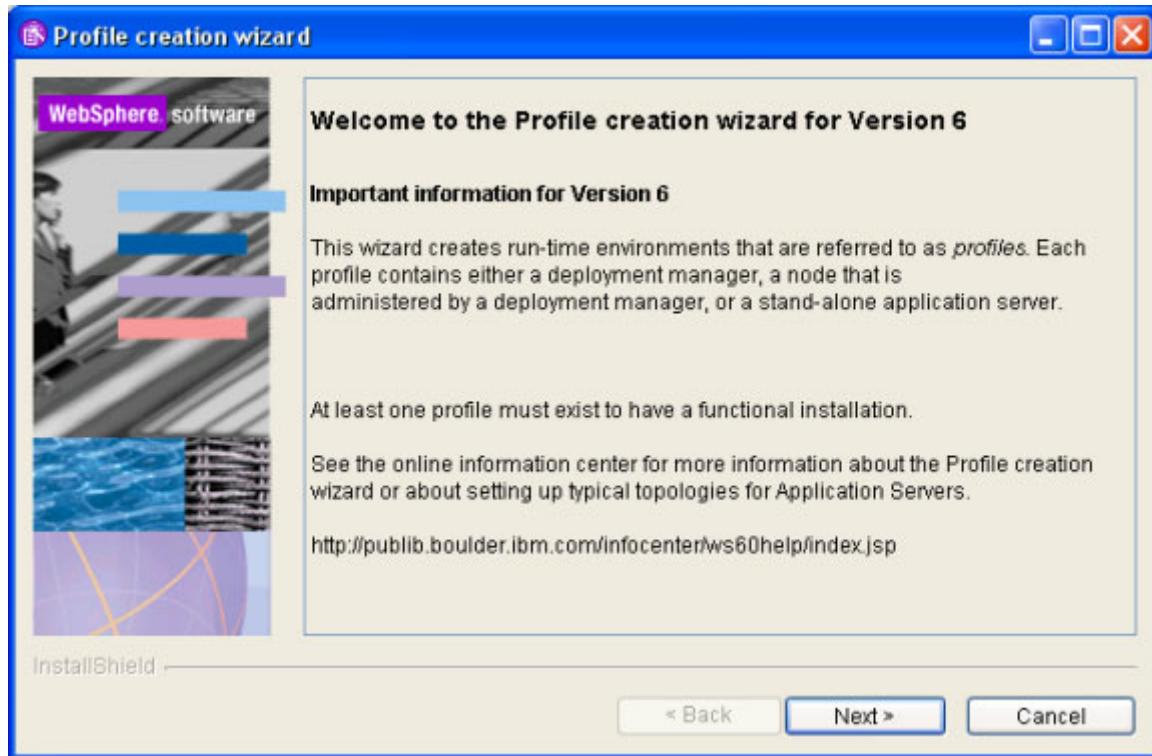
- ___ f. The last page is simply a confirmation of all the choices you have made on previous screens. Double-check your information with what is shown below. Click **Next** to begin the installation.



4. After the installation is complete, you then need to create a profile. Make sure the **Launch the Profile creation wizard** box is checked. Use the Profile creation wizard to create application server run-time environments, called profiles. Click **Next**.



- a. On the Welcome screen for the Profile creation wizard click **Next** to continue.

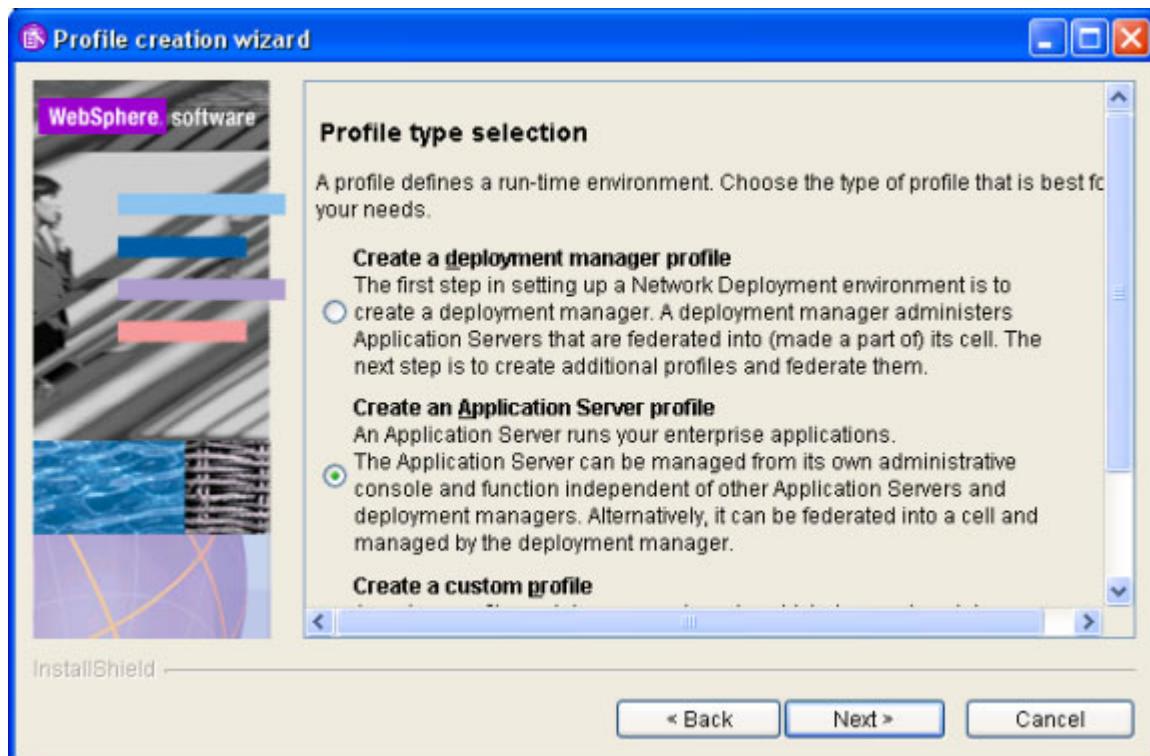


You can create multiple application servers on your system without installing the product again by creating profiles. When using the Profile creation wizard, there are three types of profiles you can create:

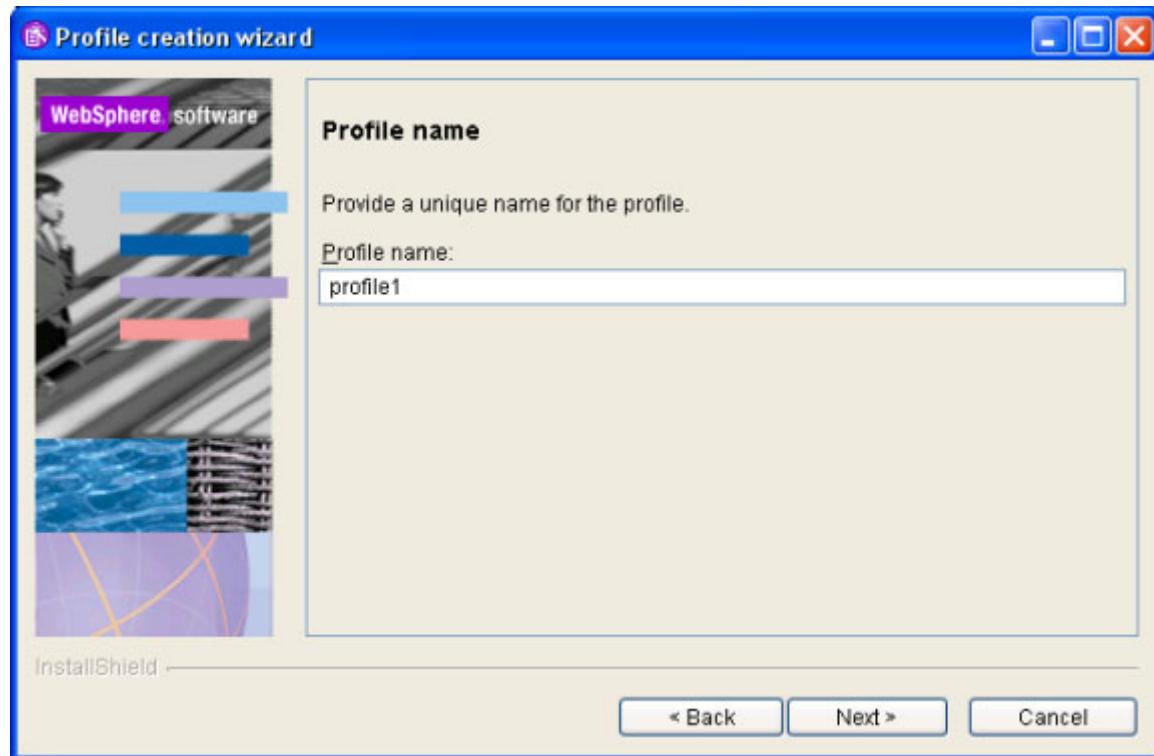
- Deployment Manager Profile which creates a new instance of deployment manager. Each instance of deployment manager is a unique cell.
- Application Server Profile which creates a new instance of a standalone node with a single application server. Standalone nodes have only one application server.
- Custom Profile which creates an automatically federated/managed node with no pre-defined application servers. Managed nodes can have multiple application servers.

In this exercise create a standalone application server profile called **profile1**.

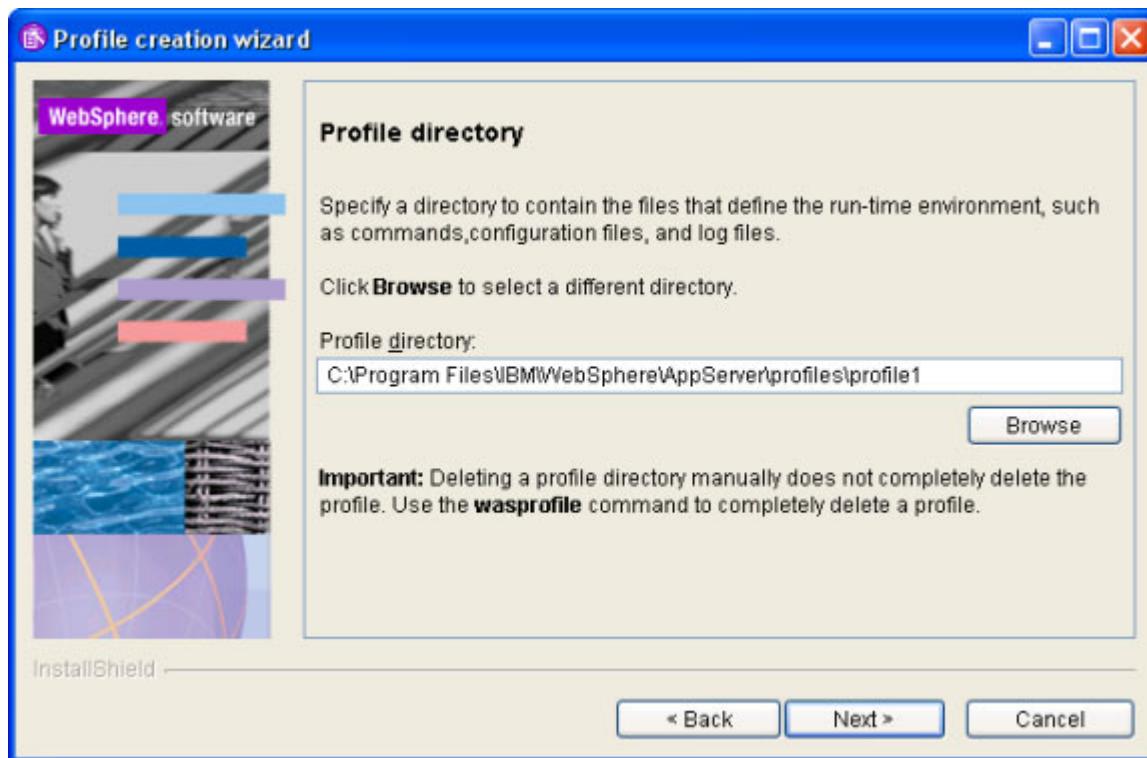
- b. Select the **Create an Application Server profile** option and click **Next**.



- __ c. Provide a unique name for a profile. Specify **profile1** for the profile name. Click **Next**.



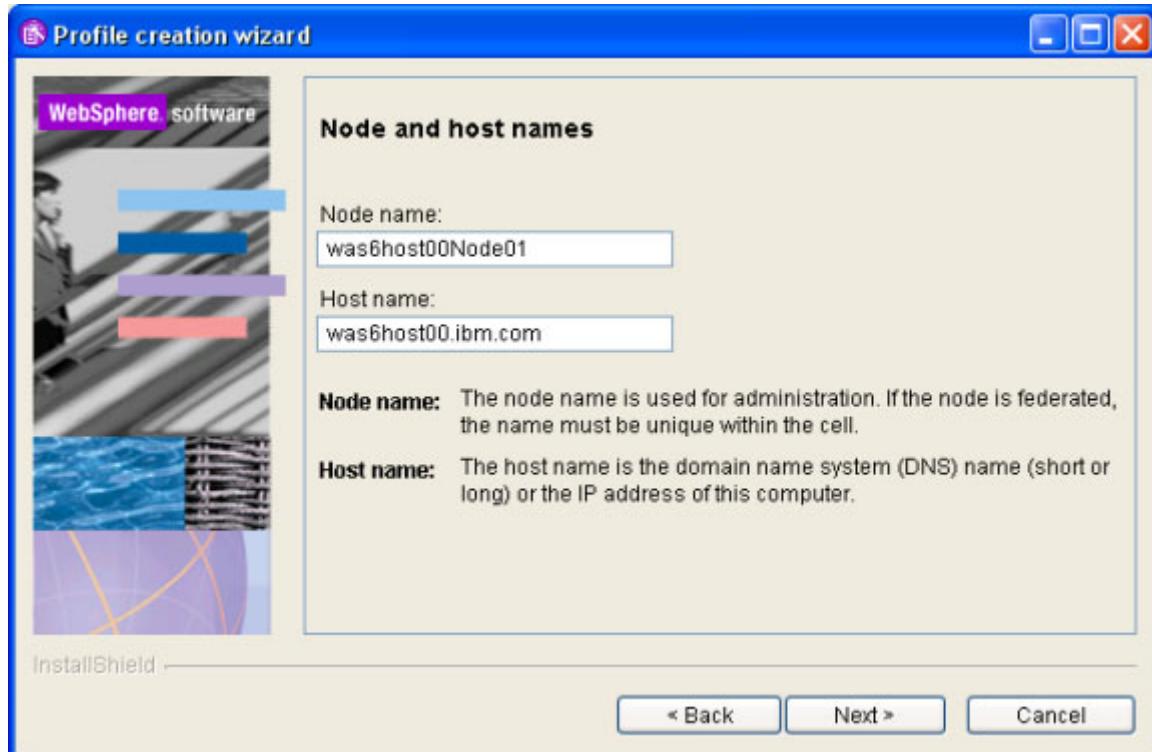
- ___ d. Specify a directory for the files for the profile. Use the default directory for **profile1**. Click **Next**.



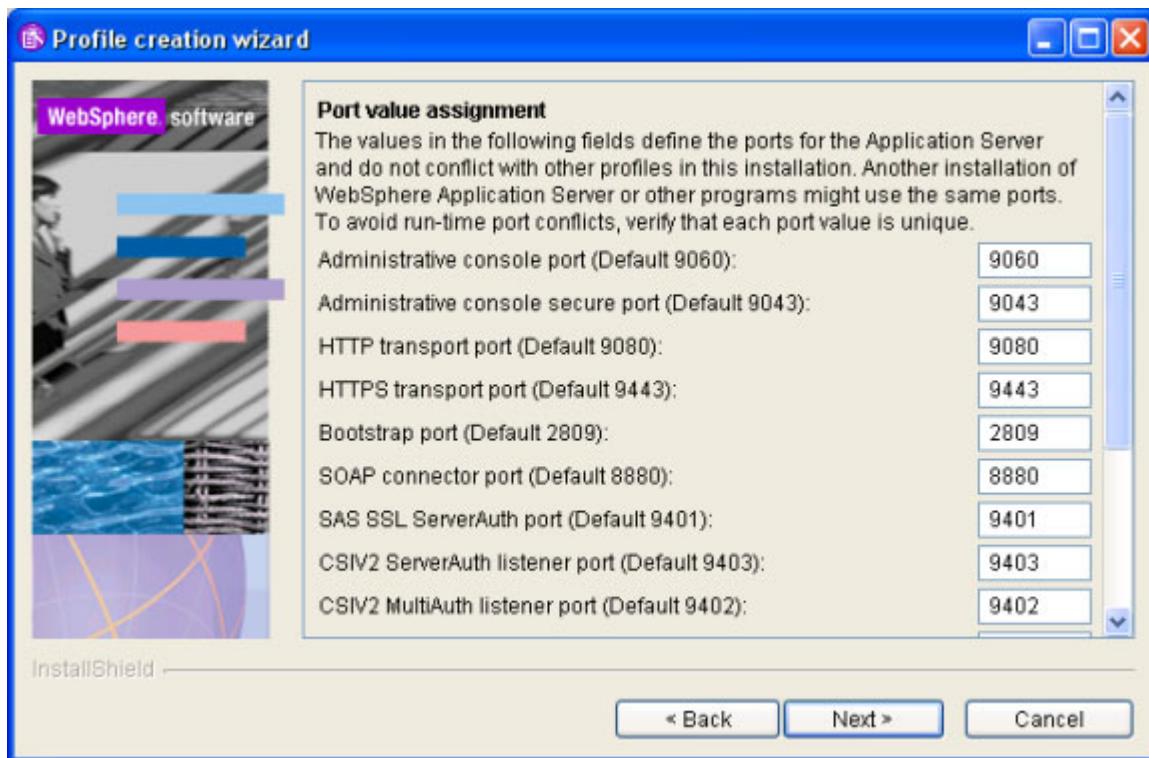
Information: It is a good idea to keep all profiles under the same directory structure under the **<profile_root>** folder.

UNIX: In UNIX based systems, the directory structures will be different. Refer to the appropriate **[Appendix]** for the appropriate directory structures. The install wizard should give the appropriate default values for these directories.

- ___ e. The next page lets you set the node name and TCP/IP host name. Default values will be filled in based on the detected TCP/IP host name for your machine. **Ensure the correct node name and host name are entered** - check with your instructor if you are unsure.



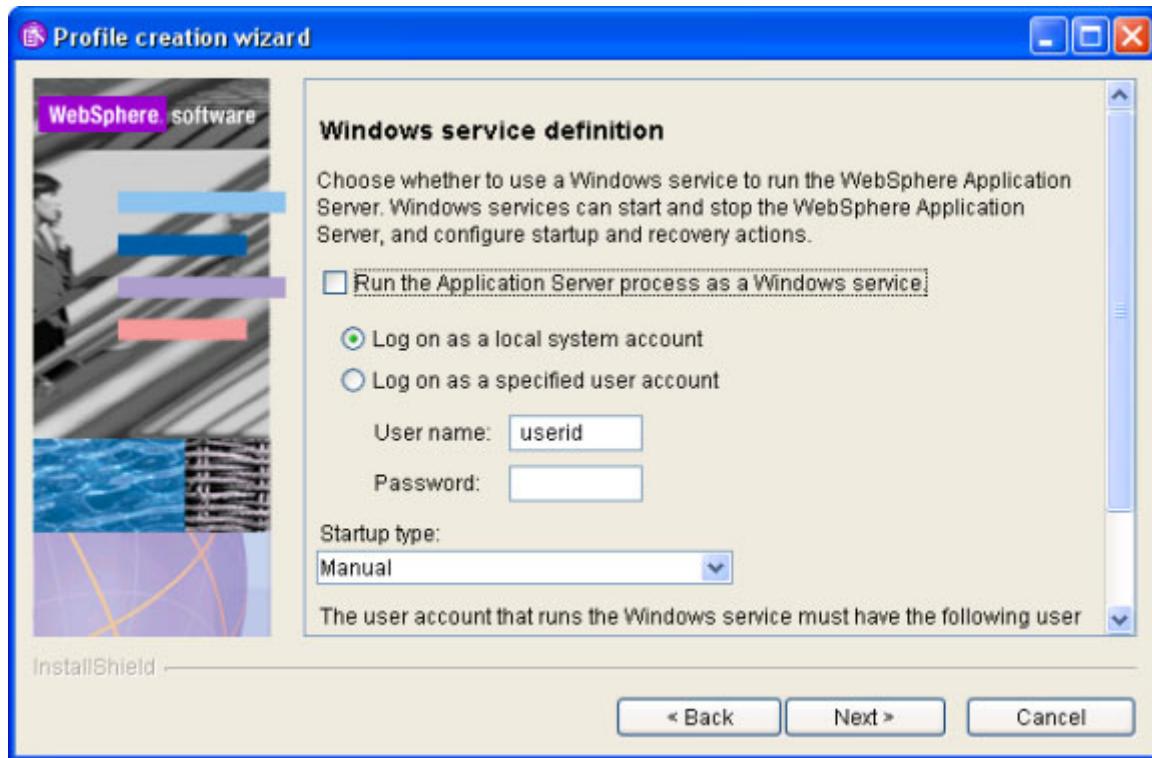
- ___ f. The next screen allows you to set any ports for the application server so there are no conflicts with other profiles. Keep all the defaults and click **Next**.



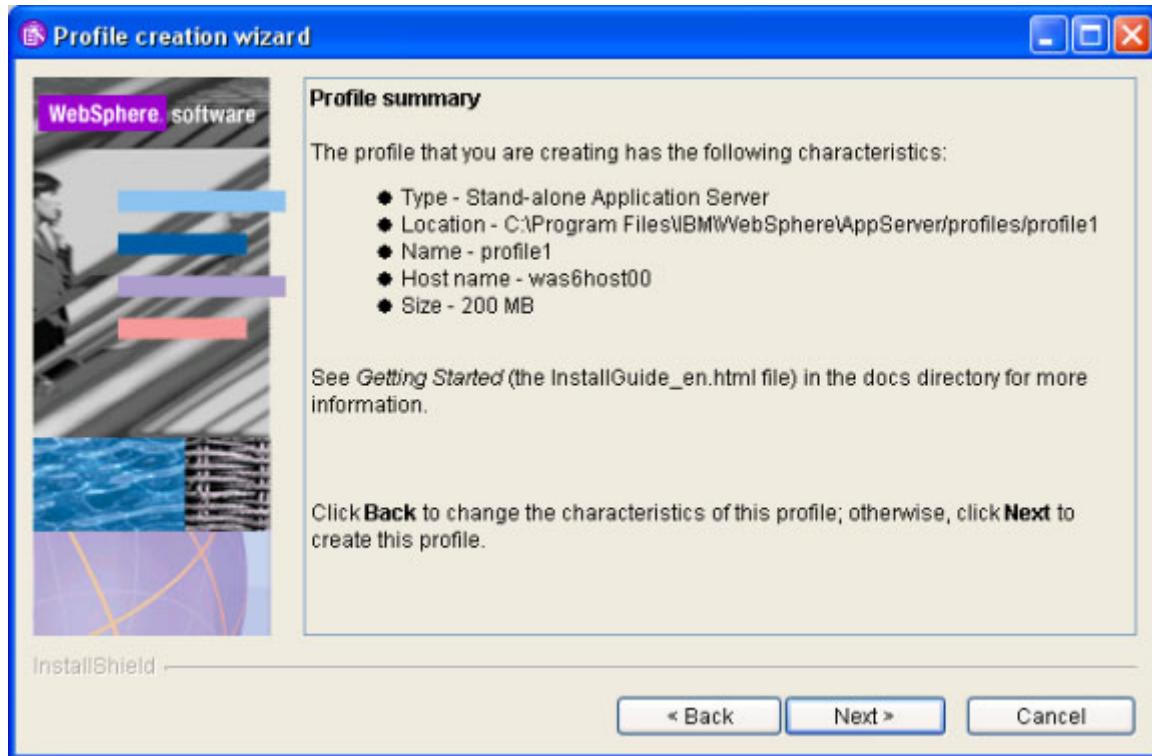
- ___ g. The next option is to select WebSphere Application Server to be run as a Windows service. If you do, you must specify the user it will run as.

UNIX: On UNIX based platforms, you will not have this option. This screen will not be available.

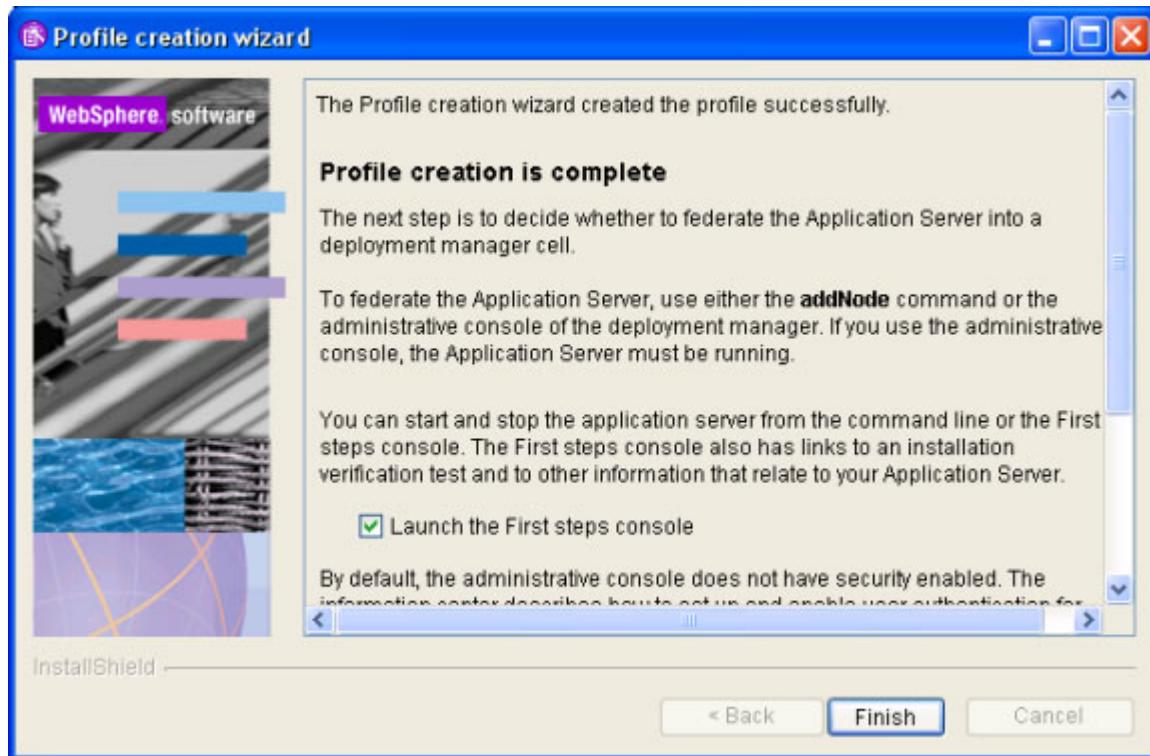
For this lab do not run the Windows service. Unselect the box for **Run the Application Server process as a Windows service**. Click **Next..**



- h. The Profile summary will appear with all of the choices you have made on previous screens. Double-check your information with what is shown below. Click **Next** to continue.



- ___ i. The profile creation is complete and **profile1** has been created. Notice that the **Launch the First steps console** box is selected. Click **Finish** and the First Steps program will launch.



Using First Steps

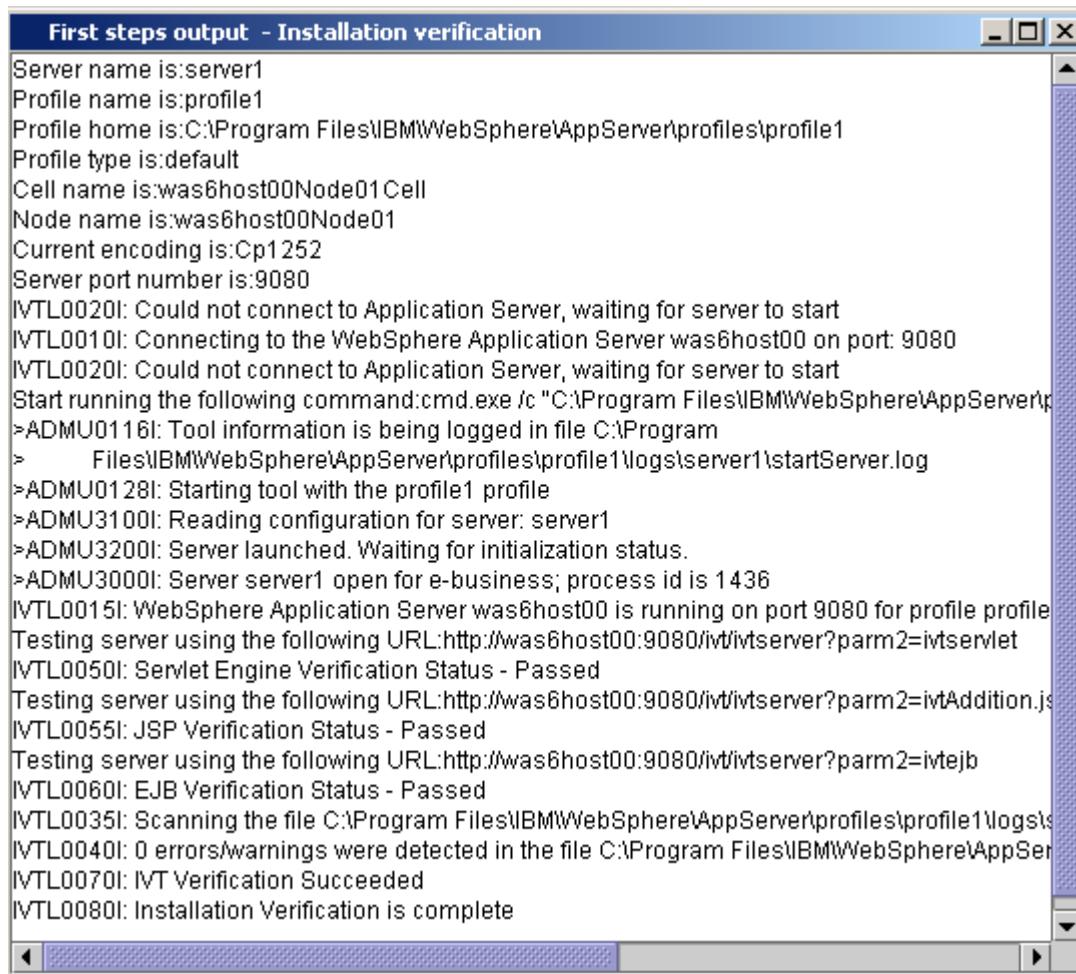
The First Steps program allows you to access the WebSphere InfoCenter (at the IBM Web site), start/stop the server and launch various tools. You can also migrate previous WebSphere Application Server versions to WebSphere Application Server V6.

- 1. The First steps console is associated with the application server profile, **profile1**, that was just created. Each profile has its own First steps console.



- 2. Click **Installation verification** from the First steps console.

- ___ a. The installation verification test tool will run and display messages to indicate its status. You should see message "IVTL0080I: Installation Verification is complete". Use the scroll bar to scroll down to the bottom to view all messages.

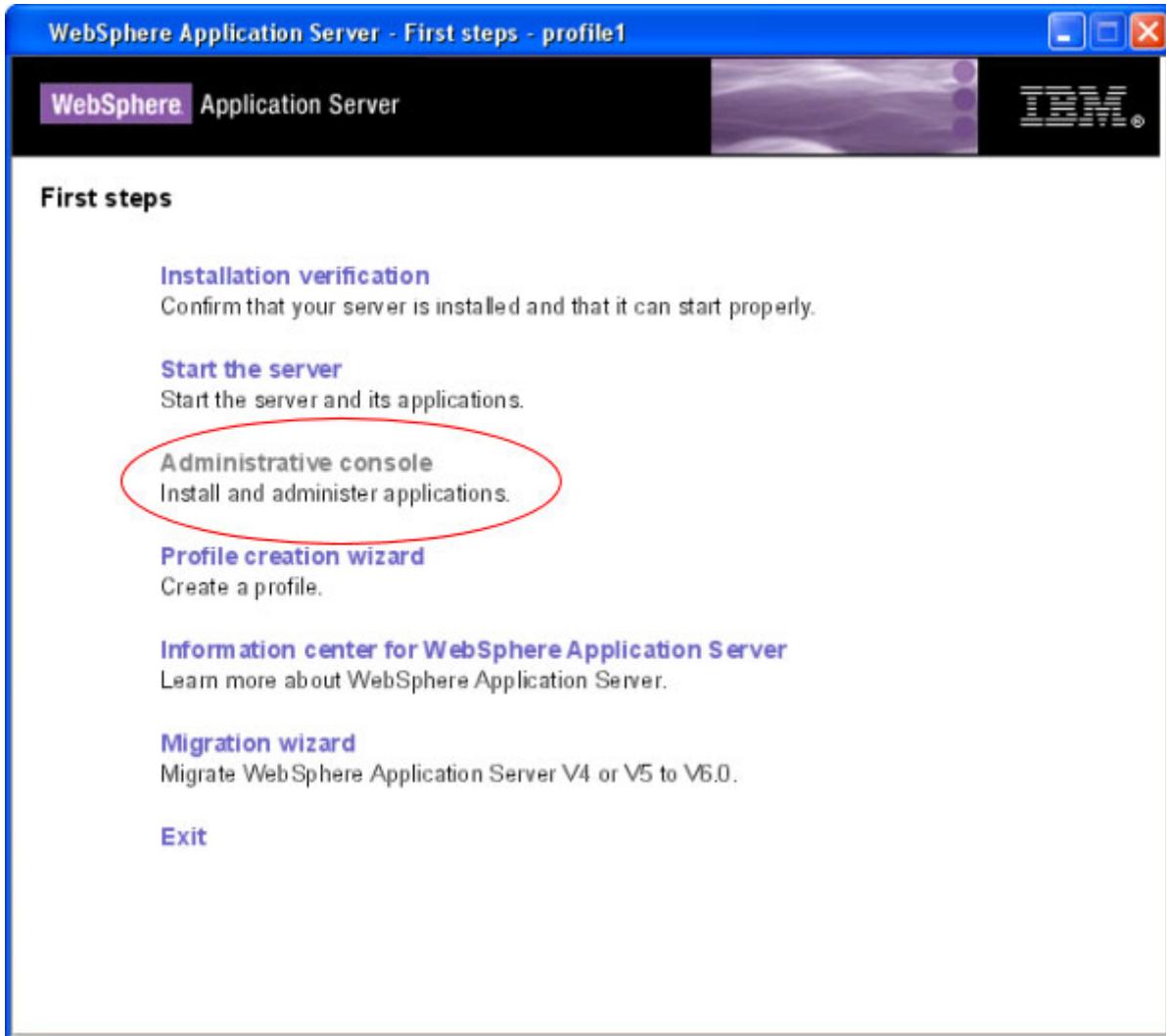


```

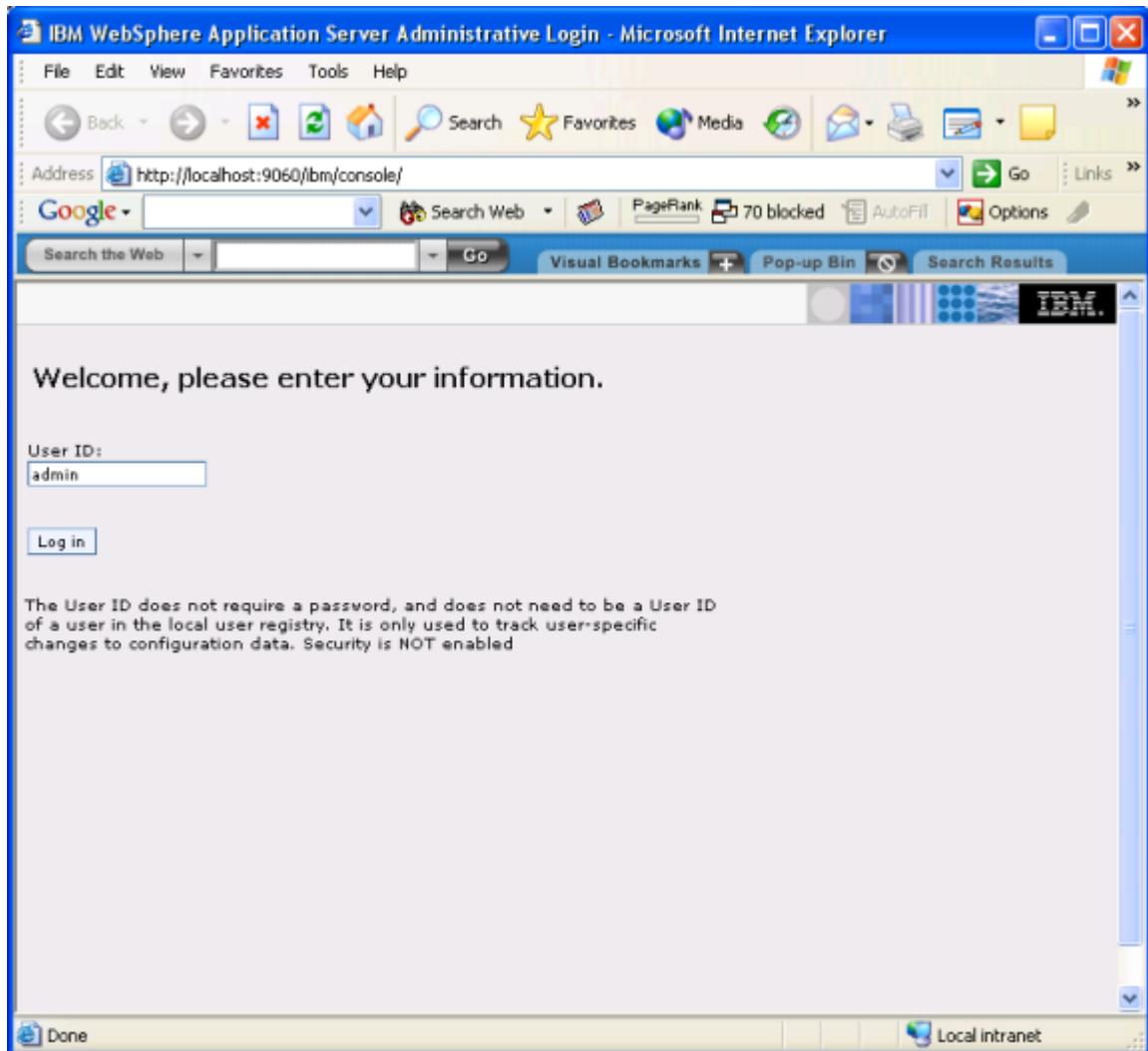
First steps output - Installation verification
Server name is:server1
Profile name is:profile1
Profile home is:C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1
Profile type is:default
Cell name is:was6host00Node01Cell
Node name is:was6host00Node01
Current encoding is:CP1252
Server port number is:9080
IVTL0020I: Could not connect to Application Server, waiting for server to start
IVTL0010I: Connecting to the WebSphere Application Server was6host00 on port: 9080
IVTL0020I: Could not connect to Application Server, waiting for server to start
Start running the following command:cmd.exe /c "C:\Program Files\IBM\WebSphere\AppServer\p>ADMU0116I: Tool information is being logged in file C:\Program
>      Files\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 1436
IVTL0015I: WebSphere Application Server was6host00 is running on port 9080 for profile profile
Testing server using the following URL:http://was6host00:9080/ivt/ivtserver?parm2=ivtservlet
IVTL0050I: Servlet Engine Verification Status - Passed
Testing server using the following URL:http://was6host00:9080/ivt/ivtserver?parm2=ivtAddition.js
IVTL0055I: JSP Verification Status - Passed
Testing server using the following URL:http://was6host00:9080/ivt/ivtserver?parm2=ivtejb
IVTL0060I: EJB Verification Status - Passed
IVTL0035I: Scanning the file C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\logs\server1\startServer.log
IVTL0040I: 0 errors/warnings were detected in the file C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\logs\server1\startServer.log
IVTL0070I: IVT Verification Succeeded
IVTL0080I: Installation Verification is complete
  
```

- ___ b. Close the First steps Installation verification output window.

- ___ c. From the First steps console select **Administrative console**. This will launch the administrative console for **profile1**.

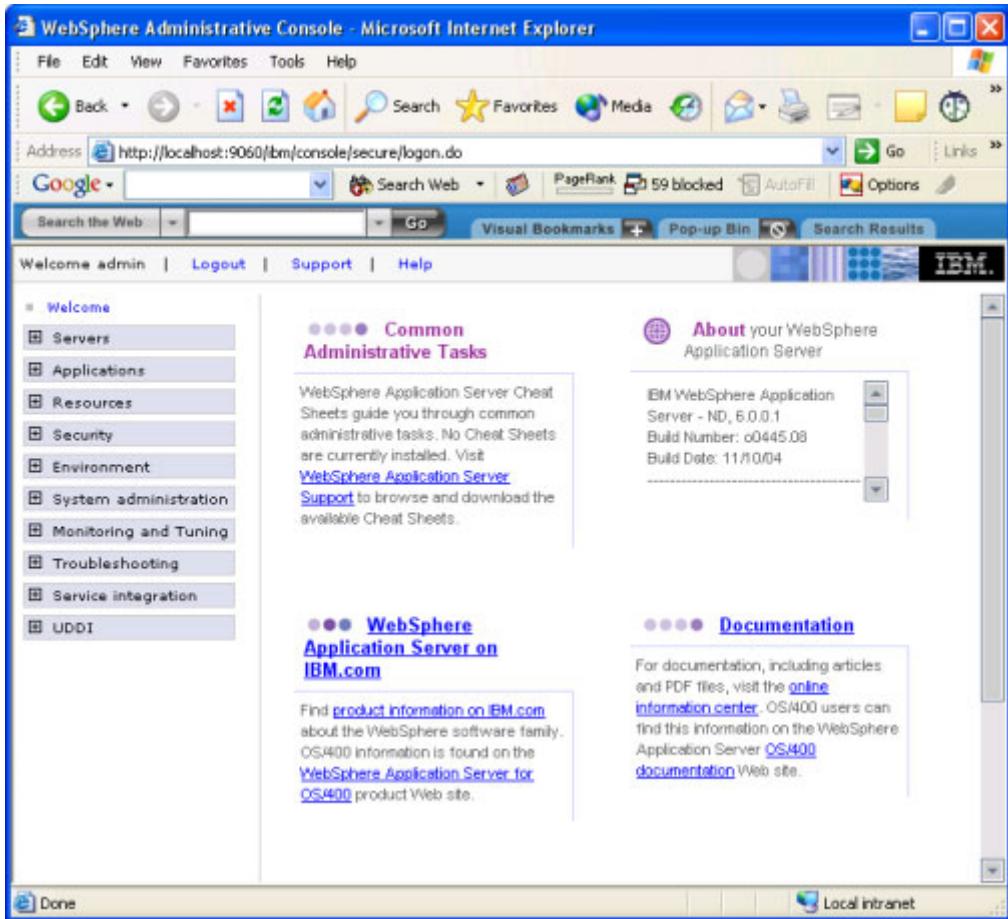


3. Log in to the administrative console as the userid **admin**.



Information: The userid **admin** is not a real user that has been defined in a registry. This userid will be used when logging into the administrative console. The userid specified during login is used to track configuration changes made by the user.

- ___ 4. The main console for the WebSphere administrative console should appear.



- ___ 5. Verify that the **DefaultApplication** has been installed and is running.

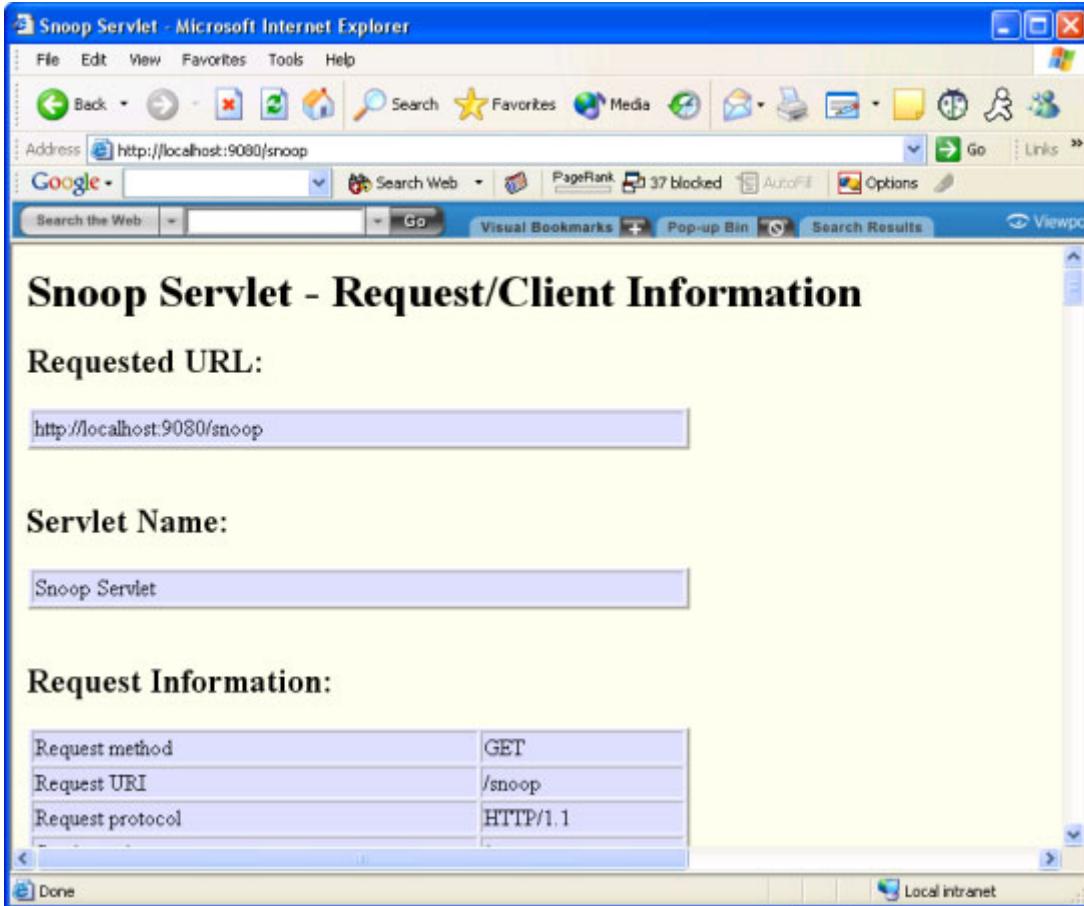
- __ a. Select **Applications --> Enterprise Applications**. You should see the DefaultApplication and it should be running.

The screenshot shows the WebSphere Administrative Console interface in Microsoft Internet Explorer. The left sidebar navigation menu includes Welcome, Servers, Applications (with Enterprise Applications and Install New Application options), Resources, Security, Environment, System administration, Monitoring and Tuning, Troubleshooting, Service integration, and UDDI. The main content area is titled "Enterprise Applications" and displays a table of installed applications. The table has columns for Select, Name, and Status. Three applications are listed: DefaultApplication, ivtApp, and query. The "DefaultApplication" row is circled in red. The "Status" column for DefaultApplication shows a green plus sign, indicating it is running. Below the table, it says "Total 3".

- __ b. Open another browser window and type in the following URL:

`http://localhost:9080/snoop`

This will execute a servlet called snoop, which comes with the DefaultApplication and will bring up a page with information about the runtime environment of the server.



This further confirms that the application server is operating correctly.

- ___ c. Close the browser.
- ___ d. Logout of the administrative console for now. It will be examined more in later exercises. Close the browser.
- ___ 6. Exit from the First steps console.
- ___ 7. Exit from the launchpad.

Create a backup of profile1 using command line tools

Before continuing create a backup of profile1 using the backupConfig command. The backupConfig command is a utility to back up the configuration of your profile to a file. You can later restore this configuration if needed. When the backupConfig commands runs it first stops the application server before creating the backup file.

- ___ 1. Run the backupConfig command.
 - ___ a. Open a command prompt window.

- ___ b. Change to the <profile_root>\profile1\bin folder.
- ___ c. Create the backup by entering the following command:
backupConfig

UNIX: Enter ./backupConfig.sh in the <profile_root>/profile1/bin directory.

```
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>backupConfig.bat
ADMU0116I: Tool information is being logged in file C:\Program
Files\IBM\WebSphere\AppServer\profiles\profile1\logs\backupConfig.log

ADMU0128I: Starting tool with the profile1 profile
ADMU5001I: Backing up config directory C:\Program
Files\IBM\WebSphere\AppServer/profiles/profile1/config to file
C:\Program
Files\IBM\WebSphere\AppServer\profiles\profile1\bin\WebSphereConfig_2
004-12-29.zip
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: server1
ADMU2010I: Stopping all server processes for node was6host00Node01
ADMU0510I: Server server1 is now STOPPED
-----
ADMU5002I: 150 files successfully backed up

C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>
```

Information: By default, all servers on the node stop before the backup is made so that partially synchronized information is not saved. The **-nostop** option tells the command not to stop the servers before backing up the configuration.

- ___ 2. The command will create a backup file called WebSphereConfig_<date>.zip using the current date and place it in the <profile_root>\profile1\bin directory. To distinguish between multiple backups that may be created modify the name using something more descriptive, such as the name of the profile.

- ___ a. Copy the **WebSphereConfig_<date>.zip** file from:

<profile_root>\profile1\bin

to:

<software_dir>\Backups

- __ b. Change the name of the backup file to **backup_profile1.zip**.

Information: If you need to restore the configuration directory structure at a later time you can use the **restoreConfig** command. You will need to specify the name of the backup file that was created after running the **backupConfig** command. The command restores the entire <profile_root>/<filename>/config directory.

```
C:\WINDOWS\System32\cmd.exe
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>restoreConfig.bat
C:\Software\Backups\backup_profile1.zip
ADMU0116I: Tool information is being logged in file C:\Program
Files\IBM\WebSphere\AppServer\profiles\profile1\logs\restoreConfig.lo
g
ADMU0128I: Starting tool with the profile1 profile
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: server1
ADMU2010I: Stopping all server processes for node was6host00Node01
ADMU0512I: Server server1 cannot be reached. It appears to be stopped.
ADMU5502I: The directory C:\Program
Files\IBM\WebSphere\AppServer\profiles\profile1\config already
exists; renaming to C:\Program
Files\IBM\WebSphere\AppServer\profiles\profile1\config.old
ADMU5504I: Restore location successfully renamed
ADMU5505I: Restoring file C:\Software\Backups\backup_profile1.zip to location
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\config
-----
ADMU5506I: 150 files successfully restored
ADMU6001I: Begin App Preparation -
ADMU6009I: Processing complete.

C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>
```

- __ 3. Verify the application server has stopped by entering the following command:

serverStatus server1

UNIX: Enter ./serverStatus.sh server1

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

Explore the WebSphere Application Server directory structure

Now that WebSphere Application Server is installed, walk through the directory structure and see what you just installed.

- ___ 1. Explore the WebSphere Application Server **profile1** directory.
 - ___ a. Bring up Windows Explorer.
 - ___ b. Navigate the tree to <**profile_root**>\profile1 (remember this subdirectory from one of the installation panels?)

UNIX:

- ___ a. Open a File Manager.
- ___ b. Navigate to <**profile_root**>/profile1.

- ___ 2. Look at the subdirectories and their contents:

- bin - programs, scripts, and DLLs
- config - configuration files
- etc - dummy keyring, keytab files, plug-in keys
- firststeps - firststeps utility
- installableApps - applications which may be installed
- installedApps - applications installed in WebSphere Application Server
- installedConnectors - installed resource adapters
- installedFilters - filters
- logs - trace and log files
- properties - configuration property files used by WebSphere
- temp - temporary area for files created during JSP processing
- tranlog - transaction log files
- wstemp - temporary area for events

Check 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. Change to the **logs** directory under the WebSphere Application Server installation directory.
 - ___ a. Examine **log.txt** in <**was_root**>\logs. This file records installation status.
- ___ 2. Change to the **wasprofile** directory under the WebSphere Application Server logs directory.
 - ___ a. Examine **wasprofile_create_profile1.log** in <**was_root**>\logs\wasprofile. This log records creation events that occur when creating the profile, profile1. Verify that profile1 creation was successful.
- ___ 3. Change to the **logs** directory under the WebSphere Application Server profile, **profile1**, installation directory.

- ___ a. Navigate to <profile_root>\profile1\logs.
- ___ b. Examine **pctLog.txt**. This log records installation events that occur when creating profiles with the Profile creation wizard.
- ___ c. Examine **backupConfig.log**. This log records events that occur when creating a backup of the configuration directory structure.
- ___ d. Examine **ivt_config.log**. This logs the messages during installation of the installation verification test application. Verify that no errors occurred during installation.
- ___ e. Examine **ivtClient.log**. This logs results from the installation verification command.
- ___ f. Examine **portdef.props**. This properties file logs information about the default ports for an application server.

Start the WebSphere Application Server

Verify the installation by starting the WebSphere Application Server profile1 and looking at the log files.

- ___ 1. A log file records the startup messages from the server. This log file can be found in the <profile_root>\profile1\logs\server1 directory in the WebSphere Application Server **profile1** installation directory) and is named **startServer.log**. Examine the contents of the **startServer.log** file.

Information: A copy of the UNIX utility **tail** is included in the lab machine (tail is not part of the standard Windows distribution). The tail utility makes it easier to monitor what is being written to a text based log file. (The **tail** utility can be copied from your <software_dir>\gnu_utils directory, or you can obtain a copy from <http://sources.redhat.com/cygwin/>.)

- ___ a. Navigate to <profile_root>\profile1\logs\server1.
 - ___ b. Enter: **tail startServer.log**. This command will show the last few lines of the file **startServer.log**, where you will see the startup messages from the server.
You can also use **tail -f startServer.log** which will continuously update your screen when new messages are added to the file. To exit the tail utility, press **Ctrl-C**.
 - ___ c. Another approach is to view the **startServer.log** using **notepad** or an another text editor. This approach is not as convenient because a text editor will only show the messages written to the file at a point in time; whereas, the **tail -f** command will display the file contents continuously.
- ___ 2. Delete all the log files in <profile_root>\profile1\logs\server1. When the server is started new log files will be created. This will provide fresh log files.
 - ___ 3. Start the WebSphere Application Server **profile1** if it is not already started.

- __ a. In a command prompt navigate to the <profile_root>\profile1\bin folder and enter:

```
startServer server1
```

UNIX: In a UNIX environment, processes will have to be checked and started using Terminal commands.

- __ a. Navigate to <profile_root>/profile1/bin and enter ./startServer.sh server1 to start the application server.

- __ b. Verify the application server has started by entering the following command:

```
serverStatus server1
```

UNIX: Enter ./serverStatus.sh server1

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

- __ 4. After successful startup of the server, at the bottom of the startServer.log, you should see the message “**Server server1 open for e-business**”, which indicates the server is ready.
- __ 5. Examine the **SystemErr.log** file (in the same directory). This contains the standard error output from the Java virtual machine (JVM) running the application server. This file should be empty if the server has started correctly.
- __ 6. Examine the **SystemOut.log** file (in the same directory). This contains the standard output from the Java virtual machine (JVM) running the application server. This file should contain a lot more detailed messages indicating the steps performed during startup of the server. These steps include security initialization, messaging initialization, registering resources in the JNDI namespace, EJB initialization, Web module initialization and HTTP transport initialization. This file also contains messages from application System.out print line code.
- __ 7. Minimize this command prompt window. You may want to refer to the messages in the **SystemOut.log** or **SystemErr.log** again as you use the server and perform more functions.
- __ 8. Stop the WebSphere Application Server profile1.
 - __ a. Navigate to the <profile_root>\profile1\bin folder and enter:

```
stopServer server1
```

- __ b. Verify the application server has stopped by entering the following command:

```
serverStatus server1
```

UNIX:

- a. Navigate to <profile_root>/profile1/bin.
- b. Enter ./**stopServer.sh** server1
- c. Enter ./**serverStatus.sh** server1

End of lab

Exercise review and wrap-up

The first part of the exercise looked at the installation of WebSphere Application Server V6. The Profile creation wizard was also used to create an application server profile called **profile1**.

Exercise 2. WebSphere Information Center installation (optional)

What this exercise is about

In this exercise, you will install the WebSphere Application Server Information Center Help System.

What you should be able to do

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

- Install the WebSphere Application Server V6 Information Center
- Navigate and search the Information Center documentation
- Learn about bookmarks
- Synchronize help with the navigation tree
- Search the system

Introduction

In this optional exercise, you will install the Help System. The IBM WebSphere Help System is a viewer for displaying product or application information developed as Eclipse documentation plug-ins. The system provides a graphical user interface for browsing and searching online documentation.

If you have installed an IBM product whose documentation is provided in the form of Eclipse plug-ins, the Help System is probably already installed, in which case you do not need to do this task. To confirm, search your file system for a directory called `eclipse\plugins`.

The Help System ZIP file has already been downloaded and has been stored in the software directory.

Required materials

To perform this exercise, you must have the zipped version of the Help System.

Instructor exercise overview

In this exercise the students will learn how to install and use the downloadable Help System.

No changes to WebSphere lab system are required for subsequent labs.

Exercise instructions

Install the Help System

Begin by unzipping the file that has been downloaded.

- ___ 1. Using Windows Explorer, navigate to <software_cds>\HelpSystem and examine the contents of this folder. Look for the file **websphere_help_system_212_win.zip**.

UNIX:

- ___ 1. Navigate to <software_cds>/HelpSystem.

This file was downloaded from the IBM WebSphere Web site

<http://www.ibm.com/software/webservers/appserv/library/infocenter.html>.

- ___ 2. Unzip file **websphere_help_system_212_win.zip** to the directory <software_cds>\HelpSystem.

- ___ a. Right-click and select **WinZip** and **Extract to...**

WinZip will open and prompt you for the location to extract the files to. If you do not have WinZip installed, you may need to double-click the file to open your unzip application or you may need to use a command line unzip utility to unzip the files.

- ___ b. Enter directory <software_cds>\HelpSystem and click **Extract**.

WinZip will extract the files. Close WinZip.

UNIX:

- ___ a. Expand the appropriate zip file to <software_cds>/HelpSystem. For example, on Linux use **unzip**
websphere_help_system_212_linux.zip -d
<software_cds>/HelpSystem

- ___ 3. Explore the <software>\WebSphere Help System directory. There should be an **installing_and_managing.htm** file and some bat files. There will also be two subdirectories **eclipse** and **license**.

UNIX: Explore the <software_cds>/HelpSystem/ibm_help directory.

Start the Help System

The Help System uses a browser to view its contents; however, you do not have to be connected to a network because all the files will be loaded from your local machine.

- ___ 1. Start the Help System.
- ___ a. Open the **WebSphere Help System** folder.

- __ b. Double-click the **WebSphere help start.bat** file.

UNIX:

- __ a. Navigate to the **ibm_help** directory.
__ b. Run the **./help_start.sh&** file.

If you get the following error:

```
java.lang.Exception: Timeout waiting for file eclipse/workspace/.metadata/.connection  
Eclipse is not running.
```

Edit the **help_start.sh** file and make it look like this:

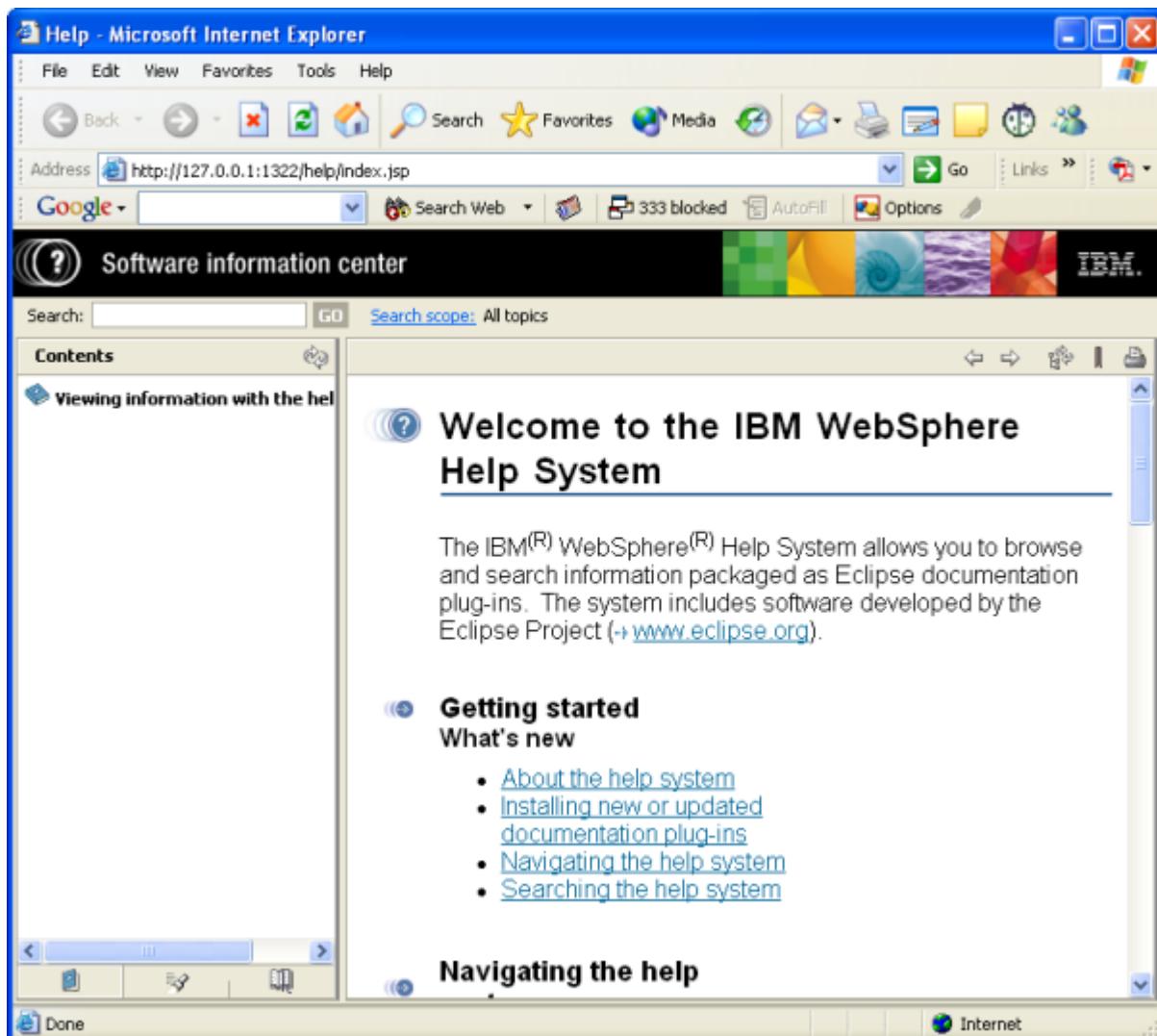
```
#!/bin/sh  
/opt/IBM/WebSphere/AppServer/java/bin/java -classpath  
eclipse/plugins/org.eclipse.help_2.1.2/help.jar  
org.eclipse.help.standalone.Help -eclipsehome eclipse -command displayHelp
```

This edit removes the path to the java command and the -Xj9 from the command line. It removes: `eclipse/jre/bin/java -Xj9`

This same change needs to be done to the **help_end.sh** file. After making the changes to the file start the help system.

- __ 2. The Help System will automatically appear. The Help System is all static HTML files and will work with either Internet Explorer or Netscape Communicator.

- ___ 3. The Help System Welcome page should be displayed.



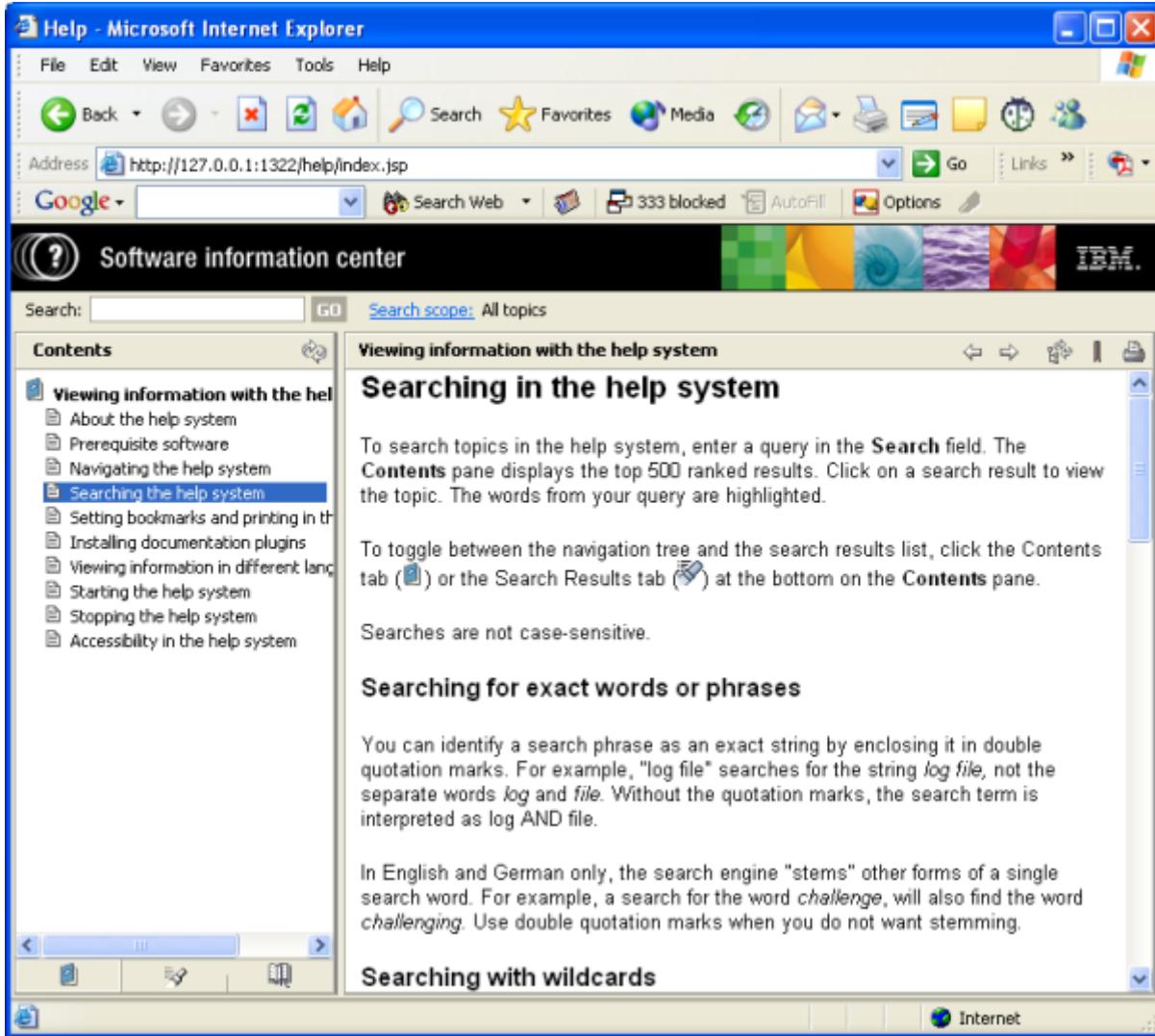
- ___ 4. In case you want to refer to this later during class, add this address to your "favorites" if using the Internet Explorer browser (or "bookmark" if using Netscape).
___ 5. Explore the documentation. Use your browser's Back button to return to the previous page.

Searching the Help System

The Help System includes a search facility that can be used. You can also find out more information about the search facility.

- 1. On the left is a navigation frame containing a hierarchical view of topics. In the left pane double-click **Viewing information with the help system** to expand this folder for the view of topics.

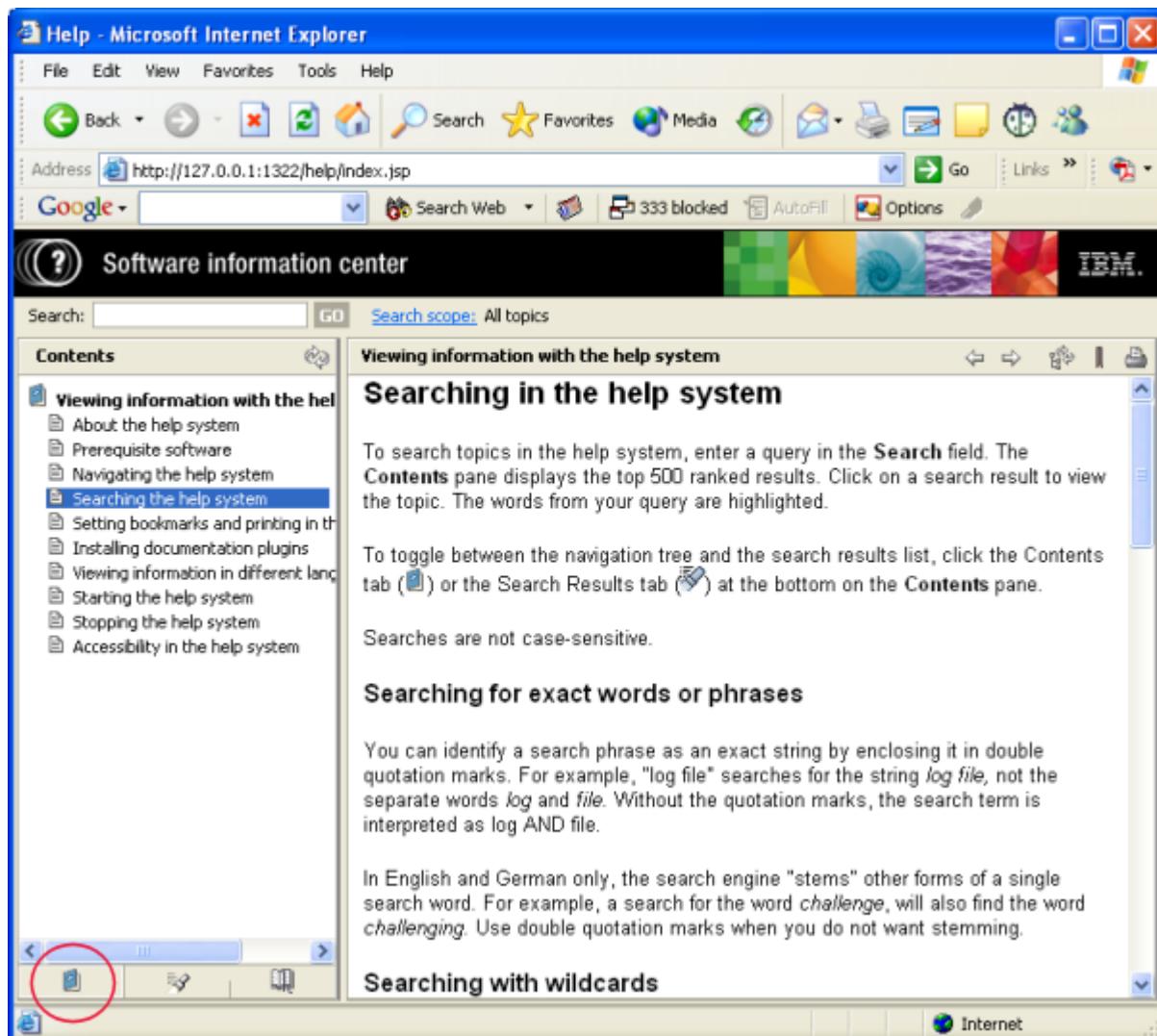
A number of topics will be listed. Select **Searching in the Help System**. The window should look like this:



- 2. Select the other options in the left pane to get more information on using the Help System.

On the right is the main frame showing the current selected document.

- ___ 3. To return to the table of contents after running a search, click the **Contents** tab at the bottom of the navigation pane.



Stopping the Help System

When closing the help system browser, the help system processes continue to run in the background. You must stop these help system processes when you install documentation plug-ins or update the help system with new plug-ins.

- ___ 1. Stop the Help System.
___ a. Open the **WebSphere Help System** folder.

- ___ b. Double-click the **WebSphere help end.bat** file.

UNIX:

- ___ a. Navigate to the **ibm_help** directory.
 ___ b. Run the **./help_end.sh&** file.

- ___ 2. The Help System will still appear. Exit out of the Help System.

Installing and using documentation plug-ins

The help system works with any information that has been packaged as an Eclipse documentation plug-in, including IBM product documentation plug-ins. IBM documentation plug-ins are easily identified because they use a common directory-naming convention (com.ibm.xxx.doc).

- ___ 1. Navigate to **<software_cds>\HelpSystem**.

UNIX:

- ___ 1. Navigate to **<software_cds>/HelpSystem**.

- ___ 2. Unzip the files **com.ibm.websphere.nd.doc.zip** and **com.ibm.websphere.ihs.doc.zip**.

- ___ a. Right-click and select **WinZip** and **Extract to...**

WinZip will open and prompt you for the location to extract the files to. If you do not have WinZip installed, you may need to double-click the file to open your unzip application or you may need to use a command line unzip utility to unzip the files.

- ___ b. Enter directory **<software_cds>\HelpSystem\WebSphere Help System\clipse\plugins** and click **Extract** for each of the zip files.

WinZip will extract the files. Close WinZip.

UNIX:

- ___ a. Expand the appropriate zip file to
<software_cds>/HelpSystem/ibm_help/eclipse/plugins directory.

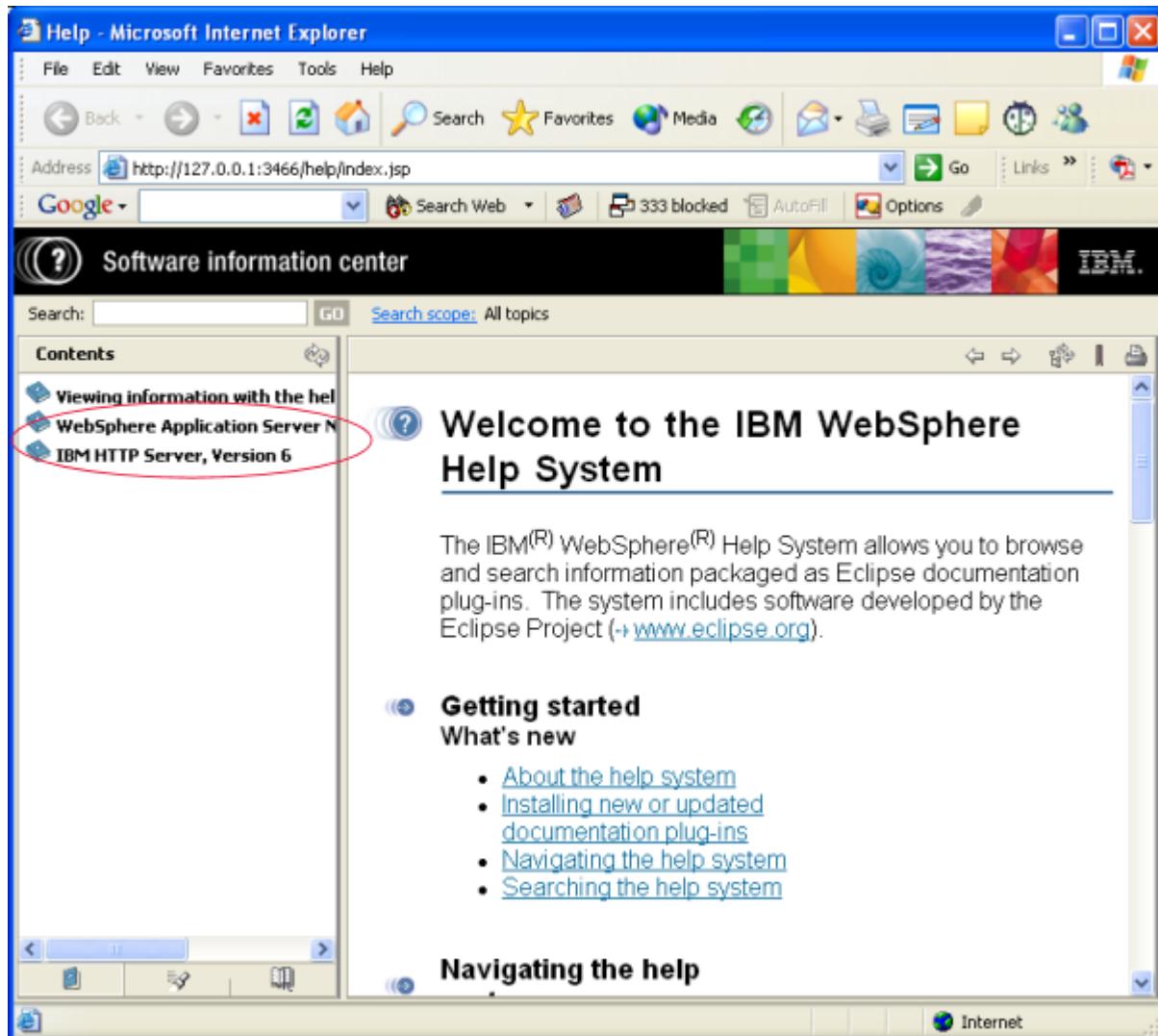
- ___ 3. Start the WebSphere Help System.

- ___ a. Double-click the **WebSphere help start.bat** file in the **WebSphere Help System** folder.

UNIX:

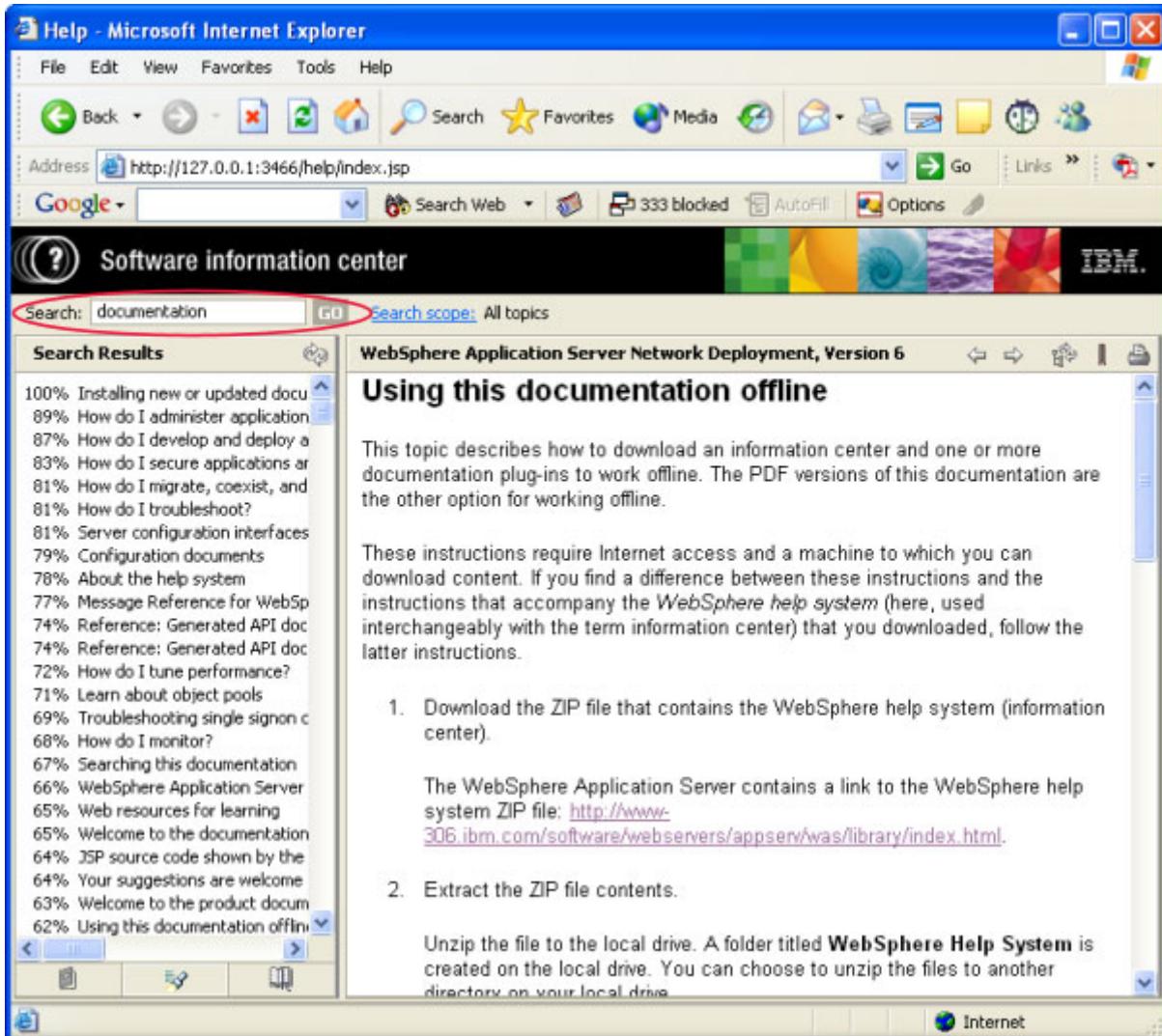
- ___ a. Run the **./help_start.sh&** file.

Notice in the Navigation frame the documentation is available for both WebSphere Application Server Network Deployment and IBM HTTP Server.



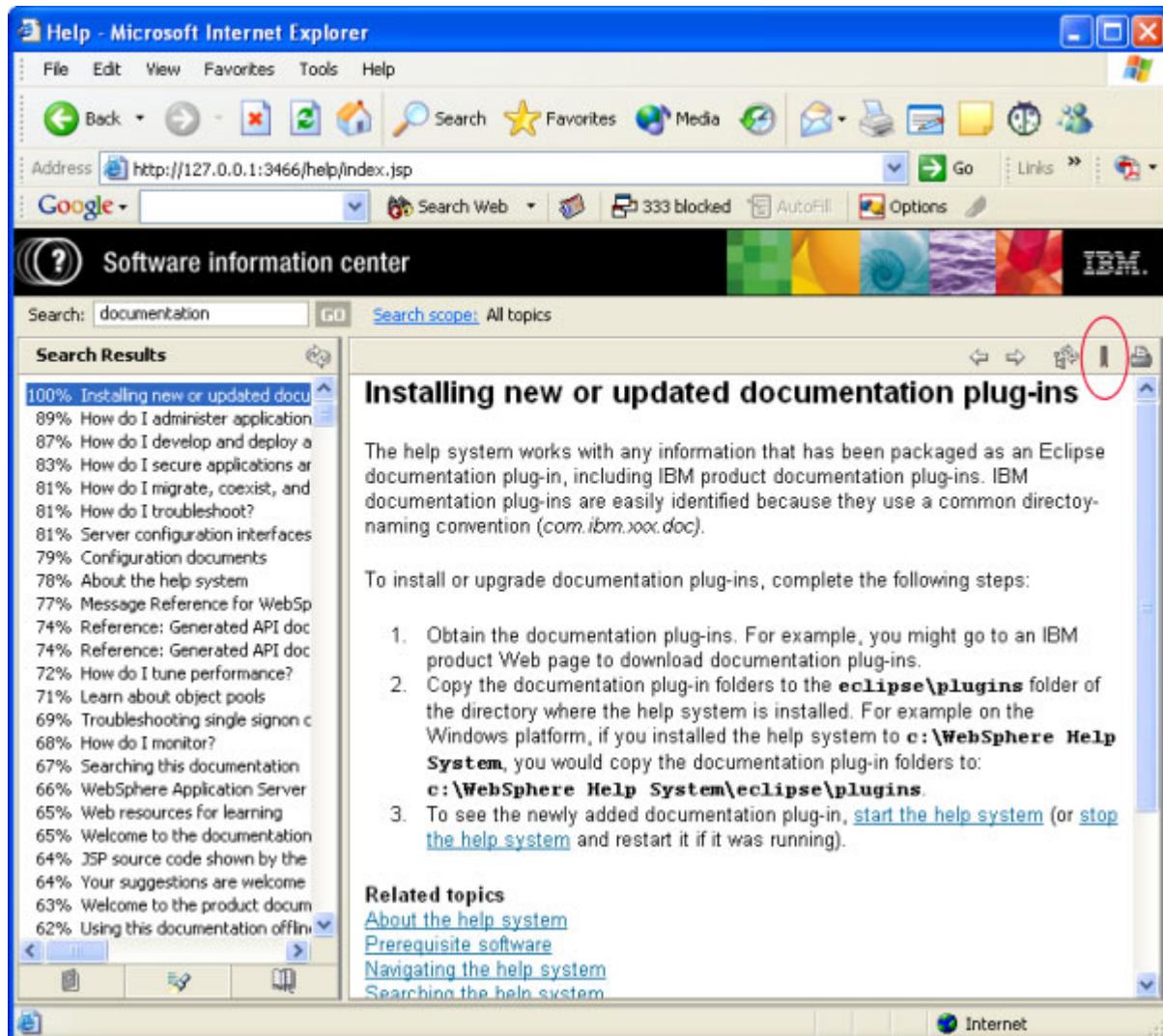
- ___ 4. Expand **WebSphere Application Server Network Deployment** and view details of the product documentation table of contents.
- ___ 5. Search the information center for information on documentation.

- __ a. In the search area specify **documentation**. A list of search results will be displayed.

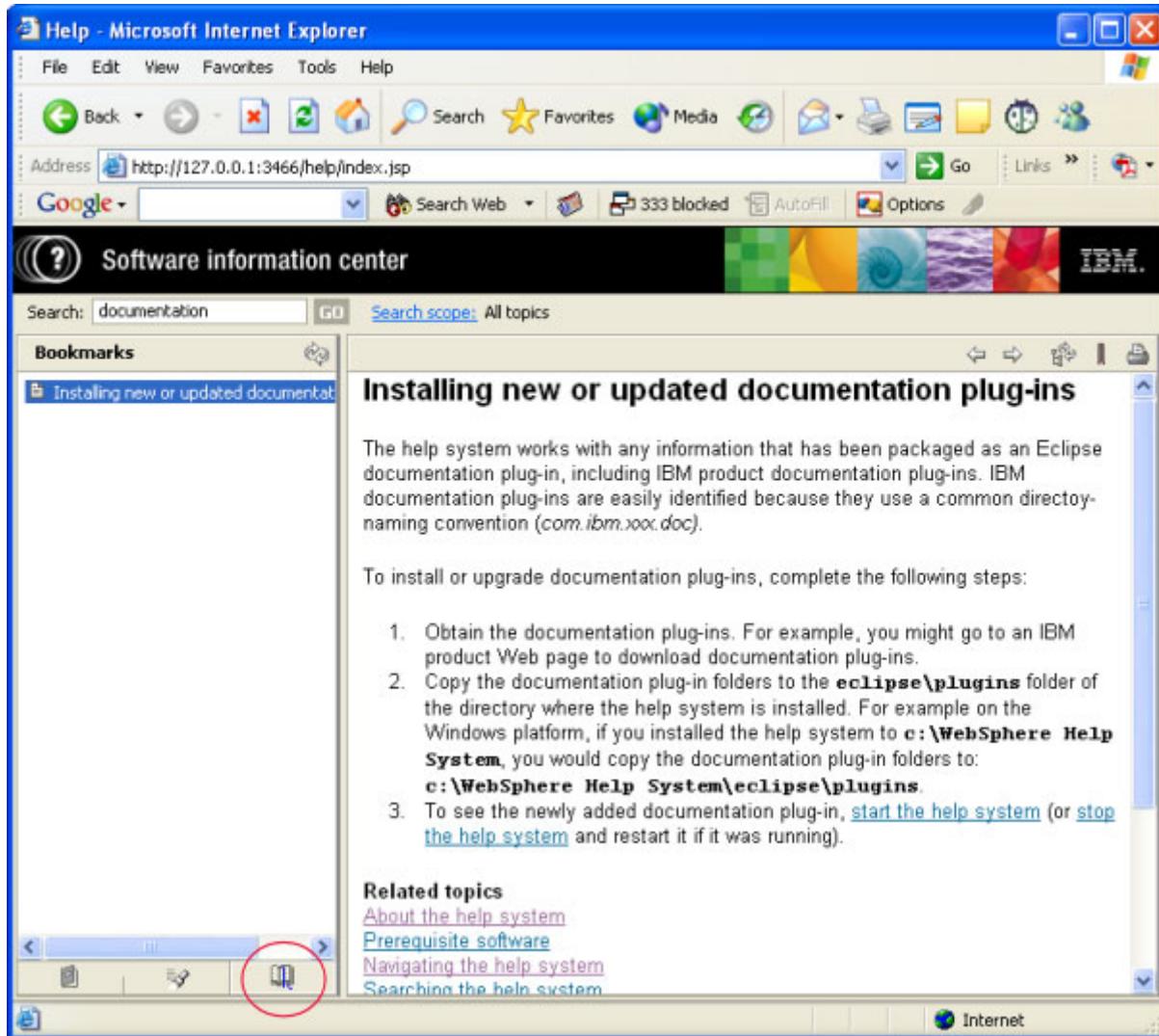


- __ 6. Select one of the search items and bookmark for later use.
- __ a. Select a search result from the list. In the right pane information on this item will appear.

- __ b. From the tool bar in the right pane in the Help System select the **Bookmark** icon.



- ___ c. Select the **Bookmark** tab at the bottom of the Contents pane. This will return a listing of your bookmarks.



- ___ 7. Continue to explore the Help System and Information Center documentation plug-ins.
- ___ 8. Stop the Help System.
- ___ a. Open the **WebSphere Help System** folder.
- ___ b. Double-click the **WebSphere help end.bat** file.

UNIX:

- ___ a. Navigate to the **ibm_help** directory.
- ___ b. Run the **./help_end.sh&** file.
- ___ 9. The Help System will still appear. Exit out of the Help System.

End of lab

Exercise review and wrap-up

This exercise installed the offline Help System documentation and demonstrated some navigation techniques.

Exercise 3. Explore WebSphere Application Server through the administrative console

What this exercise is about

In this exercise, explore WebSphere Application Server - Network Deployment v6 using the administrative console. You examine the configuration structure, and navigate within the WebSphere administrative console, which is used to perform server configuration, to gain knowledge and become familiar with what is available.

What you should be able to do

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

- Navigate within the WebSphere administrative console
- Configure resources using the WebSphere administrative console

Introduction

In this exercise, explore the WebSphere Application Server V6 configuration. Start the server and browse through some configuration files. Then start and navigate around the WebSphere administrative console.

The application server should already be installed and tested, and you should 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 perform configuration tasks.

Required materials

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

Instructor exercise overview

In this exercise the students learn how to navigate within the WebSphere Application Server V6 environment, using the WebSphere administrative console.

The first step is to log in to the system. Use the userid **userid**, and password **was1edu** for Windows systems.

The students learn to start the server and examine the configuration.

Then they learn how to start the WebSphere administrative console.

They then explore and learn how to navigate within the WebSphere administrative console, and perform functions such as:

- Examine the properties of a server, application, virtual host, JDBC Provider.
- Correctly log out of the WebSphere administrative console so that changes can be saved or discarded.

The instructor should become familiar with many of the functions within the WebSphere administrative console. Students are sure to discover and have questions beyond the scope of this lab.

Following Exercise 3. There are no changes to the system.

Exercise instructions

Start the server

Before you can configure the application server environment, you must start all the required processes. The server was not installed as a Windows service. For this exercise you will use profile1's server1.

- ___ 1. Determine how many JVMs are running on your system.
 - ___ a. Press **Ctrl-Alt-Delete** and click the **Task Manager** button.
 - ___ b. Select the **Processes** tab and you see all the processes which are currently running. Sort by name.
 - ___ c. Take note of how many **java.exe** programs are running. There should be none, unless you left the server running in an earlier exercise.
 - ___ d. Click the **Mem Usage** column heading to sort by memory usage. This ensures the larger processes (such as JVMs) are listed at the top.
 - ___ e. Now, minimize the Task Manager until later, when you look at it again after you start your server.

UNIX:

- ___ a. Use a Terminal Window.
- ___ b. Look for java processes running by typing **ps-ef | grep java** at the command prompt.
- ___ a. Take note of how many **java** processes are running and where they were started. There should be none whose description ends in **server1**, unless you left the server running in an earlier exercise.

- ___ 2. Start the server.

- ___ a. In a command prompt navigate to **<profile_root>\profile1\bin**.
- ___ b. Issue the **startServer server1** command to start the server.
- ___ c. After successful startup of the server you should see the message "**Server server1 open for e-business**", which indicates the server is ready.

UNIX:

- ___ a. Navigate to **<profile_root>/profile1/bin**.
- ___ b. Enter **./startServer.sh server1**

If you have problems starting the server, look at the **startServer.log** using the supplied tail utility.

At a command prompt window, change to the directory **<profile_root>\profile1\logs\server1** and enter command: **tail startServer.log**

This shows the startup messages from the server. You can also use **tail -f startServer.log** which continuously updates your screen when new messages are added to the startServer.log.

Another approach is to view the startServer.log using a text editor. This approach is not convenient because an editor only shows the messages written to the file at a point in time; whereas, the **tail -f** command displays the file contents continuously.

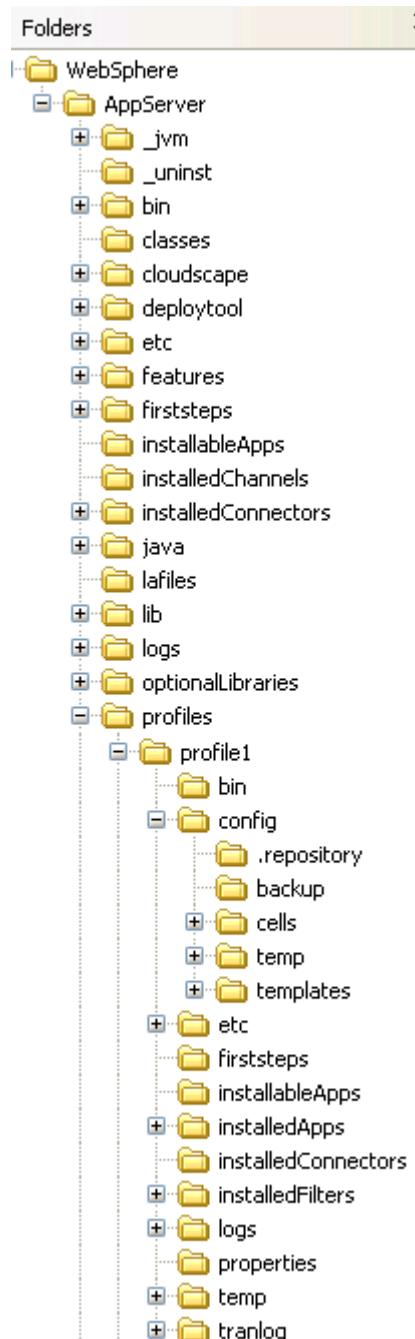
- ___ 3. Restore the Task Manager window and you see that some additional processes have been started. One of those is a JVM (java.exe) for the application server, server1.

Explore configuration files

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

- ___ 1. Explore the config directory structure and view some of the configuration files.

__ a. Navigate to <profile_root>\profile1\config.



__ b. A few important files are (where <cell> is the cell name, <node> is the node name, and <server> is the server name):

- config\cells\<cell>\resources.xml - defines cell-wide resources
- config\cells\<cell>\nodes\<node>\resources.xml - defines node-specific resources
- config\cells\<cell>\nodes\<node>\servers\<server>\resources.xml - defines server-specific resources such as JDBC and JMS providers
- config\cells\<cell>\security.xml - defines global security settings

- config\cells\<cell>\virtualhosts.xml - defines virtual hosts
- config\cells\<cell>\nodes\<node>\serverindex.xml - defines applications and endpoints for a node
- config\cells\<cell>\nodes\<node>\servers\<server>\server.xml - defines the configuration of a server

Information: You should not edit these XML files manually, instead the administrative console or wsadmin command line tool should be used to make configuration changes that affect these files.

- ___ 2. Examine the Secure Association Services (SAS) client configuration file <profile_root>\profile1\properties\sas.client.props.

- ___ a. Navigate to: <profile_root>\profile1\properties
- ___ b. Open **sas.client.props** in an editor.

This file contains security configuration information used by clients such as wsadmin to authenticate with the security service. Important parameters are:

- loginSource - specifies how the authentication information is obtained. The default is prompt which means the user is prompted for a user ID and password. This can be changed to **properties** to stop the user from being prompted.
- loginUserId and loginPassword - The ID and password must be specified using these parameters if loginSource=properties.

- ___ 3. Close any editor windows still open.

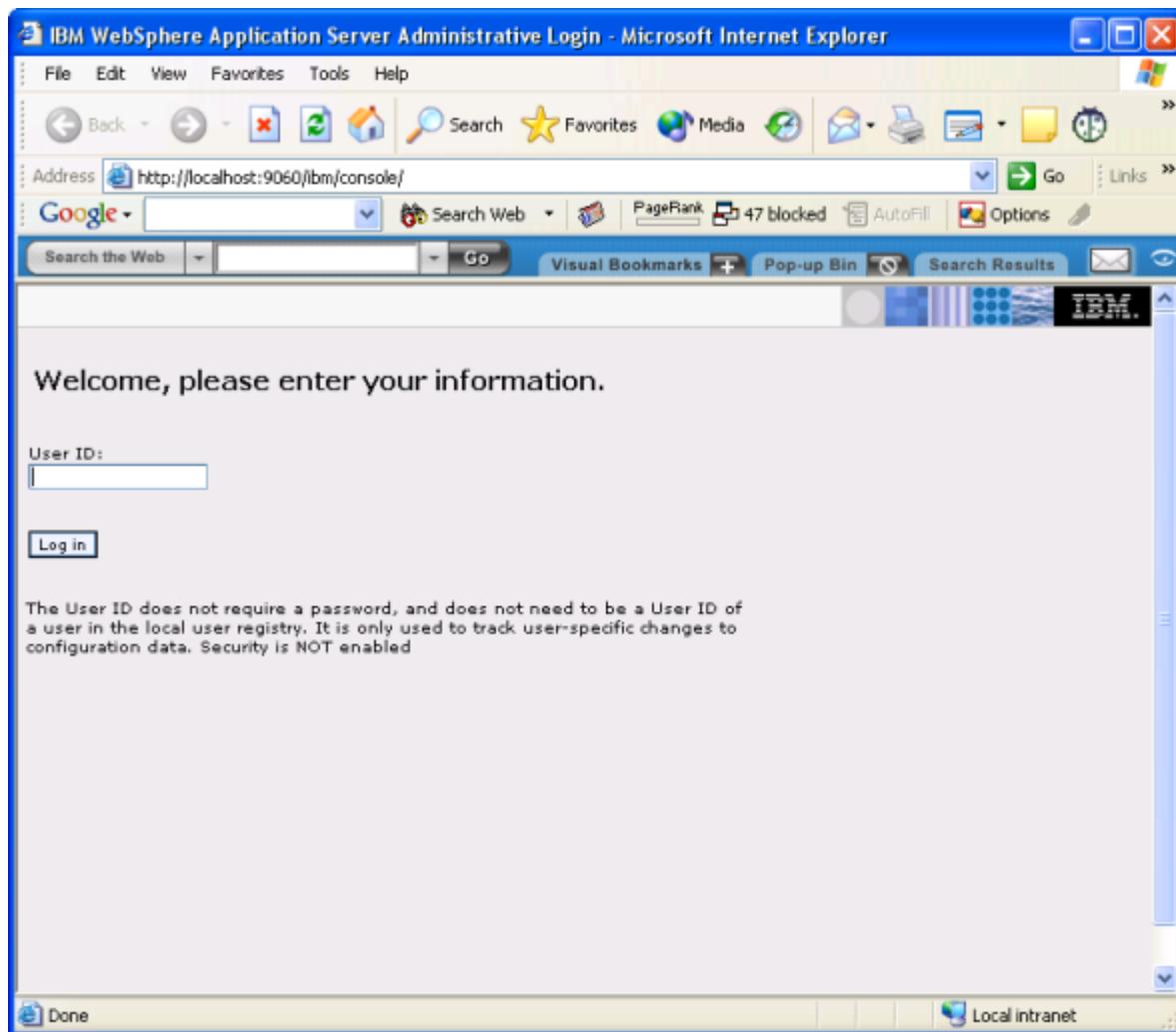
Start the administrative console

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

Information: In a Network Deployment cell you always connect to the console on the deployment manager so that changes are synchronized across the cell. In a stand-alone application server, you connect directly to the console on the server.

- ___ 1. Open the administrative console.
 - ___ a. In a Windows environment: **Start** → **Programs** → **IBM WebSphere** → **Application Server Network Deployment v6** → **Profiles** → **profile1** → **Administrative console**
 - ___ b. Another option is to open a browser and specify the address:
`http://localhost:9060/ibm/console`

The administrative console should open and look similar to the image below:

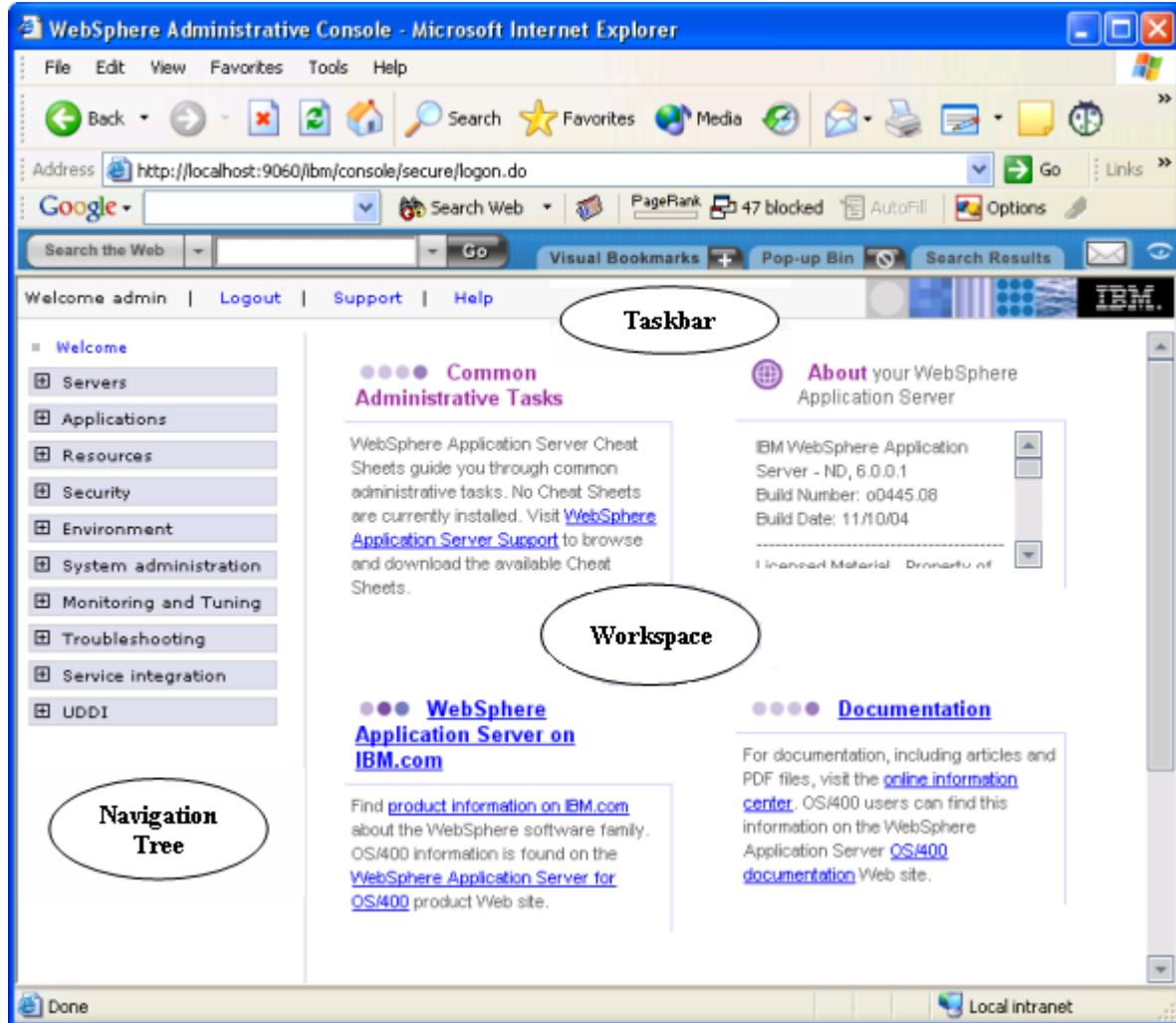


- ___ 2. Enter **admin** for the userid and click **Log in**.

The user ID you enter here does not matter when WebSphere Application Server security is not enabled, it is just used to track configuration changes. If security is enabled you must log in using a valid user ID and password.

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

The main console page appears:



The administrative console displays in three frames:

- Taskbar - this area is the top of the console. It shows a welcome for the userid you logged into the console. It displays links for Support assistance, logging out of the console, and accessing product information.
- Navigation tree - this area is the left frame of the console. It displays the types of information you can configure. There are ten areas: Servers, Applications, Resources, Security, Environment, System administration, Monitoring and Tuning, Troubleshooting, Service integration and UDDI. You also have a Welcome link which takes you back to the main workspace Home page.

- Workspace - this area is the right frame of the console. It displays the pages to create or change configuration information.

The home page displayed in the workspace shows the product version installed and links to various IBM product Web sites including the developerWorks WebSphere site.

Explore server settings

In this part of the exercise, you look at some of the settings that can be configured using the administrative console, starting with servers.

- ___ 1. In the navigation tree, expand **Servers**.

In the base product the only server types are Application servers and Web servers. In Network Deployment you can also manage Generic servers, JMS Servers, Clusters, Cluster Topology and Core groups.

- ___ 2. Click **Application servers**.

In the workspace a table listing the application servers will be displayed. You should have one server, server1.

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

Note: For some other collection pages, an additional option, Scope is presented. You see an example of scope later.

3. Select the Filter icon.

[Application servers](#)

The screenshot shows the 'Application servers' interface. At the top, there's a header bar with a question mark icon and a close button. Below it, a section titled 'Application servers' defines what an application server is. A 'Preferences' button is shown with a red circle around its icon. Below this are buttons for 'Select', 'Name' (with a dropdown arrow), 'Node' (with a dropdown arrow), and 'Version' (with a dropdown arrow). A message says 'To filter the following table, select the column by which to filter, then enter filter criteria (wildcards: *, ?, %)'. It includes a 'Filter' dropdown set to 'Name', a search input with the placeholder '*', and a 'Go' button. A table below shows one row: a checkbox column, a 'server1' name column, a 'was6host00Node01' node column, and a '6.0.0.1' version column. At the bottom left of the table is the text 'Total 1'.

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 rarely needs to be used unless there are a lot of items.

4. Expand Preferences.

The preferences feature allows you to control the maximum number of rows in the table and whether the filter is retained for use whenever you return to this page.

5. Select **server1**

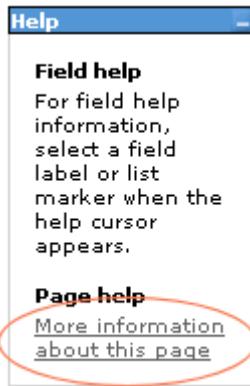
The configuration of server1 is displayed. This page is known as a details page. Notice there are two tabbed pages, **Runtime** and **Configuration**. Configuration lists

the saved settings that are used when the server is next started. Runtime lists the current information used by the running server.

The screenshot shows the 'Application servers' configuration interface for 'server1'. The 'Runtime' tab is selected. On the left, under 'General Properties', the 'Name' is set to 'server1', and 'Parallel start' is checked. Under 'Server-specific Application Settings', 'Classloader policy' is set to 'Multiple' and 'Class loading mode' is set to 'Parent first'. On the right, there are several sections: 'Container Settings' (Web Container Settings, EJB Container Settings, Container Services, Business Process Services), 'Server messaging' (Messaging engines, Messaging engine inbound transports, WebSphere MQ link inbound transports, SIB service), 'Server Infrastructure' (Java and Process Management, Administration), 'Communications' (Ports, Messaging), and 'Performance' (Performance Monitoring Infrastructure (PMI), Runtime Performance Advisor Configuration). At the bottom are 'Apply', 'OK', 'Reset', and 'Cancel' buttons.

Some basic configuration settings are shown here under General Properties, including the classloader policy and the class loading mode. Under Additional Properties area it lists other pages of properties that can be viewed for this server.

For a description of any of the settings click the **More information about this page** in the Help box.



- ___ 6. Under **Server Infrastructure** click to expand **Java and Process Management**. Select **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.

Note that the Working directory entry that starts with a \$, such as \${USER_INSTALL_ROOT}. These are called WebSphere Variables. They allow for substitutions to the absolute paths using symbolic names. You explore these later in this exercise.

- ___ 7. Select **Java virtual machine** under Additional Properties

The advanced JVM settings for server1 are displayed. 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 debugging enabled?

Is the JIT (just-in-time compiler) enabled?

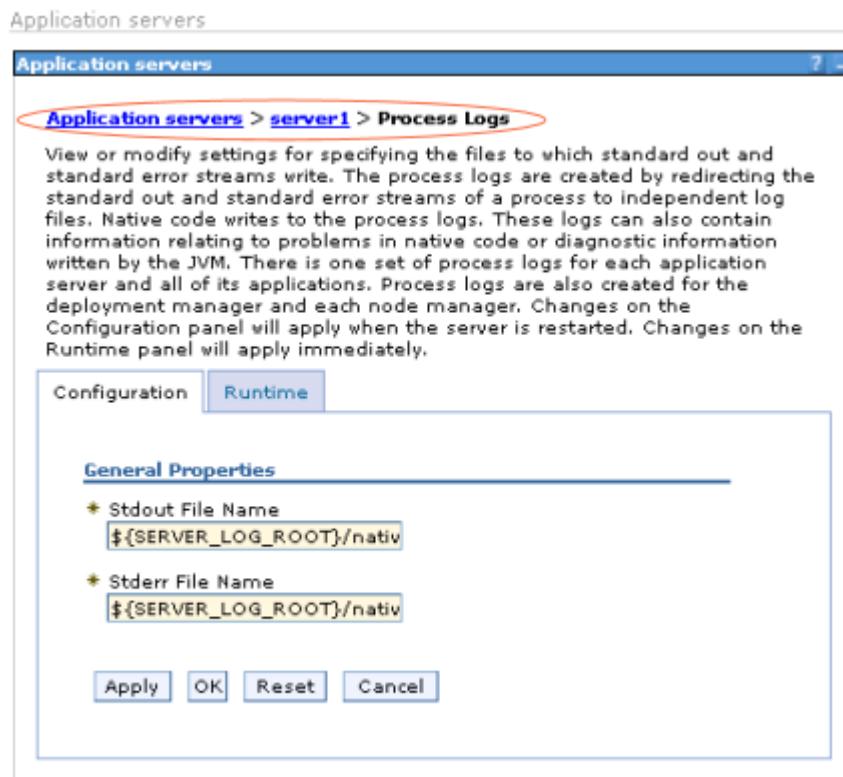
- ___ 8. Click **Cancel** to return to the Process Definition page.

- ___ 9. Select **Process Logs** under Additional Properties

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

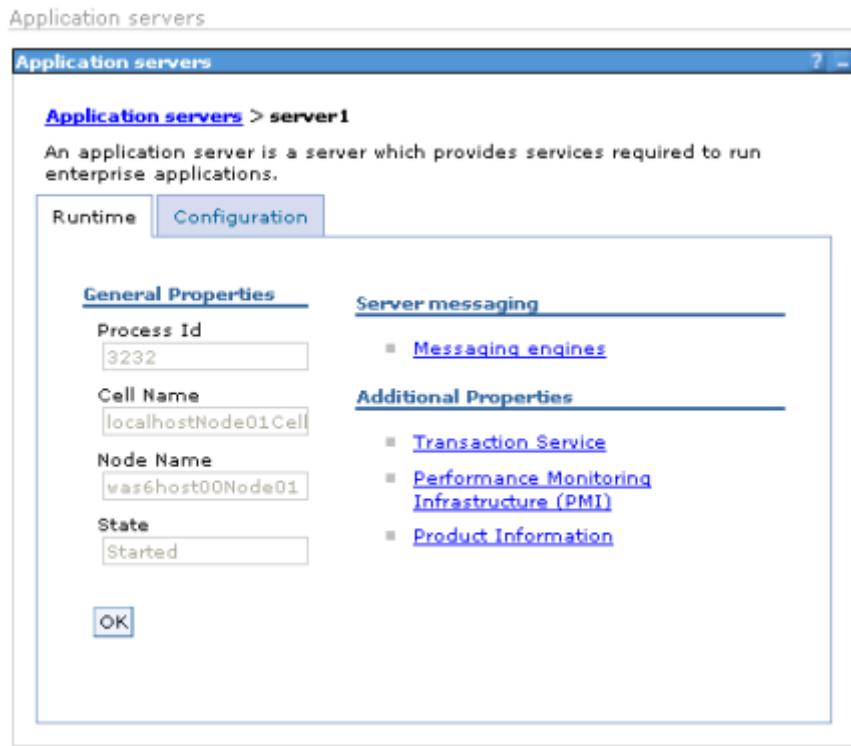
Note: These are different from the System.out and System.err log files which capture most output from the JVM, and support log file rotation to prevent the files from growing too large.

- ___ 10. At the top of the page, notice there is a breadcrumb trail showing the pages you have visited.



- ___ 11. Select **server1** from the breadcrumb trail to return to the server1 Configuration page.
- ___ 12. Under **Communications** expand **Ports**. The TCP/IP ports used by server1 are listed. Select the **details** button to get additional information on these ports.
- ___ 13. Select **SOAP_CONNECTOR_ADDRESS** to configure the port.
The host and port associated with the SOAP listener are displayed under the Port column. This is used by SOAP clients such as the wsadmin command to connect to the server to perform administration tasks. For a single server installation the default SOAP port is 8880.
- ___ 14. Select **server1** in the breadcrumb trail to return to the server1 details page.
- ___ 15. Under **Server Infrastructure** expand **Administration**. Select **Server Components**.
The internal components of server1 are listed. The components can be **Name Server** and an **Application Server**.
- ___ 16. Select **Name Server**
The Name Service settings for the application server are displayed.
- ___ 17. Select **server1** in the breadcrumb trail to return to the server1 details page.
- ___ 18. Select the **Runtime** tab.

The properties of the currently running instance of server1 are displayed. This includes the process ID, cell name, node name and state (which should be: Started).



- ___ 19. Restore the Task Manager window, and verify that the process ID shown, matches the PID for the java.exe process in the Processes list. Then minimize Task Manager.

UNIX: Issue `ps -ef | grep <PID>` from a terminal command prompt.

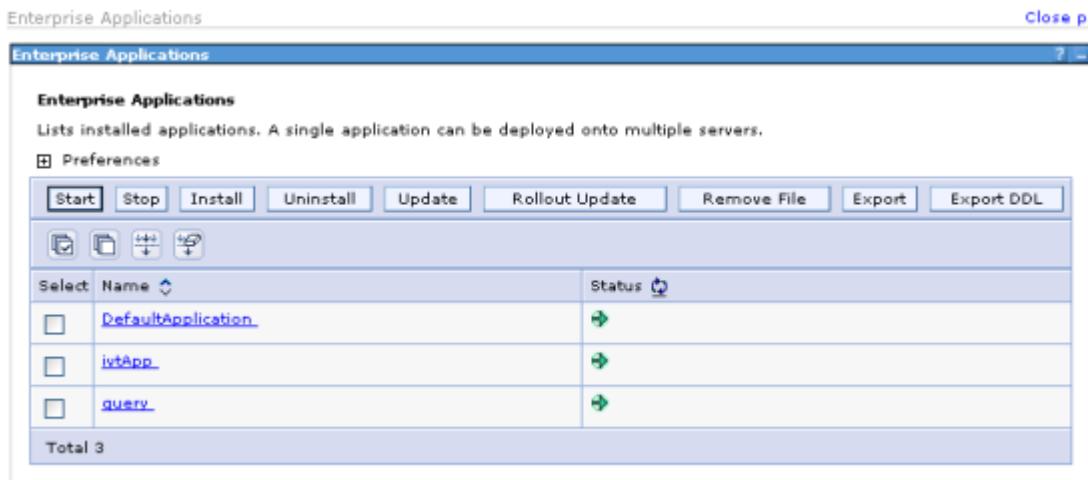
Examine application settings

- ___ 1. In the navigation tree, expand **Applications**.

There are two options, Enterprise Applications and Install New Application.

- ___ 2. Select **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.



The applications installed by default include: DefaultApplication (that includes the snoop servlet you used earlier), ivtApp (installation verification test) and query (for information on EJBs). If the sample applications were installed, they would also be listed here.

At the moment you only have one server, but potentially the list could include applications installed on multiple servers on the same machine or in a network deployment cell.

Tasks you can perform on an application include start, stop, install, uninstall, update, rollout update, remove file, export or export DDL.

3. Select **DefaultApplication**

A detail page shows the General Properties of the application with links to a number of Additional Properties pages and also Related Items pages which allow you to view settings for the Web modules, EJB modules and Connector modules in the application.

4. Select the **Local Topology** tab.

The application topology is displayed as a collapsed tree structure.

5. Expand the complete tree structure for DefaultApplication.

You should see one Web module, **DefaultWebApplication.war**, and one EJB module, **Increment.jar**.

6. Select **DefaultWebApplication.war**.

A detail page shows the General Properties associated with the deployment of the Web module. Under Additional Properties you can change the server the Web

module is mapped to, override session management settings, and view the deployment descriptor for the Web module.

- ___ 7. Select **DefaultApplication** in the breadcrumb trail to return to the DefaultApplication detail page.
- ___ 8. Scroll down and select **EJB modules** under Related Items.
- ___ 9. A collection page lists EJB modules in the application. Only Increment.jar should be listed (as you saw on the Local Topology page). This is an alternative way of listing the modules in the application.
- ___ 10. Select **Increment.jar**.

A detail page shows the General Properties associated with the deployment of the EJB module. From here and Additional Properties you can change the server the EJB module is mapped to, and view the deployment descriptor for the EJB module.

Examine environment settings

- ___ 1. In the navigation tree, expand **Environment**

The Environment options are Virtual Hosts, WebSphere Variables, Shared Libraries, Replication Domains and Naming.

- ___ 2. Click **Virtual Hosts**.

In the workspace a collection page lists the Virtual Hosts defined for the cell.

- ___ 3. Click **default_host**.

A details page shows the details for the virtual host. You can only directly change 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 which are associated with this virtual host are displayed. For default_host the default values are *:80 (any host on the standard Web server HTTP port), *:9080 (any host on the internal HTTP transport port) and *:9443 (any host on the internal SSL transport port). In some cases you may need to define additional virtual hosts or modify default_host to support additional host/port combinations. For example, you may wish to add *:443 to support SSL connections through the Web server to WebSphere Application Server.

- ___ 5. Click **default_host** in the breadcrumb trail to return to the default_host page.

- ___ 6. Click **MIME Types** under **Additional Properties**.

The MIME types associated with this virtual host are listed.

- ___ 7. Click **Virtual Hosts** in the breadcrumb trail to return to the Virtual Hosts page.

- ___ 8. Click **admin_host** and then click **Host Aliases**.

- ___ 9. Examine the admin_host virtual host and write down the port numbers associated with this virtual host: _____

Note: The browser should be accessing the administrative console on one of the ports associated with the admin_host virtual host.

- ___ 10. In the navigation tree, click **WebSphere Variables**.

In the workspace a WebSphere variables collection page is displayed. This page includes the scope feature, because variables can be defined for a Cell, Node or Server.

- ___ 11. Select scope **Cell** and click **Apply**. How many variables are defined for the cell? _____

- ___ 12. Select scope **Node** and click **Apply**. How many variables are defined for the node? _____

If there are more than the maximum rows (20 by default) a Next button is displayed to allow you to see the additional entries.

Notice that many variable values include references to other variables, for example, \${USER_INSTALL_ROOT}.

- ___ 13. Select scope **Server** and click **Apply**. How many variables are defined for the server? ___

Examine resource settings

- ___ 1. In the navigation tree, expand **Resources**.

The resources options are JMS Providers, JDBC Providers, Resource Adapters, Asynchronous beans, Schedulers, Cache instances, Object pool managers, Mail Providers, URL Providers and Resource Environment Providers.

- ___ 2. Select **JDBC Providers**

A collection page lists the JDBC providers in the configuration. There are no JDBC providers configured by default, unlike previous versions of WebSphere Application Server, because a relational database is no longer required to install the product.

In a later exercise you configure a JDBC driver and Data Source for an application.

- ___ 3. Expand **JMS Providers** and select **Default messaging**.

A details page shows some basic properties of the internal JMS provider. There are three Connection Factories pages for configuring a JMS connection factory, JMS queue connection factories and JMS topic connection factories. Under Destinations you can see JMS queue and JMS topic. These settings can be defined at the cell, node or server level so there is a scope selection option available.

- ___ 4. Select scope **Server** and click **Apply**.

- ___ 5. Select **JMS queue connection factory** under **Connection Factories**.

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.

- ___ 6. Return to the **Default messaging provider** page and select **JMS topic connection factories**.

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.

- ___ 7. If you wish return to the **Default messaging provider** page and explore the Destinations for JMS queue and JMS topic.

Information: In future lab exercises you modify settings for some of the object types you have already explored and for other areas you have not yet explored.

Examine Troubleshooting

The Troubleshooting area displays messages about run-time events and configuration problems. The area automatically refreshes and you can view both the run-time messages or configurations problems totals.

- ___ 1. In the navigation tree, expand **Troubleshooting**.
- ___ 2. Expand **Runtime Messages**. You see entries for **Error**, **Warning** and **Information**.
The total number of errors, warnings and information messages will be displayed when you select one of the options. Select **Error** to view all of the error messages.
- ___ 3. In the **Message** column, click one of the messages to see the message detail. This information is retrieved from the WebSphere Application Server logs.
- ___ 4. Click **Runtime Events** in the breadcrumb trail to return to the message list and view other messages.
- ___ 5. In the navigation tree, select **Configuration Problems** under **Troubleshooting**. You see entries for **Error**, **Warning** and **Information**.

The total number of errors, warnings and information messages are displayed when you select one of the options. Select **Error** to view all of the error messages. If you do not have any error messages select the **Information** messages.

- ___ 6. If you have a configuration problem, click the link to it. The problem detail is displayed.
On the next window you should see General Properties information about the configuration problem.
- ___ 7. Click the **Back** button to return to the Configuration Problems list and view other problems.

Logout of the administrative console

When you are working in the administrative console a workspace is saved which includes all configuration changes you make in the session. When you log out you have the option to save or discard these changes. If you just close the browser, the session workspace will be preserved. The next time you log in you have the option to recover the workspace from the previous session.

Information: If you have made configuration changes that you want to keep, these 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 you have not made any changes to resources, you are returned to the login page. If changes were made, the Save page is displayed. Click **Discard** so that you do not overwrite the configuration.

The Discard WorkSpace Changes page appears asking you to confirm the discard. Select **Yes** and the login page is displayed.

- ___ 3. Close the browser.

Congratulations. You now know your way around the WebSphere administrative console

End of lab

Exercise review and wrap-up

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

Exercise 4. Using AST to assemble the Trade Application

What this exercise is about

This lab covers the steps used to assemble the Enterprise JavaBean (EJB) Java Archive (.jar) files, the Web Archive (.war) files, and auxiliary files that comprise an Enterprise Application into an Enterprise Application Archive (.ear) file.

What you should be able to do

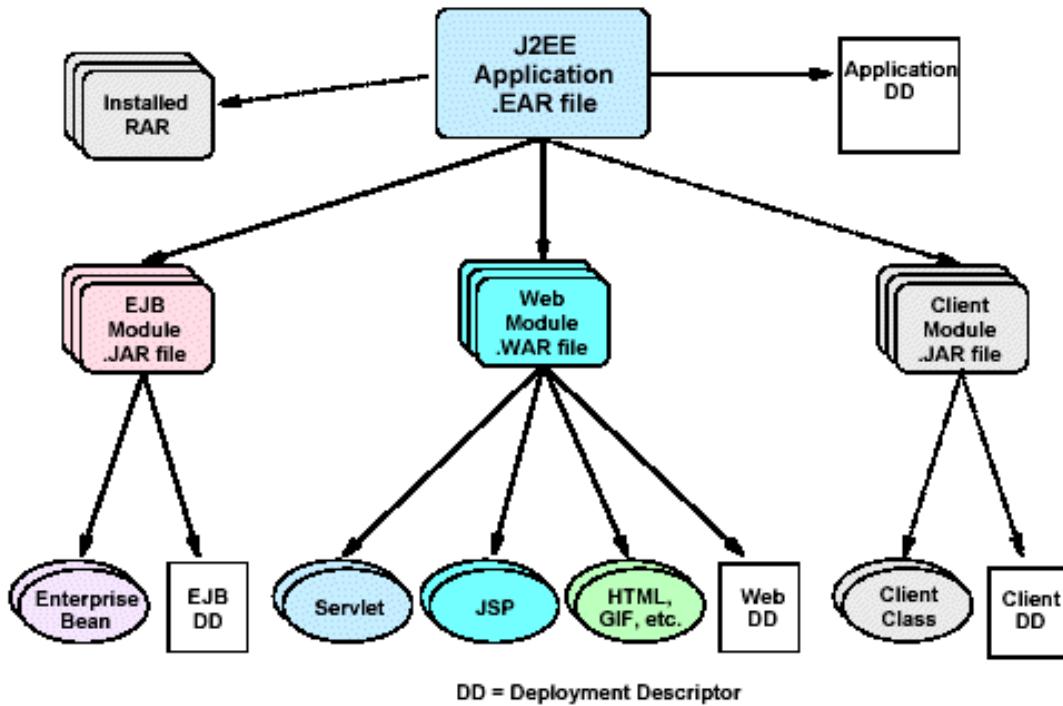
At the end of the lab, you should be able to:

- Navigate AST
- Import JAR and WAR files which make up the application
- Define a data source at the application scope
- Define an authentication alias at application scope
- Export an enhanced EAR file ready for deployment

Introduction

In this exercise, you perform the role of the application assembler. Assume that the development groups have created and tested the application in their test environment. Those responsible for bean development have provided you with the .jar files that contain the deployed code for the EJBs. The developers responsible for the presentation design have given 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.

Information: The AST tool, with the application assembly plug-in, is currently only available for Windows and Linux.



You will be using files provided by the application developers. The application is made up of:

- One EJB Module .jar file that contain the enterprise beans and a deployment descriptors
- One Web Module .war file that contain the servlets, JSPs, and presentation (HTML and graphics) files along with a deployment descriptors

In some cases, you might also be given a Resource Adapter Archive Module .rar file. For this exercise, however, there are no Resource Adapter Archive modules to be installed.

Required materials

To complete this exercise, you need WebSphere Application Server Toolkit installed on your machine. AST will be used to complete the exercise. You also need the following files, located in

<software_dir>\Assemble:

- TradeWeb.war
- TradeEJB.jar

Instructor exercise overview

In this exercise the students will take the actual components of an enterprise application and create an .ear file. This is an overview of this task, using .jar files and .war files.

The files needed for this exercise are located in <software_dir>\Assemble. The output of this exercise can be used in the follow-on lab, Install the Trade Application.

After this exercise, the students will have created a TradeApplication.ear file in the <profile_root>\profile1\installableApps directory.

For recovery purposes, a known good copy of the file called TradeApplication.ear is also in the <software_dir>\Solutions directory.

The soapbind:header warning, appearing in the Problems view (on page 4-16), is the result of AST not being able to properly validate this element in WSDL files. You may see this warning in more than one WSDL file in this application. It can be safely ignored and should be fixed in a future release of AST and IRAD.

Exercise instructions

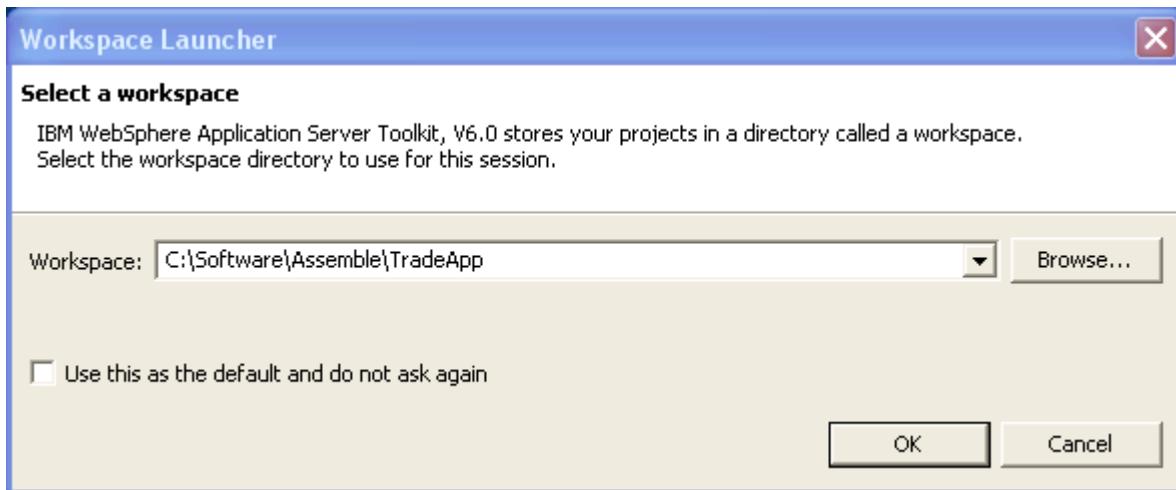
Start Application Server Toolkit

- 1. Click Start → Programs → IBM WebSphere → IBM WebSphere Application Server Toolkit, V6.0 → Application Server Toolkit in Windows.

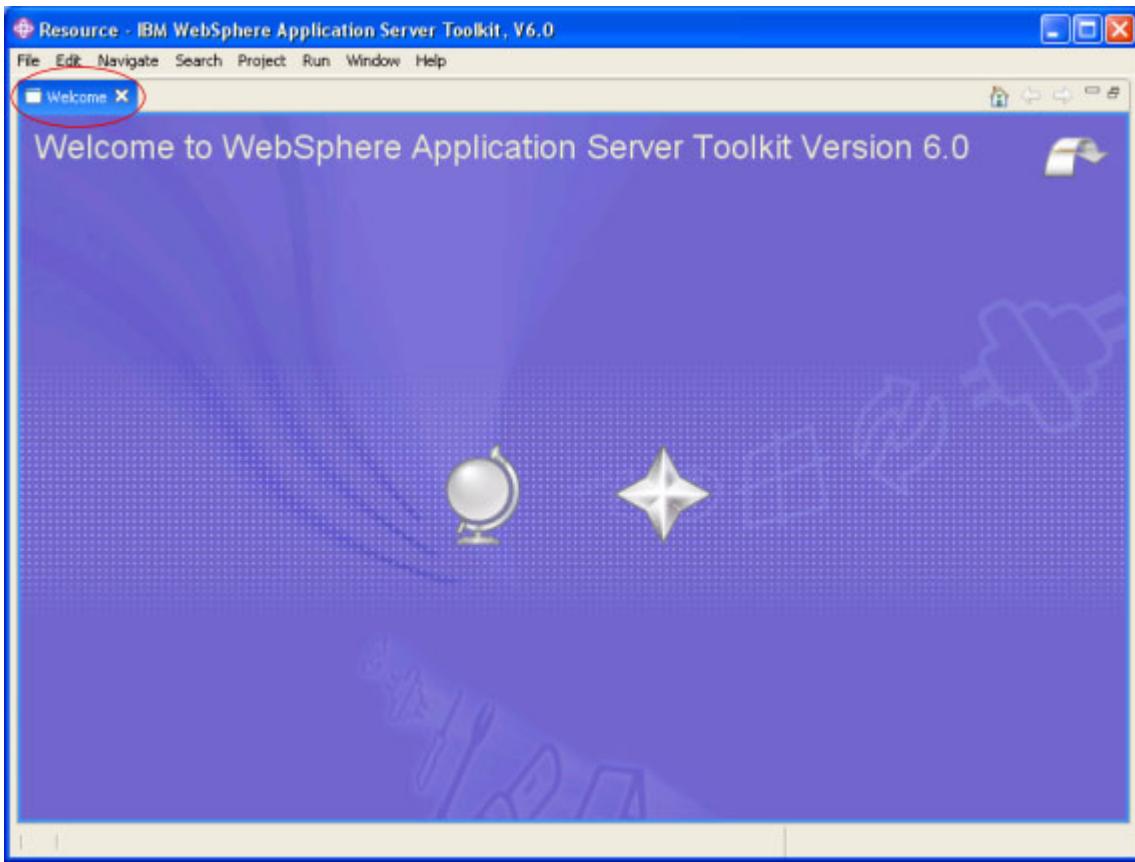
UNIX: Start the AST by navigating to /opt/IBM/WebSphere/AST and invoking ./ast.)

The AST starts by asking you which directory to use for its workspace. Once you establish a workspace you continue to use it for the project. Each project should have its own separate workspace.

- 2. Enter <software_dir>\Assemble\TradeApp and click OK.



After a few moments the Welcome screen for AST opens. From the Welcome screen you can get an Overview of the product or find out What's New in this release by selecting the icons.



- ___ 3. If open, close the **Welcome** view inside of AST by clicking the **x** to the right of its title. Do not close AST.

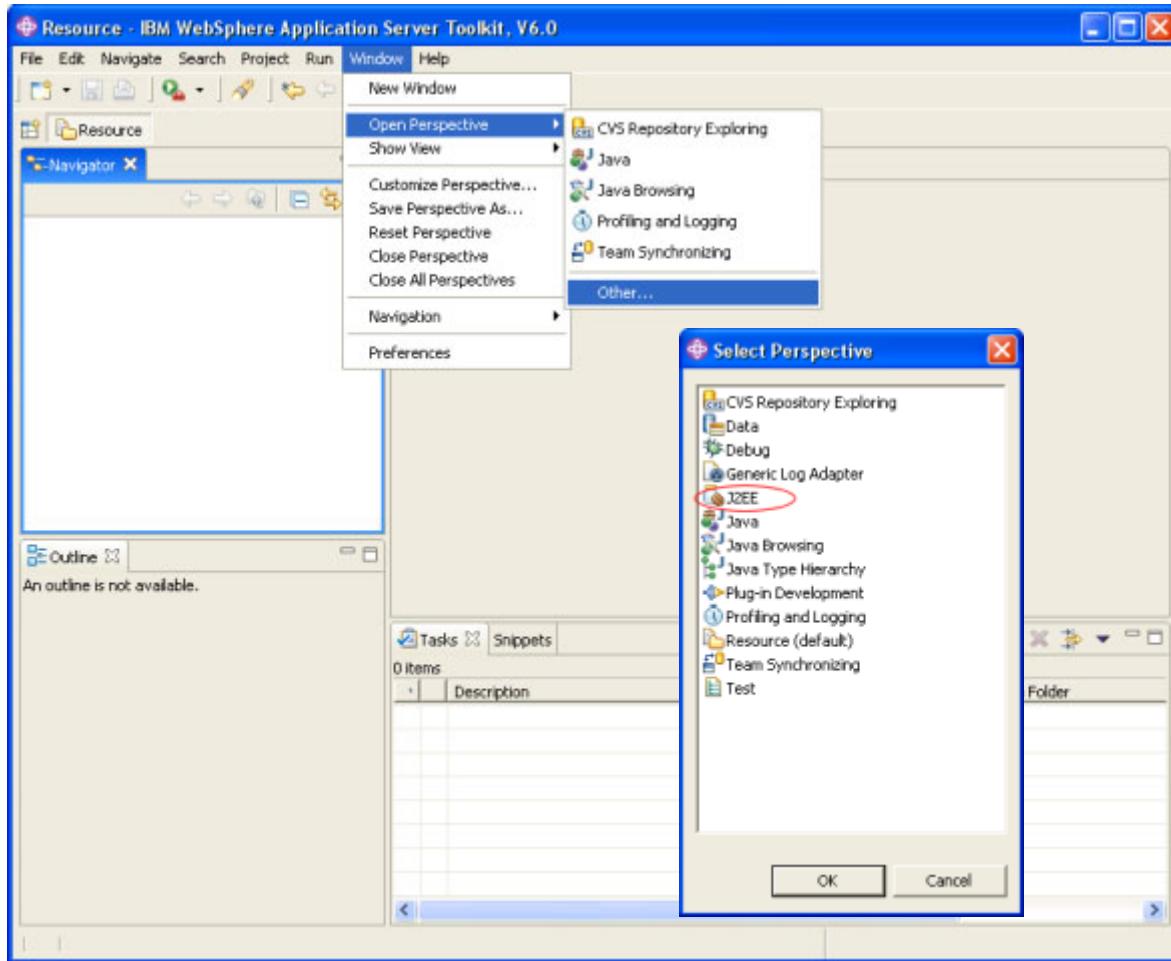
Import the Trade Application modules

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

Information: Handing over application modules is the most common way to receive application components, especially when more than one development team is involved, each team being responsible for one or more modules. Another way is to hand over an application for deployment is to receive an Enterprise Application's .jar file.

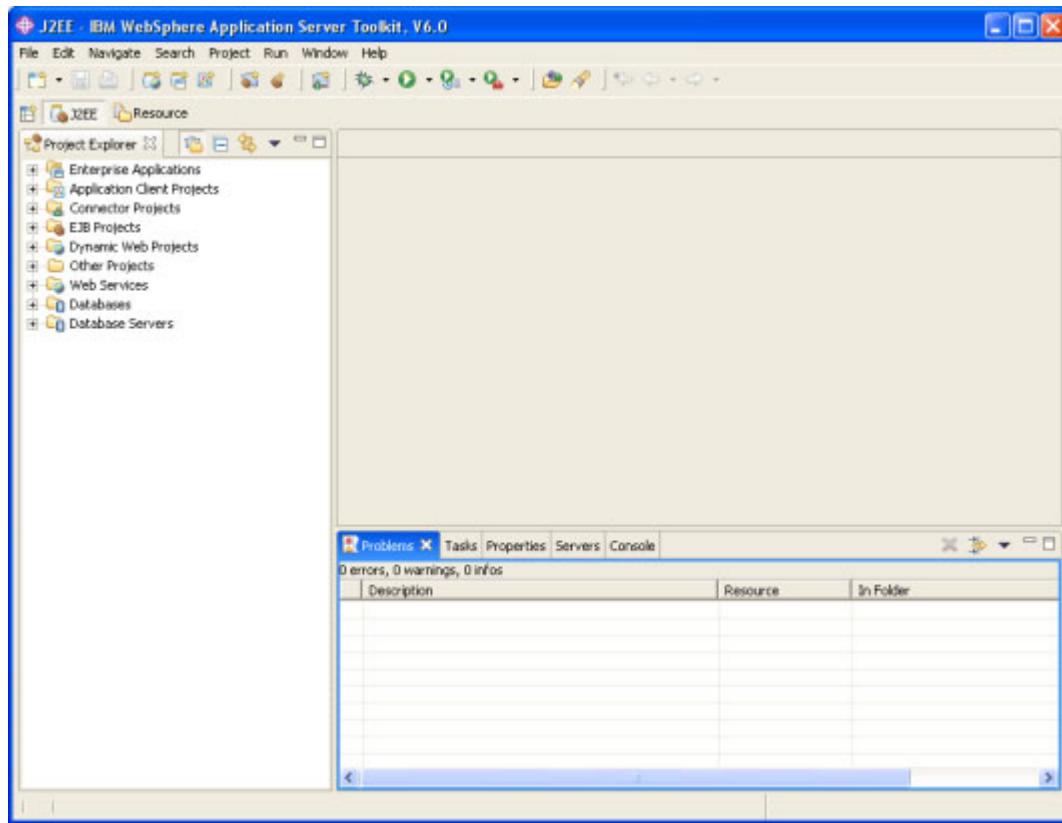
- ___ 1. Switch to the **J2EE** perspective to begin assembling the application.
 - ___ a. From the main menu select **Window** —> **Open Perspective** —> **J2EE**

If the J2EE perspective is not available on the menu, select **Other** and then **J2EE** from the list. Click **OK**.



2. The **Project Explorer** view shows all the components currently loaded in the workspace (none at this time). The components are categorized as per the function

they perform. As you add components they will appear under their corresponding folders.

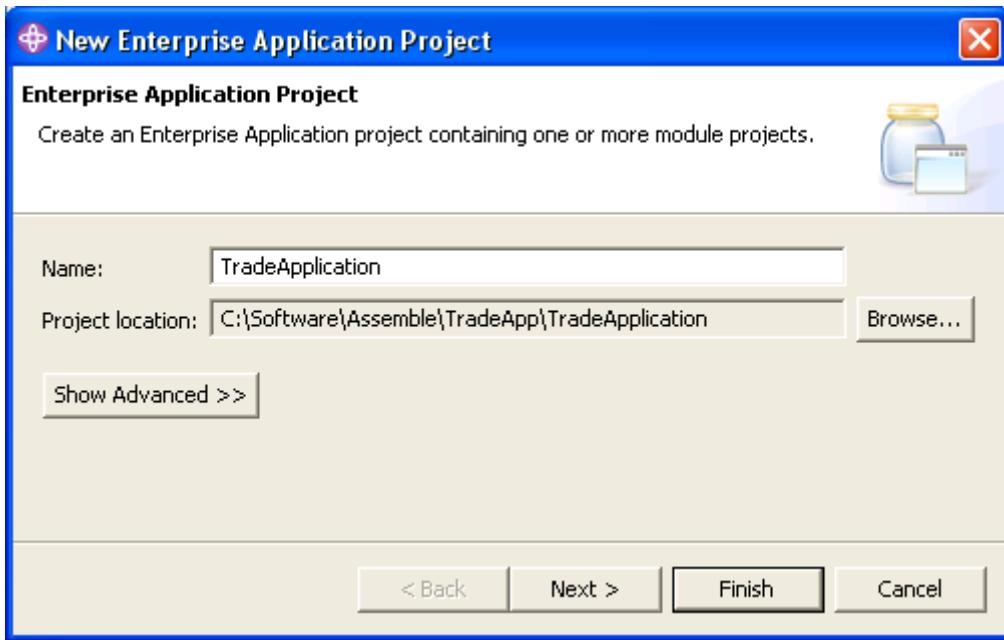


The empty area on the upper right hand side is where the different editors open and display the contents of the selected items on the other views. This area is called the editor pane.

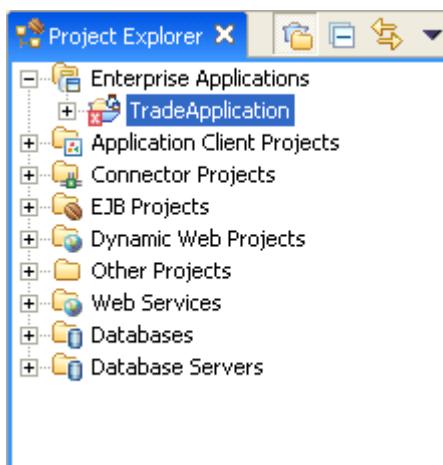
Below the editor pane is a multipurpose pane, containing several views. On the figure above the **Problems** view is currently on the foreground. It contains any outstanding errors that need to be resolved. This space is shared with several other views represented by the tabs at the bottom of the pane. You will get to use some of them in this lab. Feel free to explore and discover.

- 3. The first step is to create a new Enterprise Application Project. Create a new Enterprise Application Project named **Trade Application**.
 - a. Select the **Enterprise Application** folder. Right-click Enterprise Applications and from its context menu select **New —> Enterprise Application Project**.

- __ b. Name the project: **TradeApplication**. Click **Finish**.



- __ c. The TradeApplication project is now found under the Enterprise Applications folder.

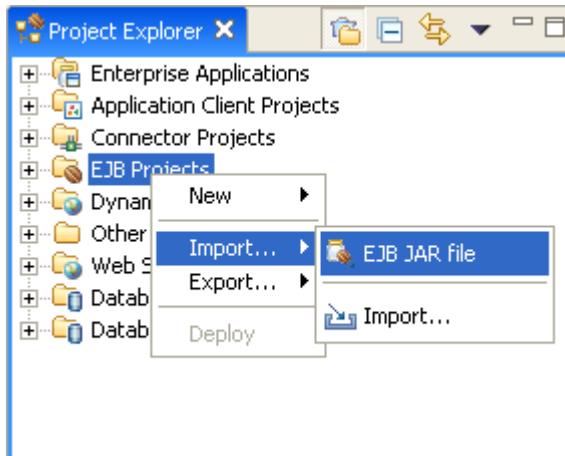


Notice the red error box that appears on the TradeApplication. This will be removed once you begin to add modules to the Enterprise Application.

Add the EJB module

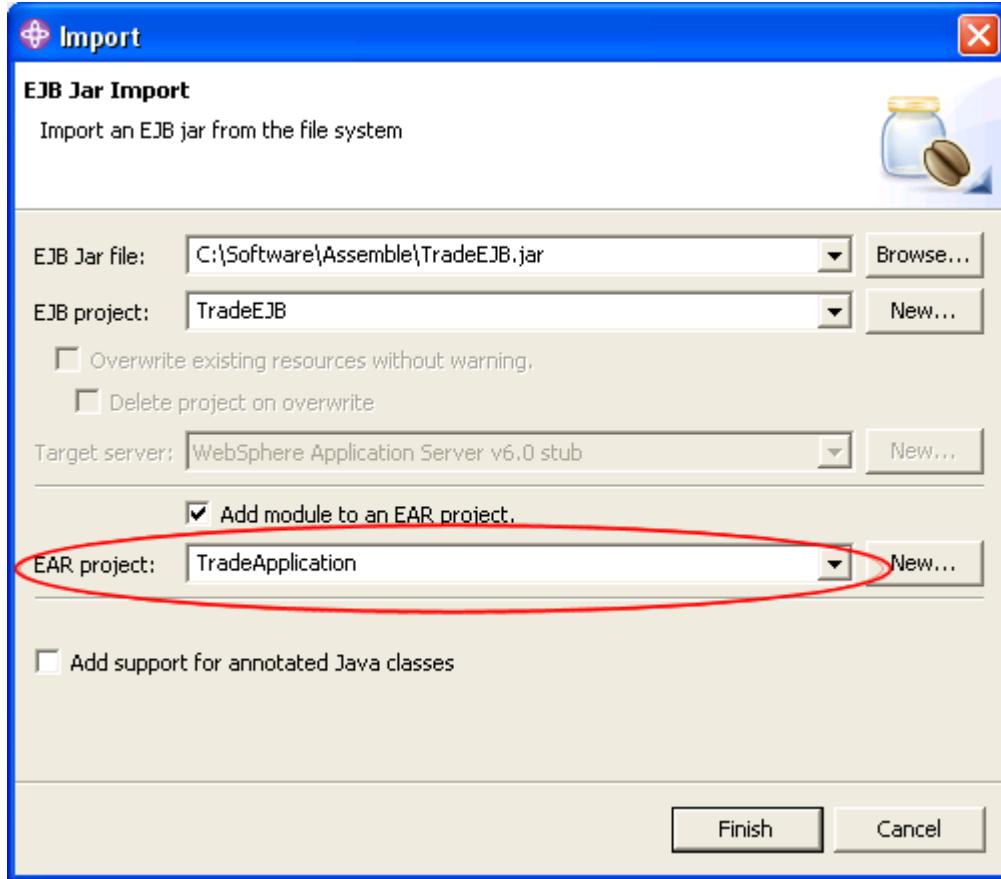
You will define the EJB module that will be used by your application. The EJB .jar file for this application has been given to you by the application developer and is named **TradeEJB.jar**. The .jar files contain the deployed code for the EJBs. This .jar file was created using IBM Rational Application Developer.

- ___ 1. In the Project Explorer view on the upper left pane, right-click **EJB Projects** and from its context menu select **Import —> EJB JAR file**.



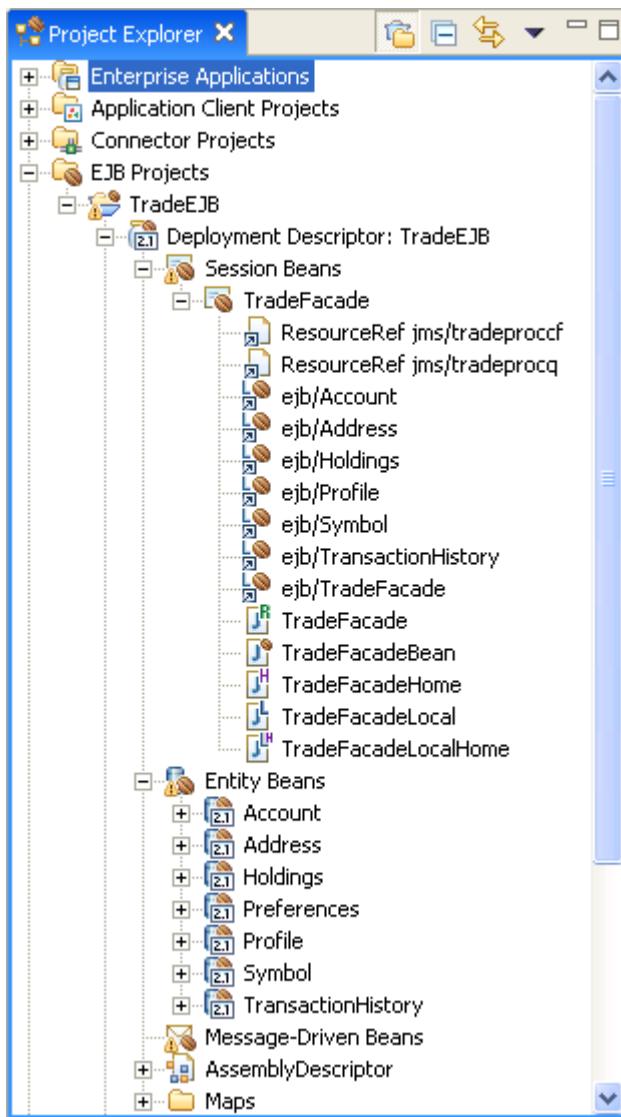
- ___ 2. In the **Import** dialog click **Browse** and navigate to <software_dir>\Assemble, select **TradeEJB.jar** and click **Open**.

3. Select **TradeApplication** from the EAR project drop-down list.



Click **Finish** to add the EJB JAR to the application.

4. The **TradeEJB.jar** file has been added to the application. Expand the module to reveal the details of its contents.



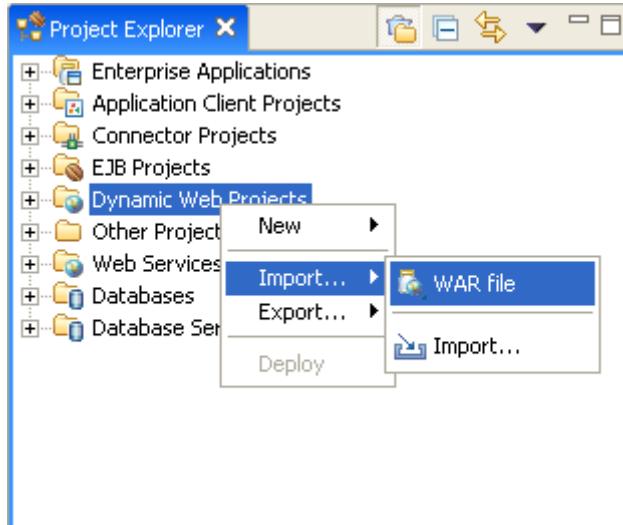
Information: Double-clicking on a module on the Project Explorer view opens its deployment descriptor in a specialized editor. Deployment descriptor editors have tabs along the bottom of the pane to give you access to the various sections of the file. Using these editors makes working with deployment descriptors much easier.

If you change anything on a deployment descriptor, or any other file, you will see an asterisk on the tab where its name is displayed along the top of the editor pane. This indicates that the file has been altered and needs to be saved. The Ctrl+S key combination saves the file.

Add Web modules

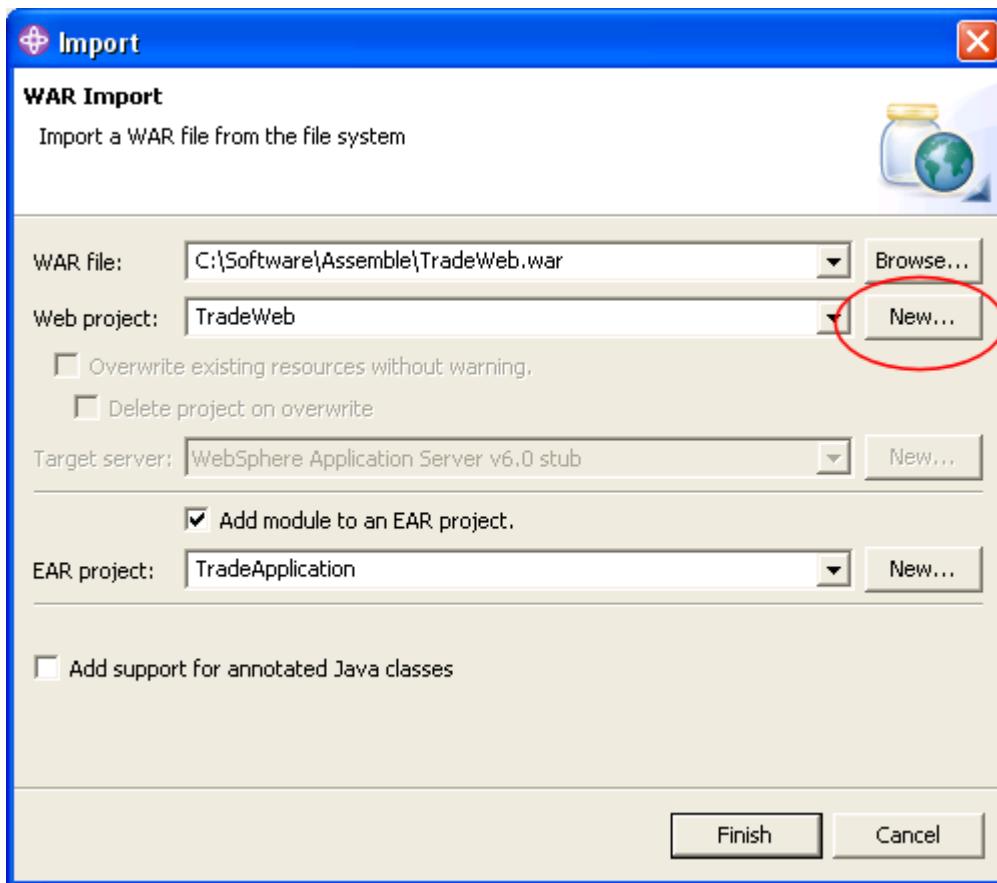
Next you will add the Web module used by your application. The file for the Web module was assembled by the presentation developer using IBM Rational Application Developer. A Web module consists of the JSPs, HTML pages, and servlets contained within the .war file.

- ___ 1. In the Project Explorer view on the upper left pane, right-click **Dynamic Web Projects** and from its context menu select **Import** —> **WAR file**.

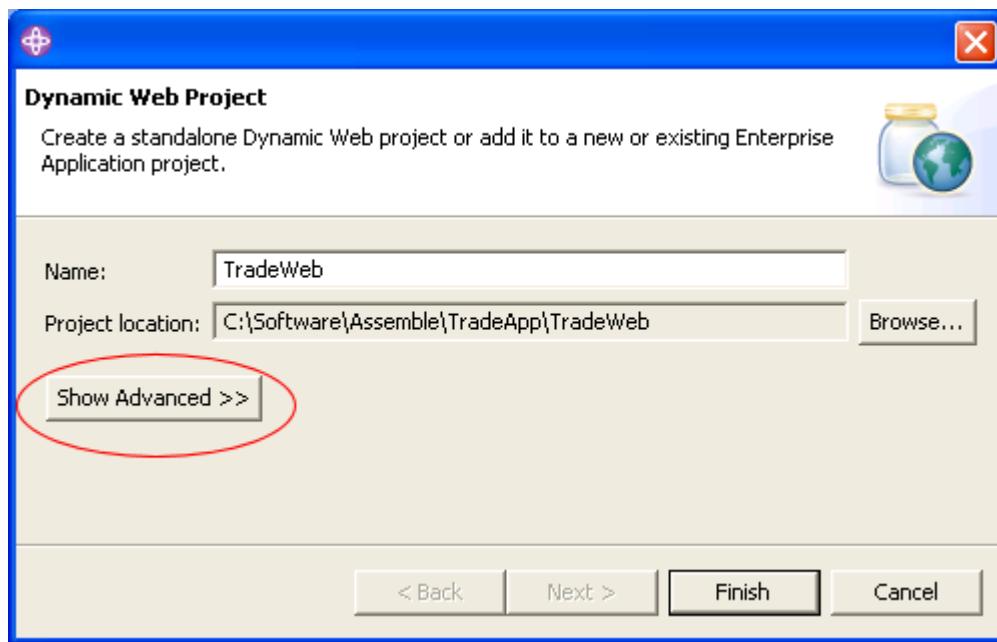


- ___ 2. In the **Import** dialog click **Browse** and navigate to <software_dir>\Assemble, select **TradeWeb.war** and click **Open**.

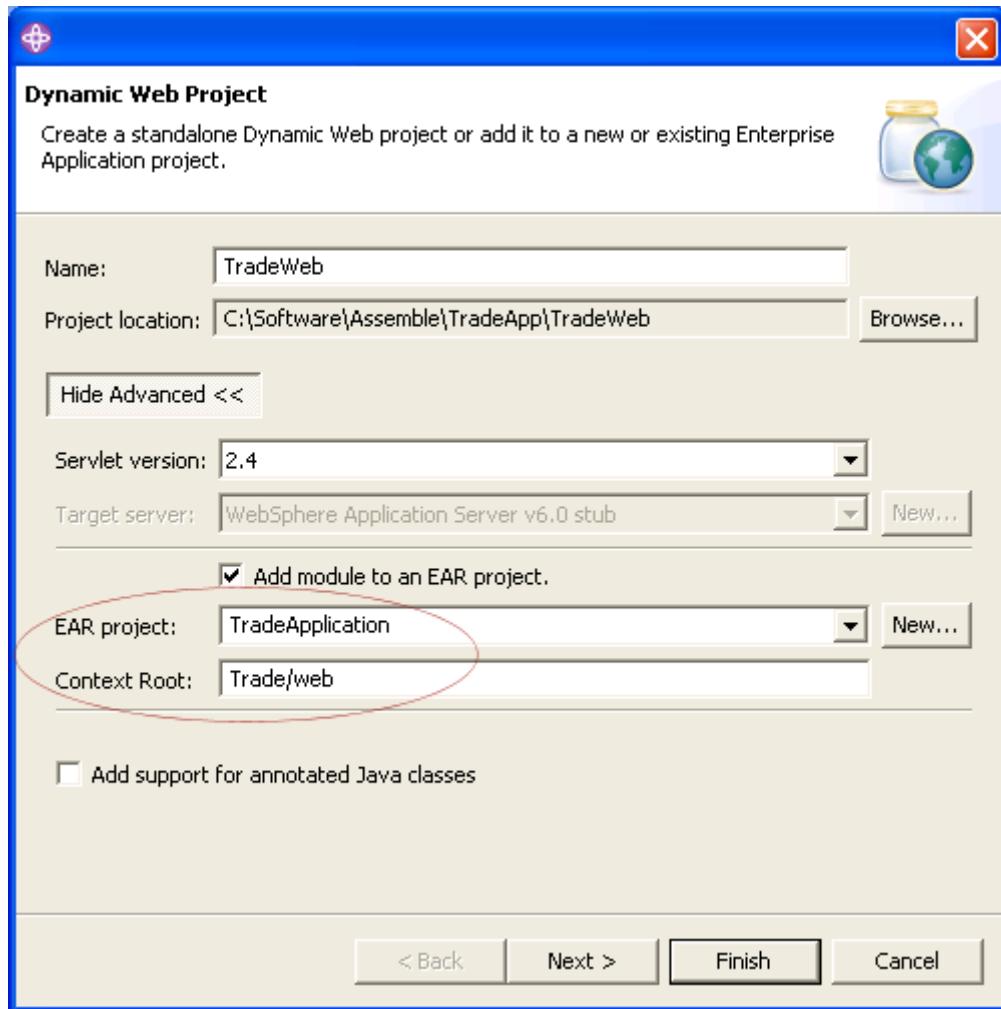
- __ a. On the same dialog, select **TradeApplication** from the **EAR project** drop-down list. Click **New** by the Web project entry field..



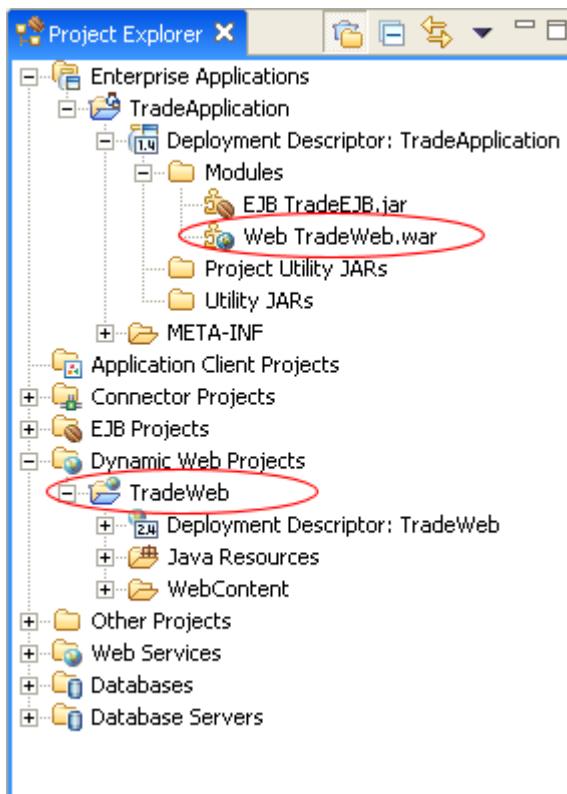
- __ b. Click the **Show Advanced** option.



- ___ c. Notice the window expands to show additional options. Select **TradeApplication** from the **EAR project** drop-down list. For the context root specify **Trade/web**. (*Be careful to type the Context root correctly.*) Click **Finish**.



- ___ d. Click **Finish** again to add the TradeWeb module to the TradeApplication Enterprise Application. Verify that the Web module has been added to the application.



Information: Note that if you did not set the context root when importing the module you can afterwards from its Properties. Right-click **TradeWeb** and from the menu select **Properties**. In the Properties window select **J2EE** and specify the context root for the Web module.

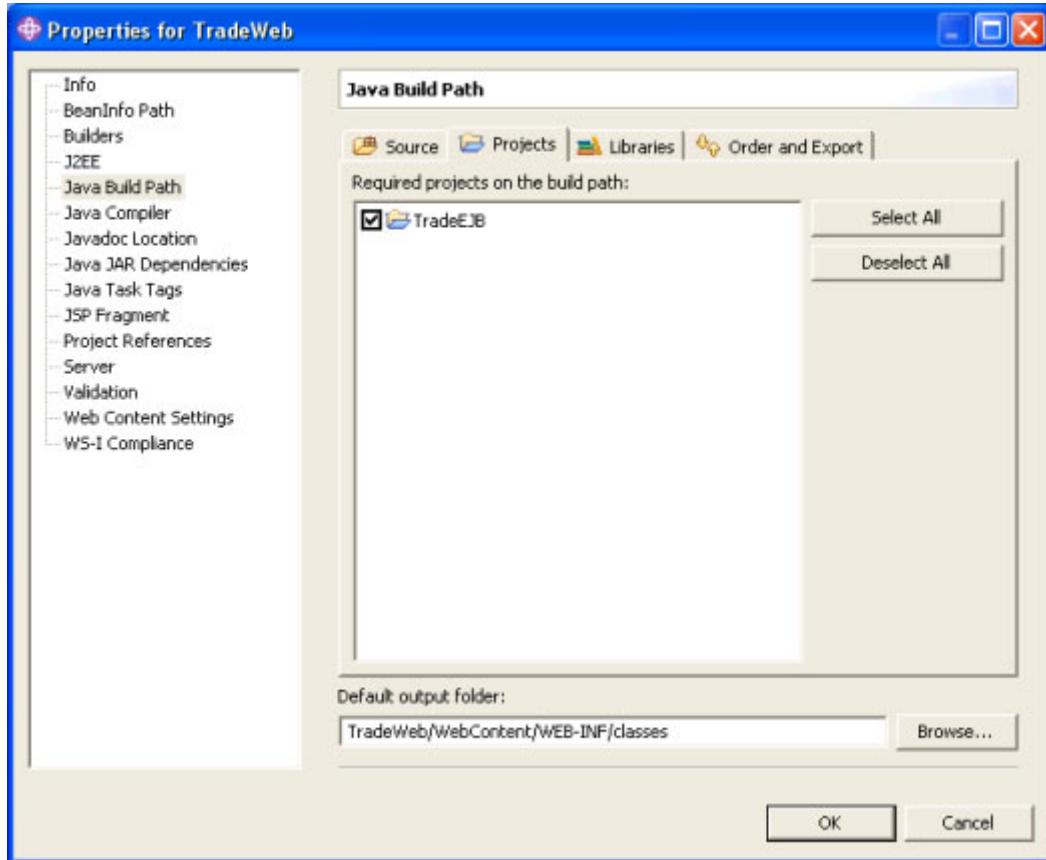
- ___ 3. At this point you will have a number of errors showing on the **Problems** view.

Problems X Tasks Properties Servers Console			
100 errors, 0 warnings, 0 infos (Filter matched 100 of 238 items)			
Description	Resource	In Folder	Location
The import com.ibm.trade.valueobjects cannot be resolved	AccountInfoForm...	TradeWeb/JavaSource/com/ibm/trade/fo...	line 6
AccountVO cannot be resolved (or is not a valid type) fo...	AccountInfoForm...	TradeWeb/JavaSource/com/ibm/trade/fo...	line 18
AccountVO cannot be resolved (or is not a valid type) fo...	AccountInfoForm...	TradeWeb/JavaSource/com/ibm/trade/fo...	line 33
registerVO cannot be resolved or is not a field	AccountInfoForm...	TradeWeb/JavaSource/com/ibm/trade/fo...	line 34
registerVO cannot be resolved	AccountInfoForm...	TradeWeb/JavaSource/com/ibm/trade/fo...	line 43
registerVO cannot be resolved	AccountInfoForm...	TradeWeb/JavaSource/com/ibm/trade/fo...	line 52
registerVO cannot be resolved	AccountInfoForm...	TradeWeb/JavaSource/com/ibm/trade/fo...	line 60
registerVO cannot be resolved	AccountInfoForm...	TradeWeb/JavaSource/com/ibm/trade/fo...	line 68

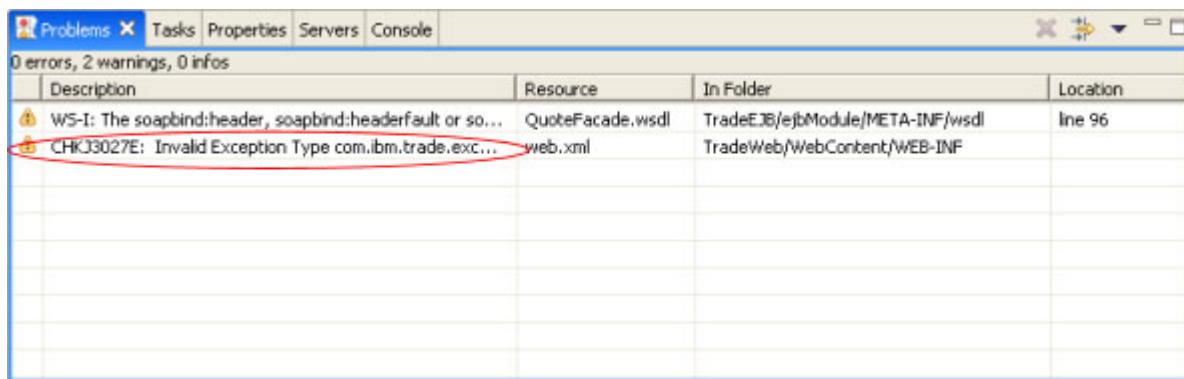
Since the **TradeWeb** module was just added to the workspace its **Java Build Path** has not been defined. When the module was added AST built and integrated the module into the Enterprise Application project, the dependency of this module on the **TradeBaseException** could not be satisfied and the corresponding error was generated. Since you do not intend to run the application inside AST, this warning

does not affect you. However, it is always a good practice to resolve all errors and to investigate any warnings you may see in the **Problems** view.

- ___ 4. To resolve this warning the **TradeWeb** Web module needs to know that some of the classes it refers to are defined in other modules, in this case the **TradeEJB** EJB module. Add the TradeEJB project to the TradeWeb module's build path.
 - ___ a. In the Project Explorer view select **TradeWeb** and from its context menu select **Properties**.
 - ___ b. Select **Java Build Path**. Click the **Projects** tab and select the **TradeEJB** project. Click **OK**.



- ___ c. At this point you may have some warnings showing on the **Problems** view.



The soapbind:header warning, appearing in the Problems view, is the result of AST not being able to properly validate this element in WSDL files. You may see this warning in more than one WSDL file in this application. It can be safely ignored.

The warning is about a definition of the error page which should be displayed when a **TradeBaseException** is thrown by the application. It appears that the newly added Web module does not recognize this exception. At this point you have a couple of choices:

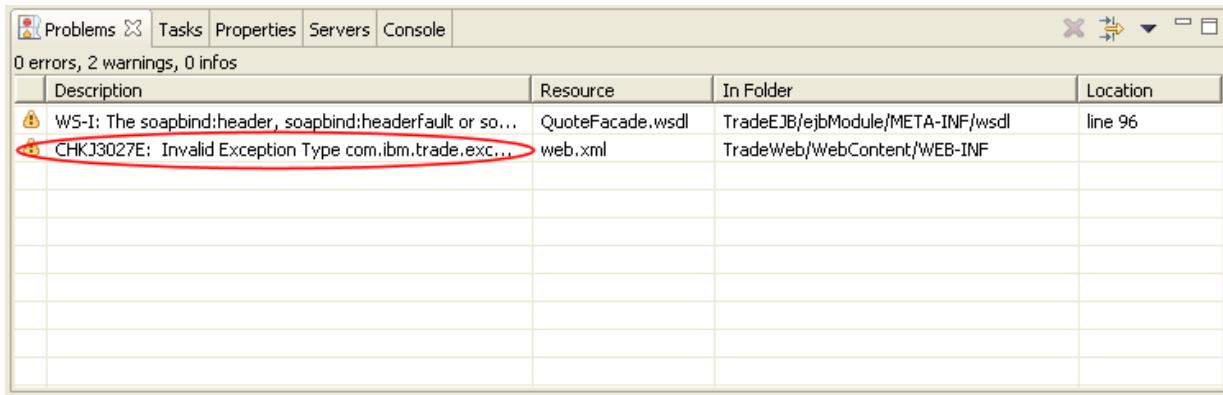
1. Ignore the warning and continue on to the **Configure WebSphere Data Sources** section
2. Find out why this warning is being generated

Fixing the TradeBaseException (optional)

This warning, in this particular case, is quite harmless and will not affect the operation of the application, so if you chose to ignore it, no harm will come of it and you can continue with the exercise below **Configure WebSphere Data Sources**. However, if warnings bother you, as they should, continue here and fix the problem.

As mentioned before it appears as though the TradeBaseException is not being seen by the TradeWeb module.

- 1. Further examine the warning.
- a. Double-click the warning line on the Problems view. The file containing the



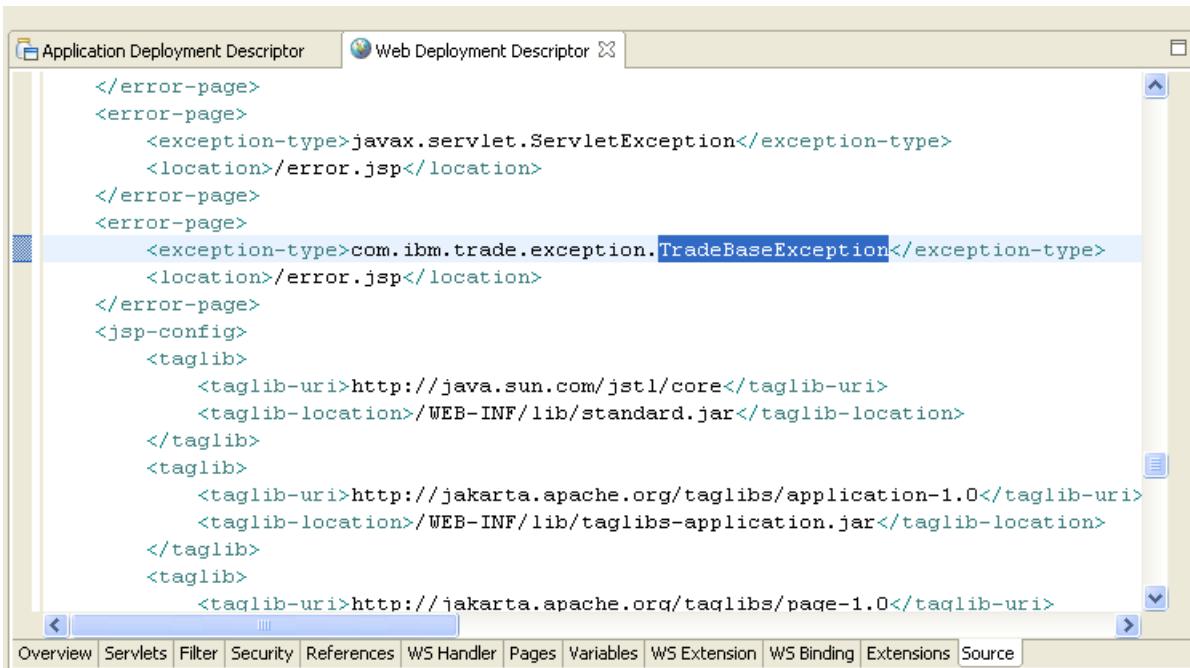
The file containing the warning (or error), in this case the deployment descriptor for the TradeWeb Web module, opens. Select the **Source** tab and search for the exception.

- b. On the deployment descriptor you see that the warning is generated where the different error pages are being defined.

```
<error-page>
<exception-type>com.ibm.trade.exception.TradeBaseException</exception-type>
<location>/error.jsp</location>
</error-page>
```

- ___ c. The class of the exception in question is:

com.ibm.trade.exception.TradeBaseException



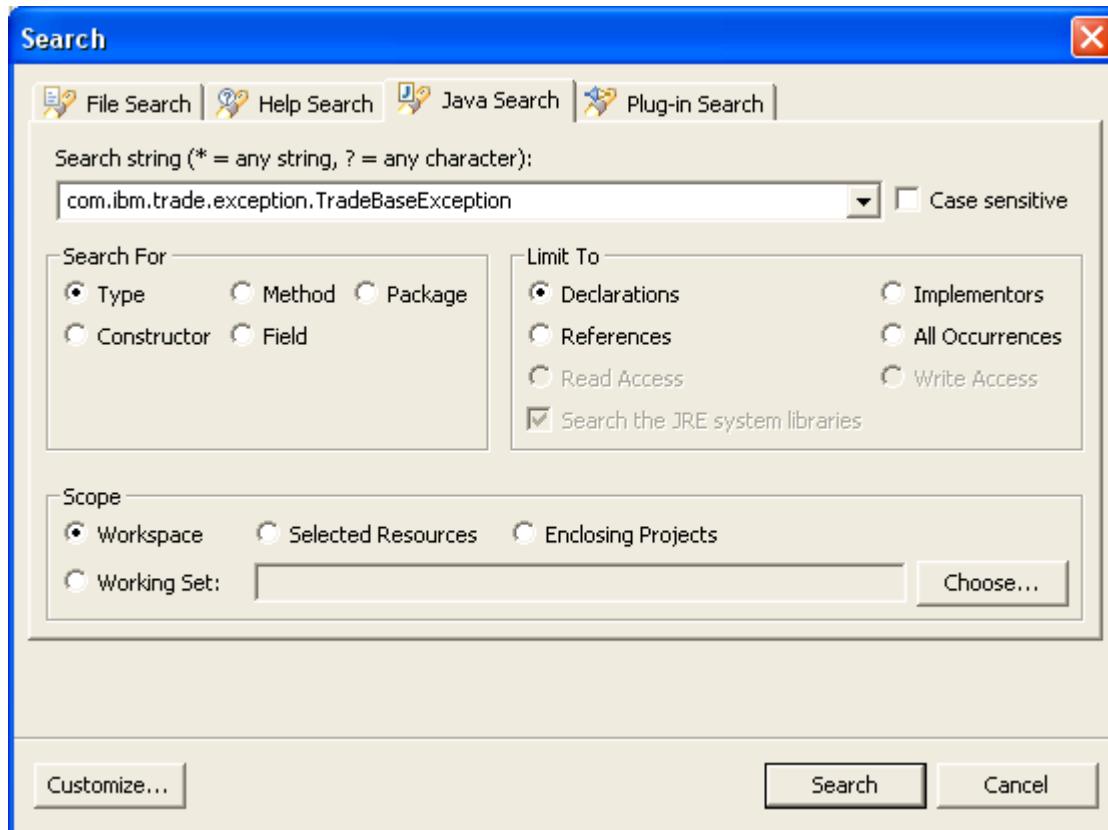
The screenshot shows the 'Web Deployment Descriptor' tab selected in an IDE. The XML code displays the configuration for handling exceptions in a web application. A specific line of code is highlighted: <exception-type>com.ibm.trade.exception.TradeBaseException</exception-type>. This line defines that if a TradeBaseException occurs, it should be handled by the error.jsp page.

```
</error-page>
<error-page>
    <exception-type>javax.servlet.ServletException</exception-type>
    <location>/error.jsp</location>
</error-page>
<error-page>
    <exception-type>com.ibm.trade.exception.TradeBaseException</exception-type>
    <location>/error.jsp</location>
</error-page>
<jsp-config>
    <taglib>
        <taglib-uri>http://java.sun.com/jstl/core</taglib-uri>
        <taglib-location>/WEB-INF/lib/standard.jar</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>http://jakarta.apache.org/taglibs/application-1.0</taglib-uri>
        <taglib-location>/WEB-INF/lib/taglibs-application.jar</taglib-location>
    </taglib>
    <taglib>
        <taglib-uri>http://jakarta.apache.org/taglibs/page-1.0</taglib-uri>
    </taglib>
</jsp-config>
```

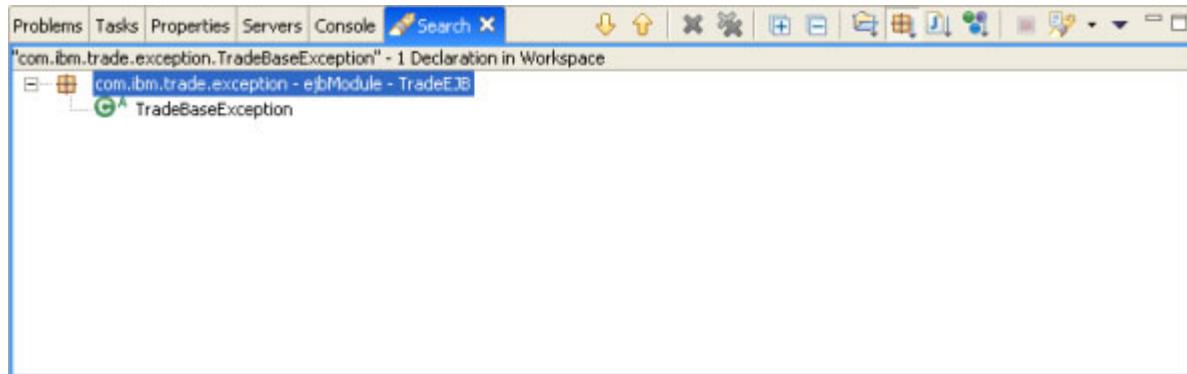
- ___ 2. Determine if the exception exists in the workspace.

- ___ a. Copy the string containing the class name from the deployment descriptor editor.
___ b. From the main menu click **Search —> Search**.

- ___ c. On the **Search** dialog select the **Java Search** tab. Paste the exception's class name on the **Search string** entry field. Select **Declarations** from the **Limit To** group box. Click **Search**.



The search operation reveals that the class declaration was found within the workspace (the modules currently loaded).



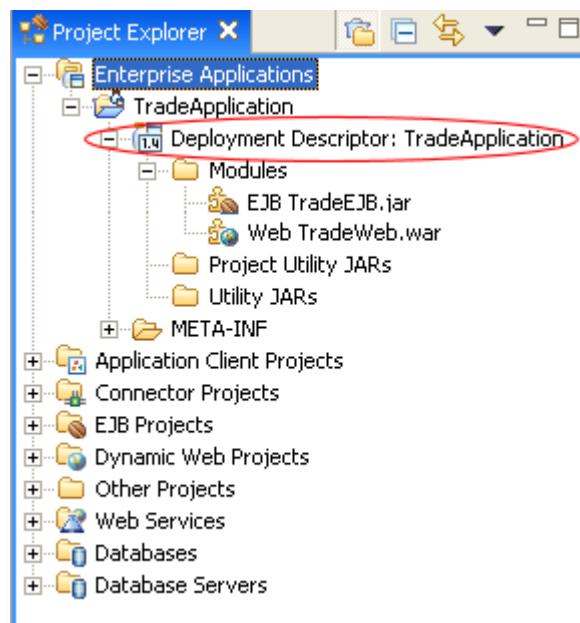
- ___ 3. Many times, just correcting the build path does not clear the task immediately and cleaning the project is necessary to get everything back in sync. Clean all the projects
- ___ a. From the main menu select **Project** —> **Clean**
- ___ b. Click **OK**.
- ___ c. Verify, that after cleaning, the Problems view does not contain this warning.

Configure WebSphere data sources

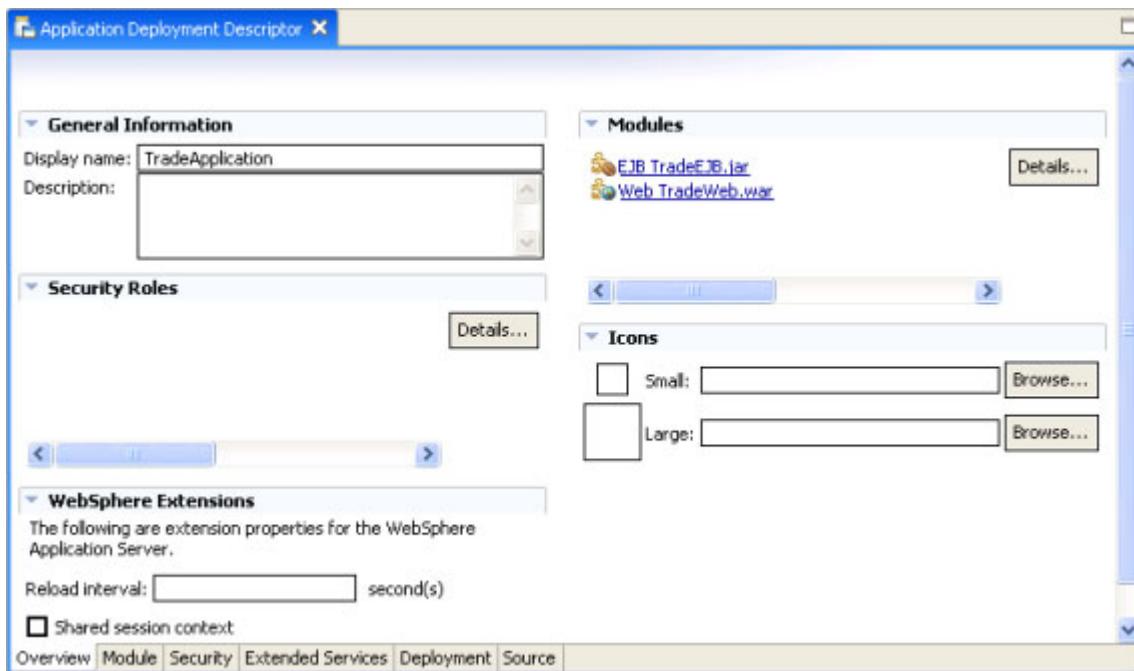
On the Deployment tab of the deployment descriptor editor for the application you can define certain resources that are included within the EAR file. Any resources defined on this page are defined at the new Application scope. You can use AST or wsadmin to view or modify any application scope resources. Application scope resources cannot be configured from the WebSphere administrative console.

In this section you will define a JDBC provider and then define the Data Source required by the Trade application. The JDBC resources will now be defined at the application scope and visible to the Trade application.

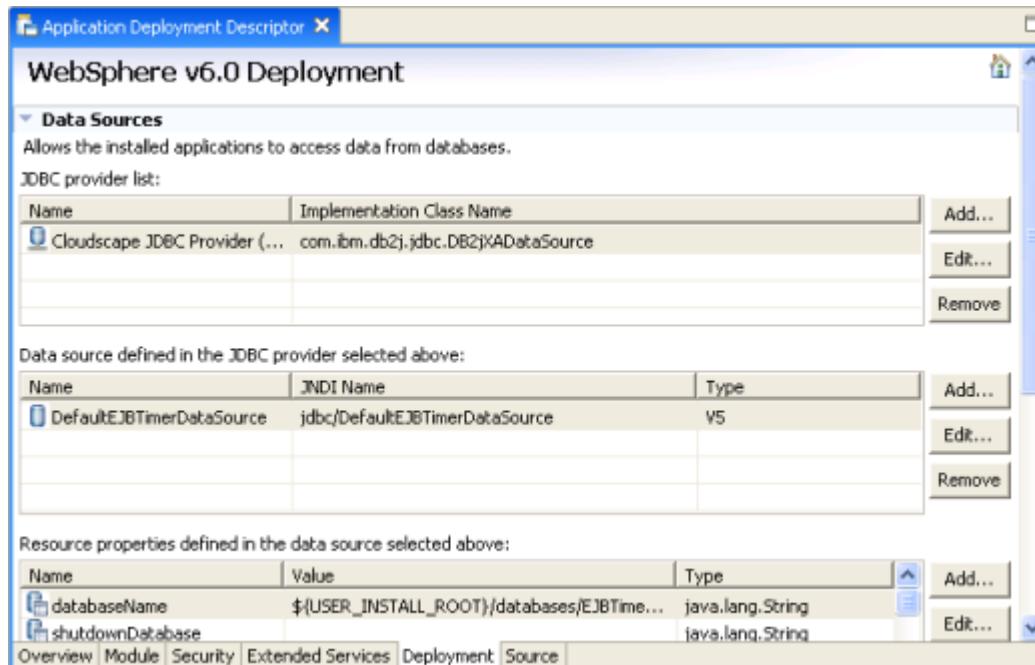
1. Expand Enterprise Application and the Trade Application. Double-click **Deployment Descriptor: Trade Application** to open the Trade Application deployment descriptor editor.



2. The deployment descriptor for the application's ear file opens in its specialized editor located on the upper right hand side of the perspective. Select the **Deployment** tab to view the current data sources.

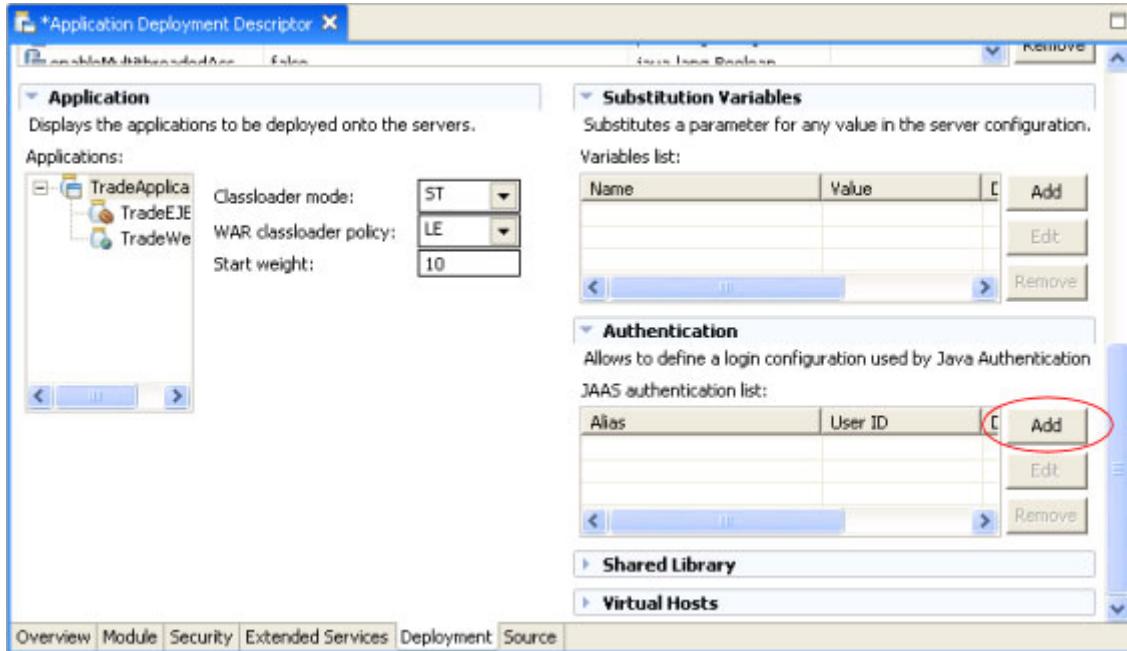


3. Configure the data sources for the application. Remove the default Cloudscape data source.



- a. Select the **Cloudscape JDBC provider** in the JDBC provider list and click **Remove**. The JDBC Providers list will now be empty.

4. Create a new J2C Authentication Data Entry with UserID **db2admin** and Password **was1edu**.
- a. From the application deployment descriptor select **Add** next to Authentication to add a JAAS authentication alias. You may need to scroll to the bottom and expand **Authentication**.



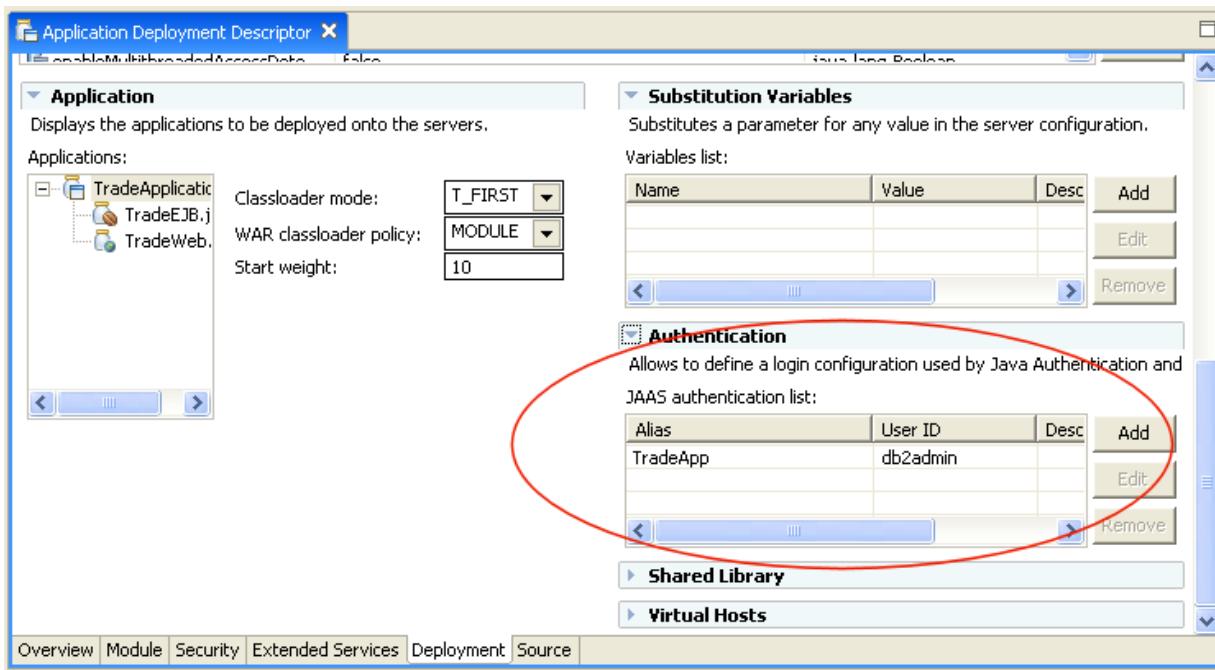
- b. Add a new alias with the following properties:

Field Name	Value
Alias	TradeApp
User id	db2admin
Password	was1edu
Description	Authentication for Trade Application

- c. Click **OK**.

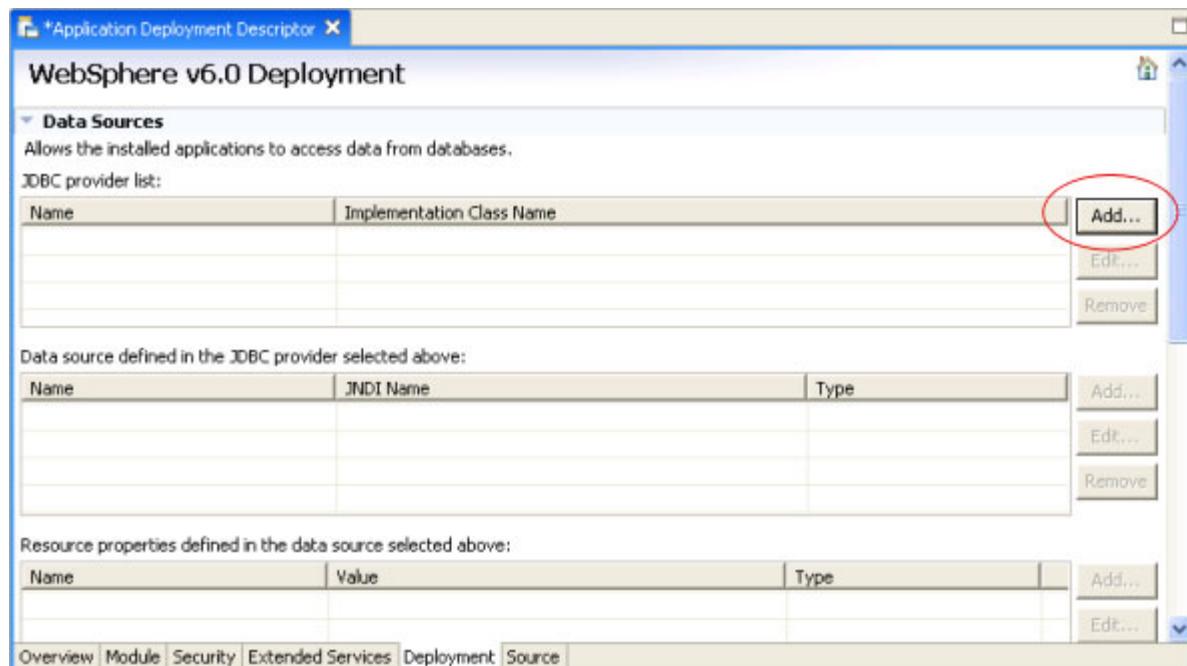


- ___ d. You should now see the TradeApp authentication alias is added to the JAAS authentication list.

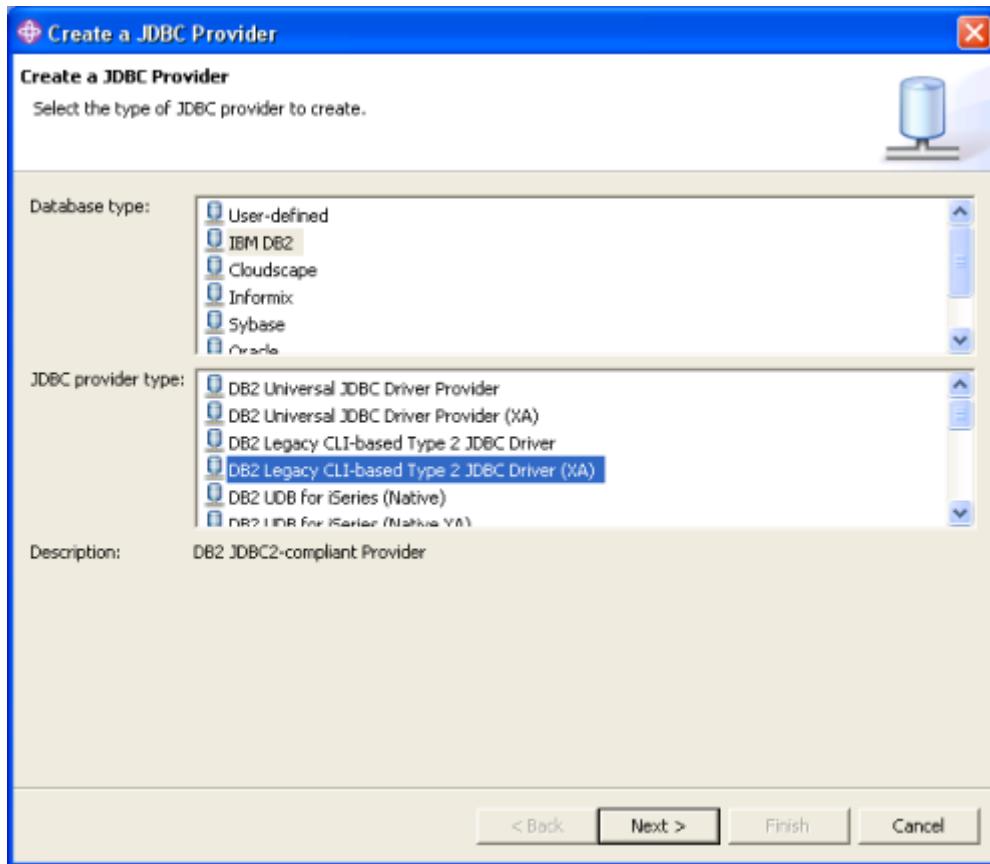


___ 5. Create a JDBC Provider for DB2.

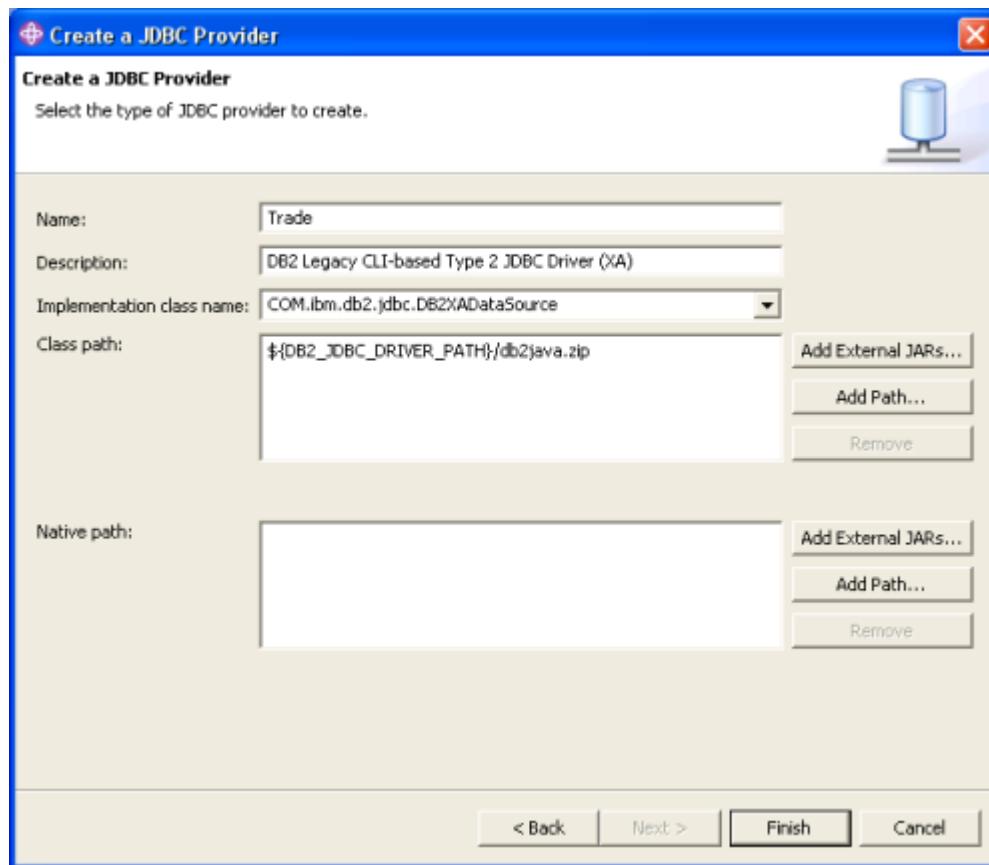
- ___ a. Click **Add** by the **JDBC provider list**. You may need to scroll to the top of the Application Deployment Descriptor.



- ___ b. In the Create a JDBC Provider window, select **IBM DB2** as the Database type. Select **DB2 Legacy CLI-based Type 2 JDBC Driver (XA)** as the JDBC provider type. Click **Next**.

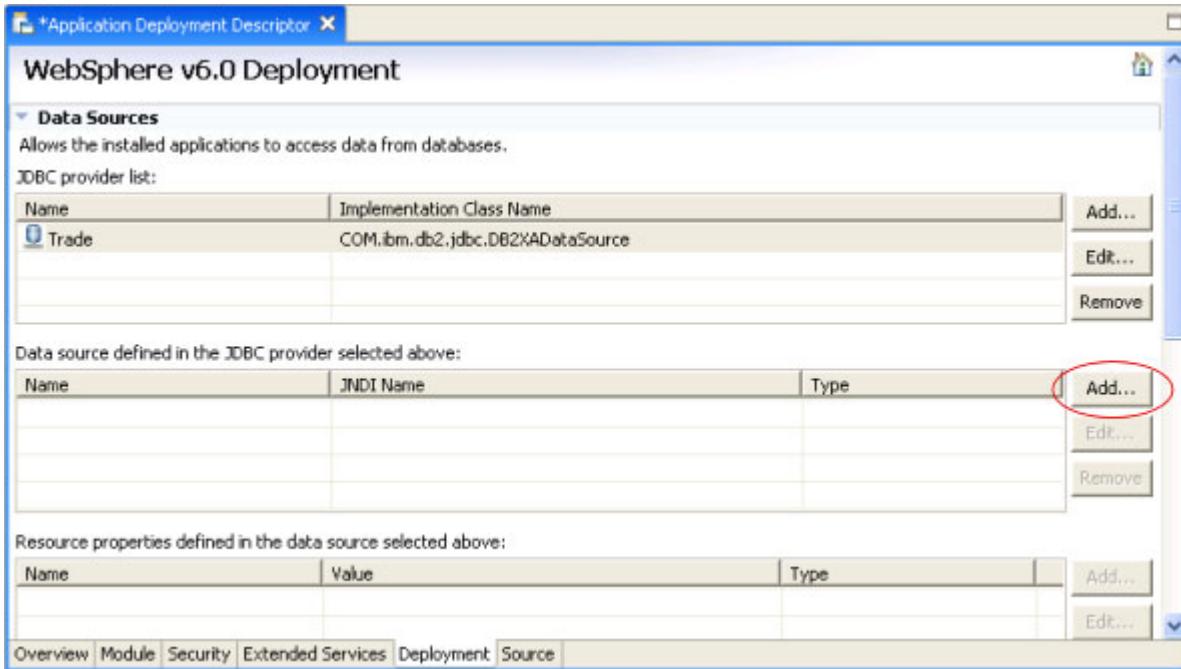


- ___ c. In the next window specify **Trade** for the name of the provider.

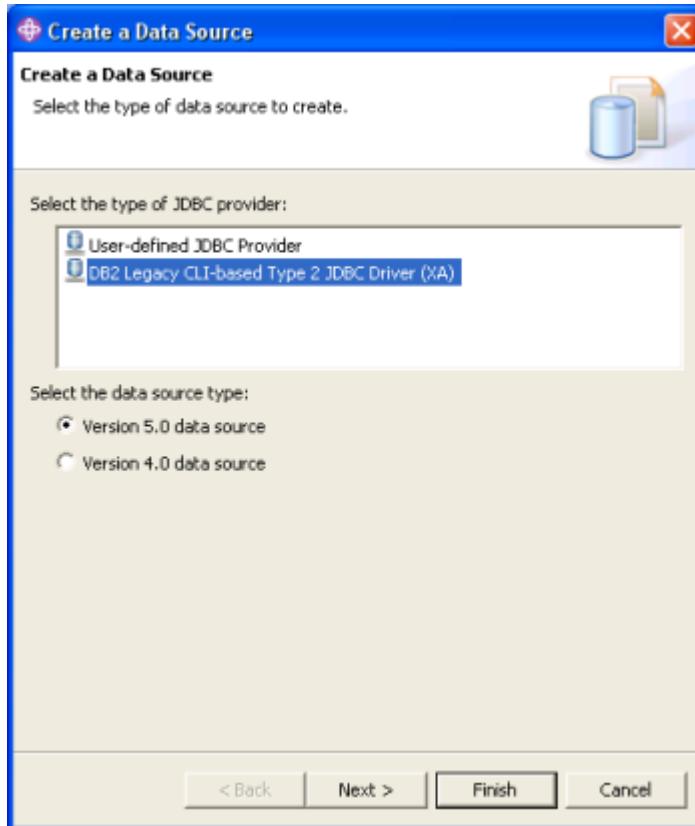


- ___ d. Click **Finish**. Verify that **Trade** has been added to the JDBC provider list.
___ 6. Next create the data source.

- __ a. Click **Add** to create a data source.

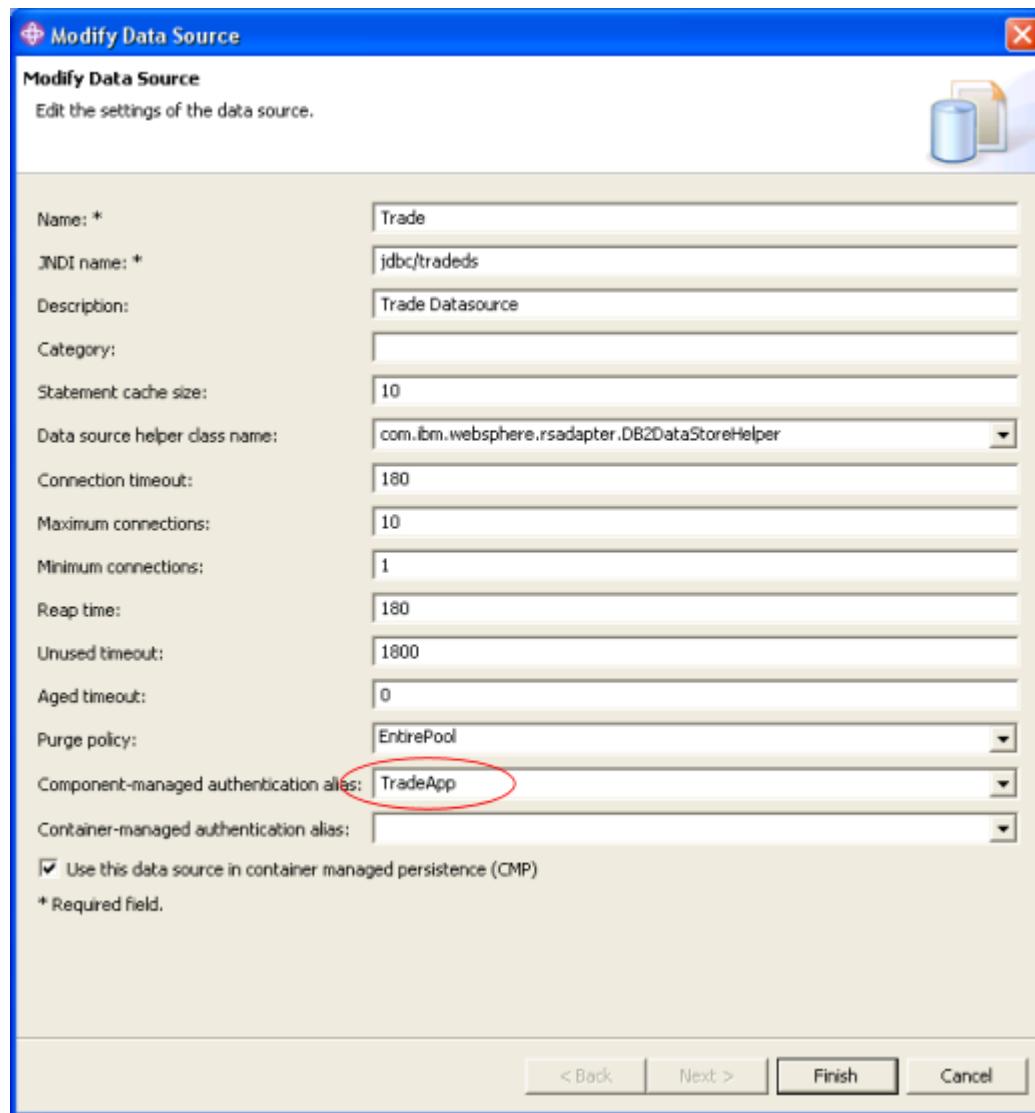


- __ b. In the Create a Data Source window select **DB2 Legacy CLI-based Type 2 JDBC Driver (XA)** as the JDBC provider. Make sure that the Version 5.0 data source option is selected. Click **Next**.



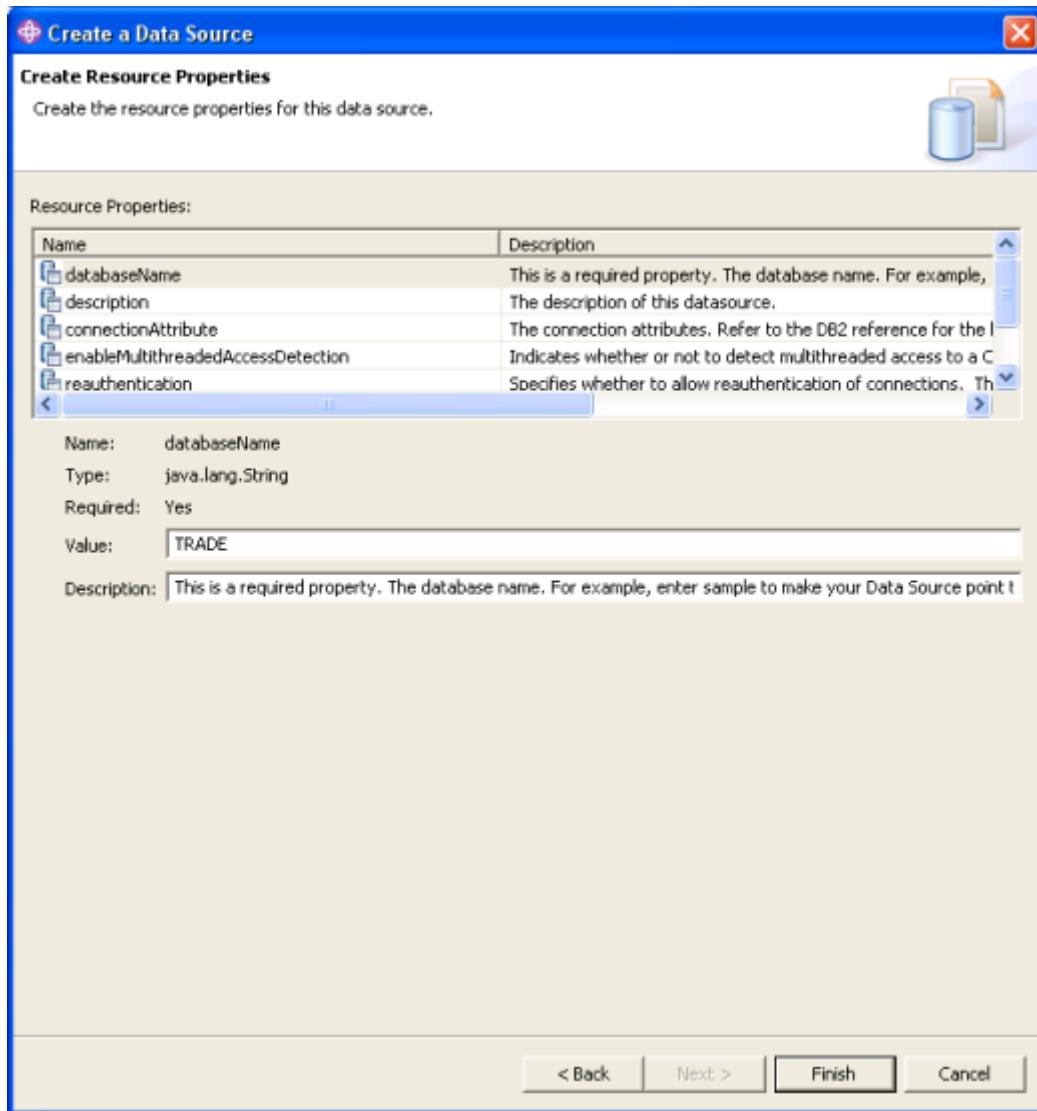
- ___ c. On the next page enter the properties of the data source. Enter the following information in the fields as provided. Leave default values for all other fields.

Field Name	Value
Name	Trade
JNDI Name	jdbc/tradeds
Description	Trade Datasource
Component-managed authentication alias	select: TradeApp
Container managed persistence	Checked



- ___ d. Click **Next**.

- ___ e. Select **databaseName** under the Resource Properties. For the **Value** specify **TRADE**. Click **Finish**.



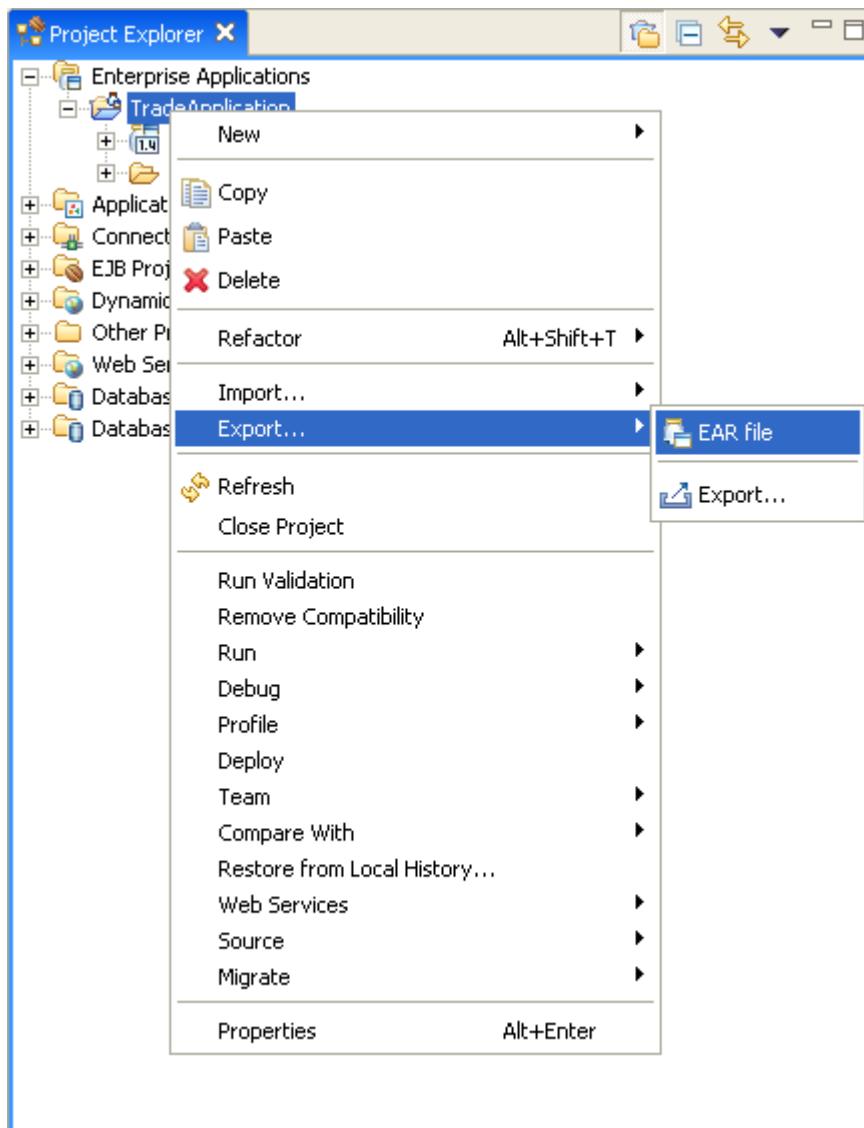
- ___ 7. Save your changes. From the main menu select **File** —> **Save** (or press **Ctrl-S**).

Information: These resource settings are saved with the EAR file in a file called **deployment.xml**. This file is not part of the J2EE 1.4 specification but is recognized by WebSphere Application Server V6. The Enterprise Archive containing this file is called an enhanced EAR file.

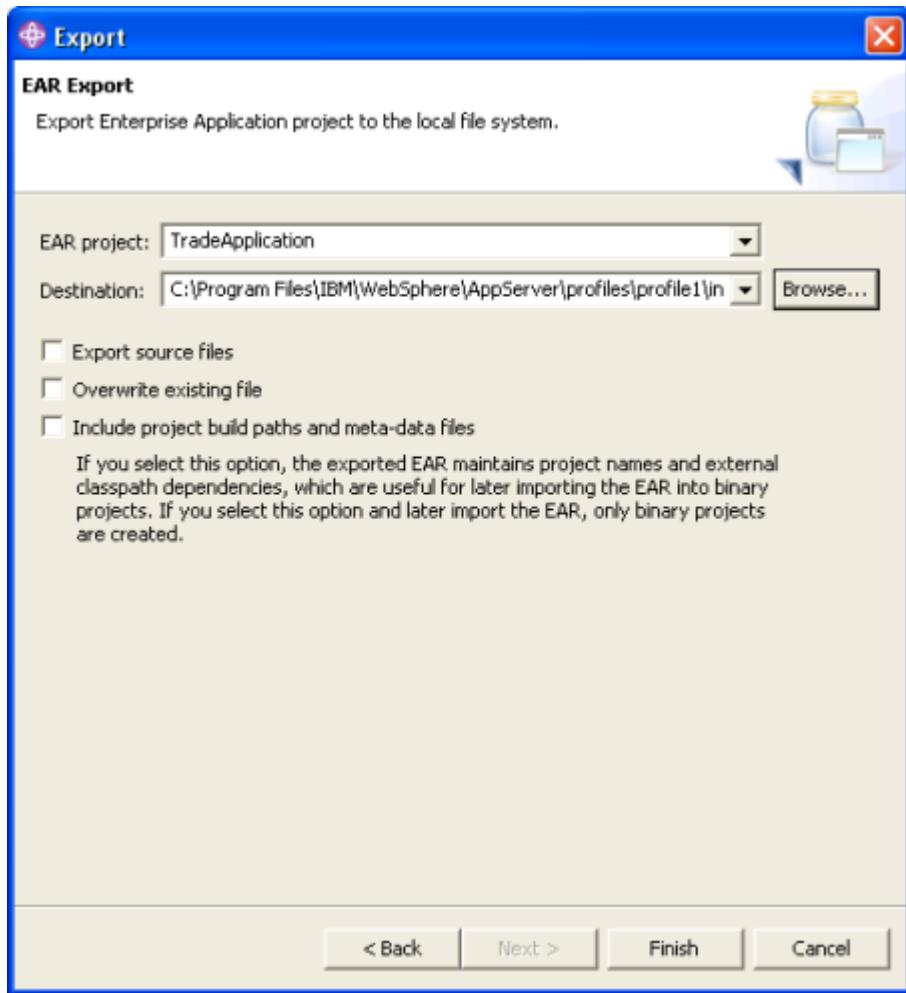
Save the Enterprise Application Archive (.ear) file

Save the file in the <profile_root>\profile1\installableApps directory.

- 1. In the J2EE Hierarchy Project Explorer view on the upper left pane, right-click **TradeApplication** —> **Export** —> **EAR file**.



- __ 2. On the **Export** dialog, click **Browse** and navigate to the folder <profile_root>\profile1\installableApps and type in the file name: **TradeApplication.ear**. Click **Save**.



- __ 3. Click **Finish** to export the EAR file.
__ 4. Verify that the ear file was saved successfully.
__ 5. Exit the AST tool.

End of lab

Exercise review and wrap-up

In this lab, the Application Server Toolkit was opened and used to assemble the modules for the Trade Application Case Study application into an Enterprise Application Archive.

Exercise 5. Installing the Trade Application

What this exercise is about

This lab covers the tasks used to install an Enterprise Application in WebSphere Application Server using the WebSphere Application Server administrative console. The Trade Enterprise Application Archive (.ear) file packaged as an enhanced EAR file will be installed.

What you should be able to do

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

- Use the administrative console to install an Enterprise Archive (EAR) in WebSphere Application Server V6
- Test the application installation using a browser

Introduction

In this exercise, you will install the Enterprise Application (.ear) file that was assembled using the Assembly Toolkit (AST) as an enhanced EAR file. The EAR file contains all the application modules, as any other EAR file would, and also contains the definition of other resources required by the application. The application will be tested by accessing it from a Web browser.

Required materials

To perform this exercise, you must have WebSphere Application Server installed on the machine and a working application server with an administrative console. In addition, you will need a browser and DB2 installed and the Trade Application Case Study databases created and populated.

You will also need the **TradeApplication.ear** file.

Instructor exercise overview

The focus of this exercise is to introduce the student to the wizards in the administrative console. The student will use a wizard to install the application and a browser to test that application.

The students will complete the following tasks:

- Copy the TradeApplication.ear file to <profile_root>\profile1\installableApps (if the student has completed the Assembling the Trade Application lab, this file will already be in place).
- Use the wizard to install the application.
- Start the application.
- Finally, the students will open a browser and test the application. If they've successfully installed the application, they should get the trade application's welcome page to log in to the system. At this stage, the student should be able to log in with userid of **client**, and look at the account information and the holdings for the user. They should also be able to register as a new user and log in using the new ID.

Note: If the student's application is damaged beyond repair at this point, you can have them start over by copying the TradeApplication.ear file from the <software>\Solutions directory to the InstallableApps directory and redo the exercise.

Following this exercise the Trade application has been installed and the necessary resources created using the console.

To complete this lab quickly you must do the following:

- Be certain that the install lab was complete.
- Copy TradeApplication.ear to the <profile_root>\profile1\installableApps directory.
- Test.

Exercise instructions

Start the server and administrative console

Use the WebSphere Application Server administrative console to install the Trade Application. Since the console is a Web application running on the server, the server has to be up and running before the console is started.

- ___ 1. Start WebSphere Application Server if it is not already running.
 - ___ a. A server can also be started from the <was_root> directory. When using the <was_root> directory the profile name must also be specified. In a command prompt navigate to <was_root>\bin.
 - ___ b. Enter the **startserver server1 -profileName profile1** command.
 - ___ c. The server has started when a message similar to the following appears.

Server server1 open for e-business; process id is 2136

Information: The process id is a unique number which represents the server process that is running on system.

UNIX:

- ___ a. Navigate to the <was_root>/bin directory and enter the **./startServer.sh server1 -profileName profile1** command.
- ___ 2. Open the administrative console for the application server.
 - ___ a. In a Web browser, specify the address **http://localhost:9060/ibm/console**.
 - ___ b. Enter **admin** for the userid and click **OK**.

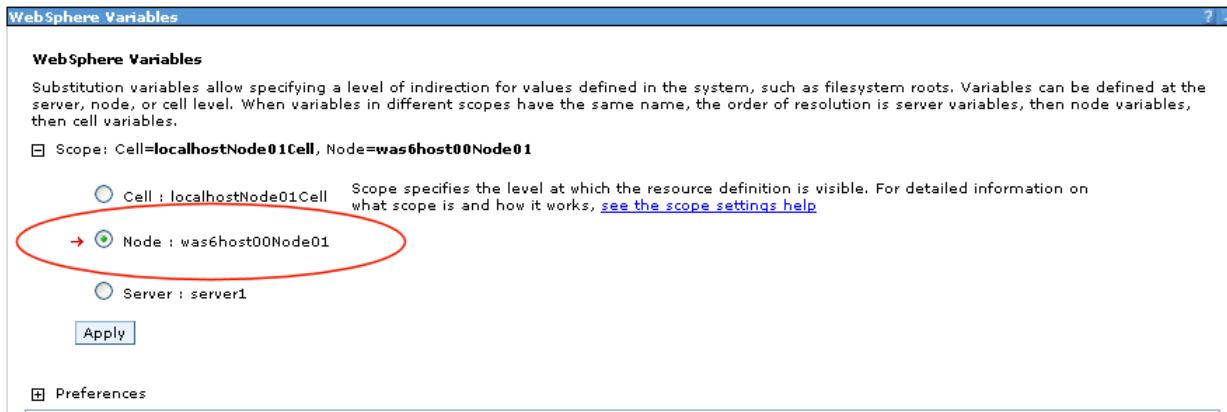
Information: Because security is not enabled, you can enter anything you like here. The user ID you enter will only be used to keep track of who has made changes to the configuration.

Set an environment variable

To ensure your configuration information is not machine specific, set an environment variable to the directory where the DB2 JDBC driver (**db2java.zip**) is installed on your node.

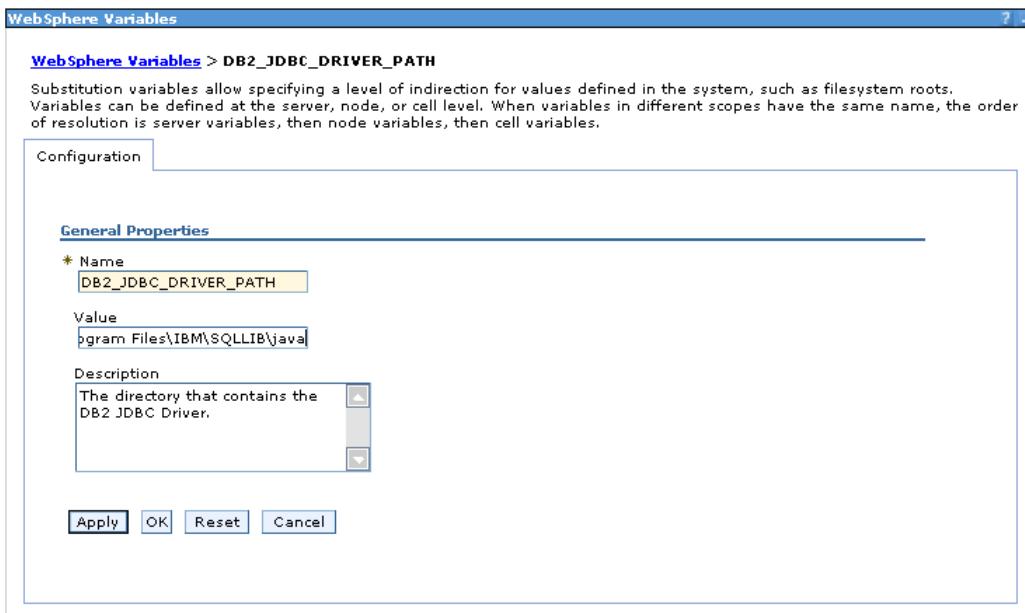
- ___ 1. Set the **DB2_JDBC_DRIVER_PATH** environment variable to point to the directory where **db2java.zip** is located.
 - ___ a. Expand **Environment** in the left navigation, and select **WebSphere Variables**.

- ___ b. In the workspace on the right, ensure Scope is set to Node.



- ___ c. Scroll down to **DB2_JDBC_DRIVER_PATH**, and click it.
 ___ d. A new page will be displayed allowing you to assign a value to this variable. In the **Value** field, enter the directory where **db2java.zip** is located.

Enter <db2>\java.



Information: Note that WebSphere Application Server, on Windows, accepts both forward and backward slashes, and might actually change what you enter to forward slashes.

- ___ e. Click **OK**.
 ___ 2. You will see a message saying that your changes need to be saved, and the server may need to be restarted. Save the changes you have made so far.

UNIX: Verify that two environment variables are set on the machine. Open a command window and enter the following commands:

```
echo $DB2INSTANCE
echo $LD_LIBRARY_PATH (on Linux and Sun Solaris)
echo $LIBPATH (on AIX)
```

When using the Bash shell, if the environmental variables are not set, edit the **.bashrc** file for root. When using the ksh shell, if the environment variables are not set, edit the **.profile** file for root. Open **/root/.bashrc** and add the following four lines to the file:

```
#Setup DB2 environment for root user.
if [ -f /home/db2inst1/sqllib/db2profile ] ; then
  . /home/db2inst1/sqllib/db2profile
fi
```

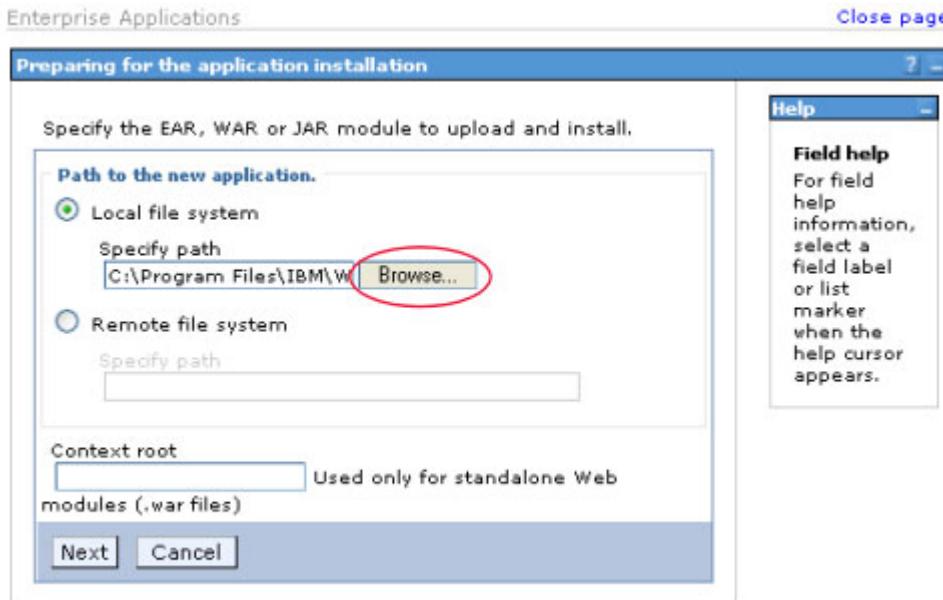
Save the file when completed.

Install the Enterprise Application

Install and test the Trade Enterprise Application. The TradeApplication EAR is an enhanced EAR file that contains definition of other resources required by the application. At this point the Trade Application should be attempting to get quotes off a public internet quote Web service.

- ___ 1. Install Trade Enterprise Application to the application server, **server1**.
 - ___ a. Expand **Applications** in the left navigation, and select **Install New Application**.
 - ___ b. In the workspace the Preparing for application install page will be displayed. Select **Local file system** and click **Browse**. Navigate to the **<profile_root>\profile1\installableApps** directory where the **TradeApplication.ear** was saved in the previous exercise. Select it and click

Open. The **Local file system** field should now contain the path of the EAR file.



Click Next.

- ___ c. At this point WebSphere is expanding and examining the EAR file in memory. Depending on the system this could take a little while. The next pane to appear gives you the option to generate default bindings and mappings. At this time you will not be changing any bindings. Scroll down and click **Next**.
- ___ d. The next page will show *Step 1: Select installation options*. Scroll down the page, and notice that there are twelve steps to complete. You can complete each step

in turn and press **Next** to navigate to the next step, or you can jump directly to a step by clicking on its link.

Enterprise Applications

Install New Application

Specify options for installing enterprise applications and modules.

→ **Step 1: Select installation options**

- [Step 2 Map modules to servers](#)
- [Step 3 Select current backend ID.](#)
- [Step 4 Provide JNDI Names for Beans](#)
- [Step 5 Provide default data source mapping for modules containing 2.x entity beans](#)
- [Step 6 Map data sources for all 2.x CMP beans](#)
- [Step 7 Map EJB references to beans](#)
- [Step 8 Map resource references to resources](#)
- [Step 9 Map virtual hosts for Web modules](#)
- [Step 10 Map security roles to users/groups](#)
- [Step 11 Ensure all unprotected 2.x methods have the correct level of protection](#)
- [Step 12 Summary](#)

Select installation options

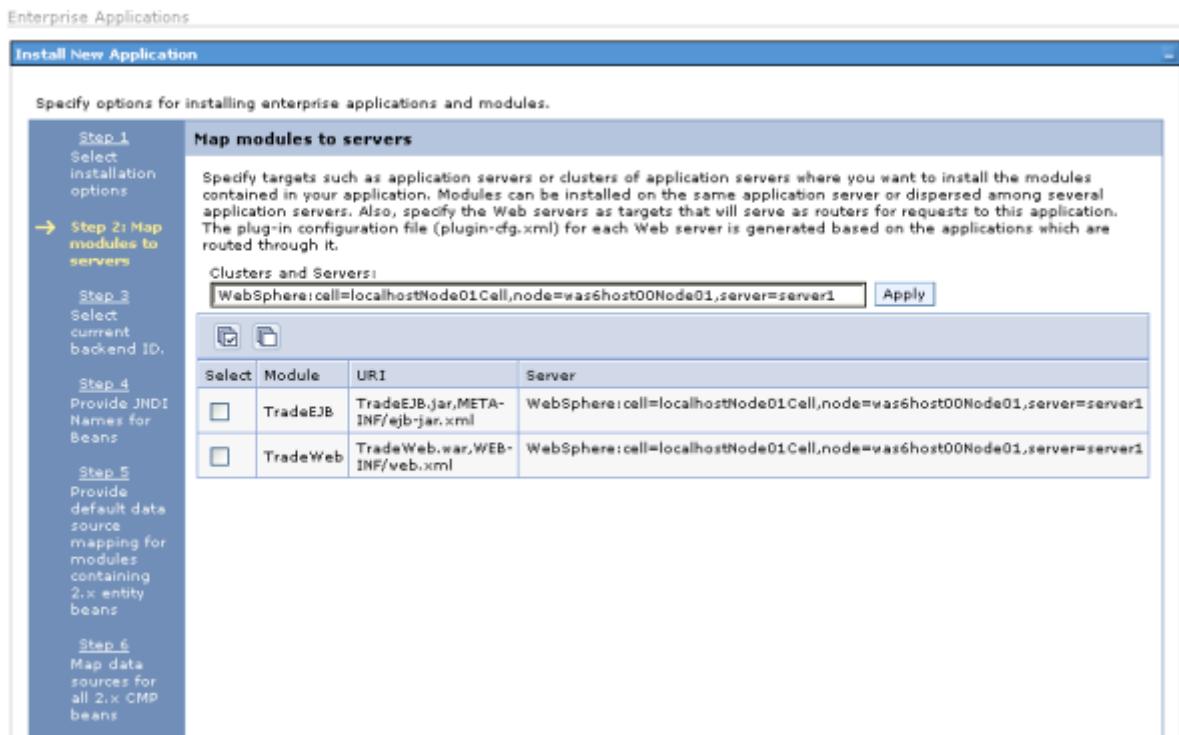
Specify the various options that are available to prepare and install your application.

Pre-compile JSP
 Distribute application
 Use Binary Configuration
 Deploy enterprise beans
Application name: TradeApplication
 Create MBeans for resources
 Enable class reloading
Reload interval in seconds:
 Deploy Web services
Validate Input off/warn/fail: warn
 Process embedded configuration

Next **Cancel**

- ___ e. In Step 1, select the check box to enable **Pre-compile JSP**. This operation takes a little while to complete, but is an excellent option to select when testing. This way you can ensure all JSPs will compile. In production, this is a way to reduce the response time for the first request of a JSP. Notice there is also an option to deploy EJBs. This would regenerate deploy code for the EJBs in the enterprise application. The developer deployed the EJBs in IBM Rational Application Developer before it was exported, so this step is not necessary. Click **Next**.
- ___ f. Step 2 will now be displayed. Notice that Step 1 is still listed as a hyperlink, allowing you to return to any previous step. Step 2 allows you to map modules to servers which allows you to specify the particular application server to install the enterprise application to. In this case there will only be one server, **server1**.

Notice that under the Server heading in the table, both the EJB and Web module will be installed to this server.



- ___ g. Step 3 allows you to select the current back-end ID. Click Next to get to Step 3. Step 4 allows you to set JNDI names for each beans in the enterprise application. Verify that the JNDI names are as you would expect.
- ___ h. Step 5 allows you to specify a default Data Sources for all EJB 2.0 CMP beans. All the EJBs in the application have their own individual data source defined, so you will not define a default data source.
- ___ i. Continue moving forward and observe the kinds of settings available. You should not need to change any of the values.

- j. Jump directly to the Summary page by clicking on the Step xx link next to Summary. This displays a summary of the options you have selected. Verify this information, then click Finish..

Enterprise Applications

Install New Application

Specify options for installing enterprise applications and modules.

Step 1 Select installation options	Summary																														
Step 2 Map modules to servers	Summary of installation options																														
Step 3 Select current backend ID.	<table border="1"> <thead> <tr> <th>Options</th> <th>Values</th> </tr> </thead> <tbody> <tr> <td>Use Binary Configuration</td> <td>No</td> </tr> <tr> <td>Create MBeans for resources</td> <td>Yes</td> </tr> <tr> <td>Cell/Node/Server</td> <td>Click here</td> </tr> <tr> <td>Reload interval in seconds</td> <td></td> </tr> <tr> <td>Enable class reloading</td> <td>No</td> </tr> <tr> <td>Process embedded configuration</td> <td>Yes</td> </tr> <tr> <td>Application name</td> <td>TradeApplication</td> </tr> <tr> <td>Validate Input off/warn/fail</td> <td>warn</td> </tr> <tr> <td>Application Scoped Resources</td> <td>Yes</td> </tr> <tr> <td>Directory to install application</td> <td></td> </tr> <tr> <td>Distribute application</td> <td>Yes</td> </tr> <tr> <td>Deploy Web services</td> <td>No</td> </tr> <tr> <td>Pre-compile JSP</td> <td>No</td> </tr> <tr> <td>Deploy enterprise beans</td> <td>No</td> </tr> </tbody> </table>	Options	Values	Use Binary Configuration	No	Create MBeans for resources	Yes	Cell/Node/Server	Click here	Reload interval in seconds		Enable class reloading	No	Process embedded configuration	Yes	Application name	TradeApplication	Validate Input off/warn/fail	warn	Application Scoped Resources	Yes	Directory to install application		Distribute application	Yes	Deploy Web services	No	Pre-compile JSP	No	Deploy enterprise beans	No
Options	Values																														
Use Binary Configuration	No																														
Create MBeans for resources	Yes																														
Cell/Node/Server	Click here																														
Reload interval in seconds																															
Enable class reloading	No																														
Process embedded configuration	Yes																														
Application name	TradeApplication																														
Validate Input off/warn/fail	warn																														
Application Scoped Resources	Yes																														
Directory to install application																															
Distribute application	Yes																														
Deploy Web services	No																														
Pre-compile JSP	No																														
Deploy enterprise beans	No																														
Step 4 Provide JNDI Names for Beans																															
Step 5 Provide default data source mapping for modules containing 2.x entity beans																															
Step 6 Map data sources for all 2.x CMP beans																															
Step 7 Map EJB references to beans																															
Step 8 Map resource references to resources																															
Step 9 Map virtual hosts for Web modules																															
Step 10 Map security roles to users/groups																															
Step 11 Ensure all unprotected 2.x methods have the correct level of protection																															
→ Step 12: Summary																															
<input type="button" value="Previous"/> <input type="button" value="Finish"/> <input type="button" value="Cancel"/>																															

- ___ k. You should see a page headed **Installing**, and messages will scroll down to indicate the progress of the installation. The application server will step through each module, performing installation tasks on each one. Look for the following message to confirm the enterprise application was successfully installed.

ADMA0115W: Resource Assignment of name jms/tradeproccf and type javax.jms.ConnectionFactory, with JNDI name jms/TradeLstrJMSCF is not found within scope of module TradeEJB with URI TradeEJB.jar,META-INF/ejb-jar.xml deployed to target WebSphere:cell=localhostNode01Cell,node=was6host00Node01,server=server1.

ADMA5068I: The resource validation for application TradeApplication completed successfully, but warnings occurred during validation.

ADMA5058I: Application and module versions validated with versions of deployment targets.

ADMA5005I: The application TradeApplication is configured in the WebSphere Application Server repository.

ADMA5053I: The library references for the installed optional package are created.

ADMA5005I: The application TradeApplication is configured in the WebSphere Application Server repository.

ADMA5001I: The application binaries are saved in C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\twstemp\92668751\workspace\cells\localhostNode01Cell\applications\TradeApplication.ear\TradeApplication.ear

ADMA5005I: The application TradeApplication is configured in the WebSphere Application Server repository.

SECJ0400I: Successfully updated the application TradeApplication with the appContextIDForSecurity information.

ADMA5011I: The cleanup of the temp directory for application TradeApplication is complete.

ADMA5013I: Application TradeApplication installed successfully.

Application TradeApplication installed successfully.

To start the application, first save changes to the master configuration.

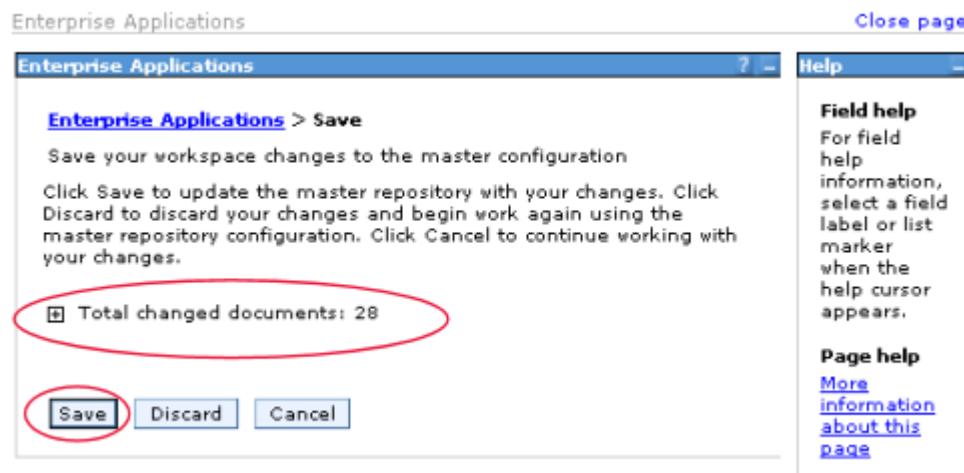
Save to Master Configuration

To work with installed applications, click the "Manage Applications" button.

- ___ 2. Save the application configuration changes to the master configuration.

- ___ a. Below the success message, click the **Save to Master Configuration** link.

- ___ b. A Save page will appear. Expand the plus in front of **Total changed documents**. This will provide a list of all the XML files being updated. From the Save page, click **Save** to save to the master configuration.



Start the Enterprise Application

Before you can test the application you must start the TradeApplication.

- ___ 1. Start the Trade enterprise application.
- ___ a. Expand **Applications** in the left navigation if needed, and select **Enterprise Applications**. The page displayed in the workspace lists the enterprise applications installed and lists whether they are started or stopped.

Enterprise Applications	
Lists installed applications. A single application can be deployed onto multiple servers.	
<input type="checkbox"/> Preferences	
	Start Stop Install Uninstall Update Rollout Update Remove File Export Export DDL
<input type="checkbox"/>	
Select	Name
<input type="checkbox"/>	DefaultApplication
<input type="checkbox"/>	TradeApplication
<input type="checkbox"/>	iwbApp
<input type="checkbox"/>	query
Total 4	

- ___ b. Find **TradeApplication** and select it by adding a check to the check box to the left.

- ___ c. Click **Start**. This will attempt to start every enterprise application that has been checked; in this case it will attempt to start the TradeApplication.
- ___ d. Once completed, look for a message at the top of the screen indicating the TradeApplication started successfully. Scroll down to the table, and you should also see that TradeApplication has a status of Started.

The screenshot shows the 'Enterprise Applications' interface. At the top, there is a 'Messages' panel containing a single message: 'Application TradeApplication on server server1 and node was6host00Node01 started successfully.' Below this is a section titled 'Enterprise Applications' with the sub-instruction: 'Lists installed applications. A single application can be deployed onto multiple servers.' There is a 'Preferences' link. Below these are several buttons: Start, Stop, Install, Uninstall, Update, Rollout Update, Remove File, Export, and Export DDL. Underneath these buttons is a toolbar with icons for selecting, opening, saving, and other functions. The main area is a table with columns 'Select', 'Name', and 'Status'. The table contains four rows, each with a checkbox in the 'Select' column and a link in the 'Name' column: 'DefaultApplication', 'TradeApplication', 'ivtApp', and 'query'. All four entries have a green circular status icon in the 'Status' column. At the bottom of the table, it says 'Total 4'.

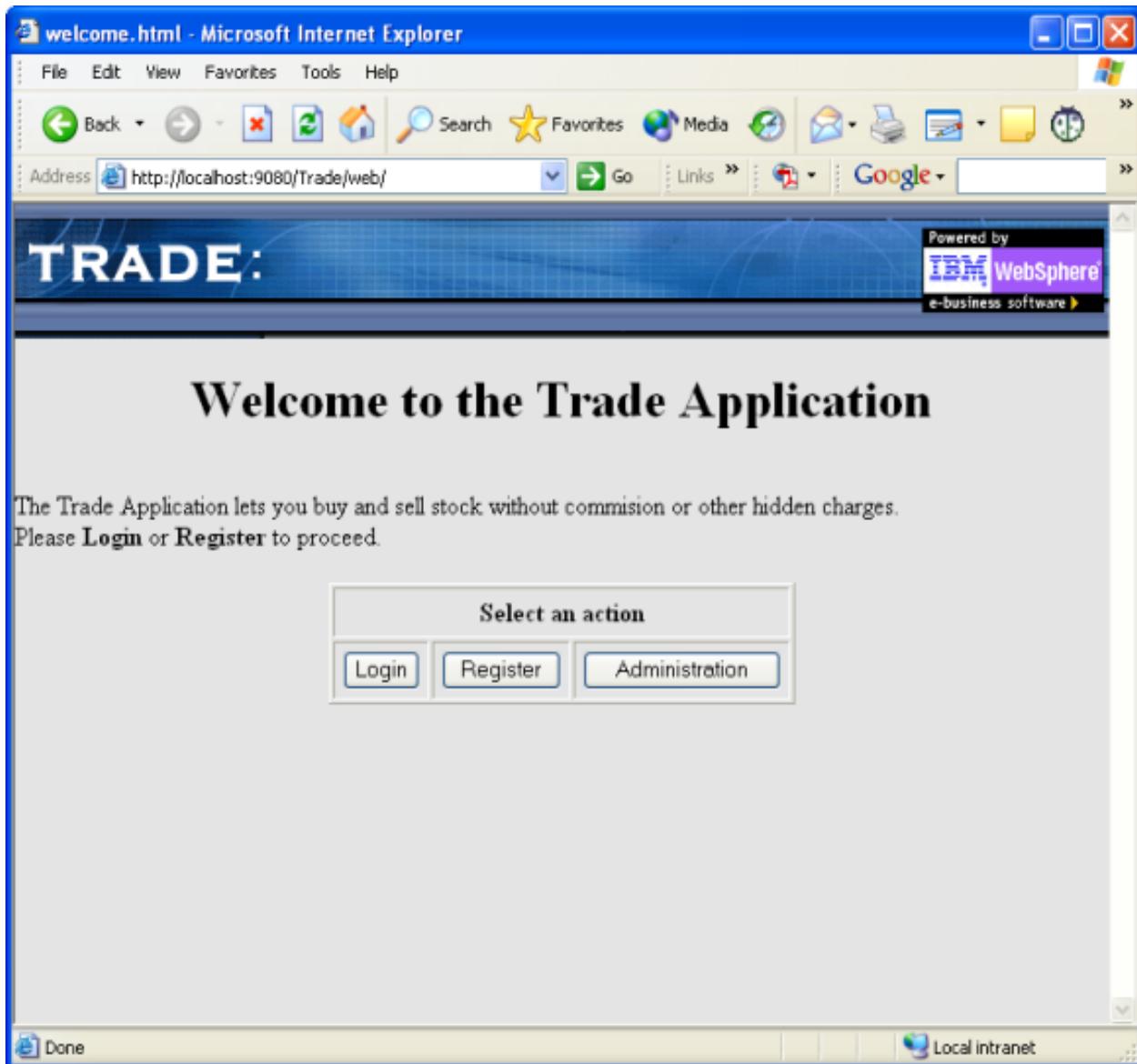
Select	Name	Status
<input type="checkbox"/>	DefaultApplication	
<input type="checkbox"/>	TradeApplication	
<input type="checkbox"/>	ivtApp	
<input type="checkbox"/>	query	

- ___ e. If an error is reported, use the command prompt to change to the <profile_root>\profile1\logs\server1 directory and display the contents of **SystemOut.log**. This should provide more information about the error. Verify that you followed all of the above steps correctly, and try restarting the server if you still experience problems.

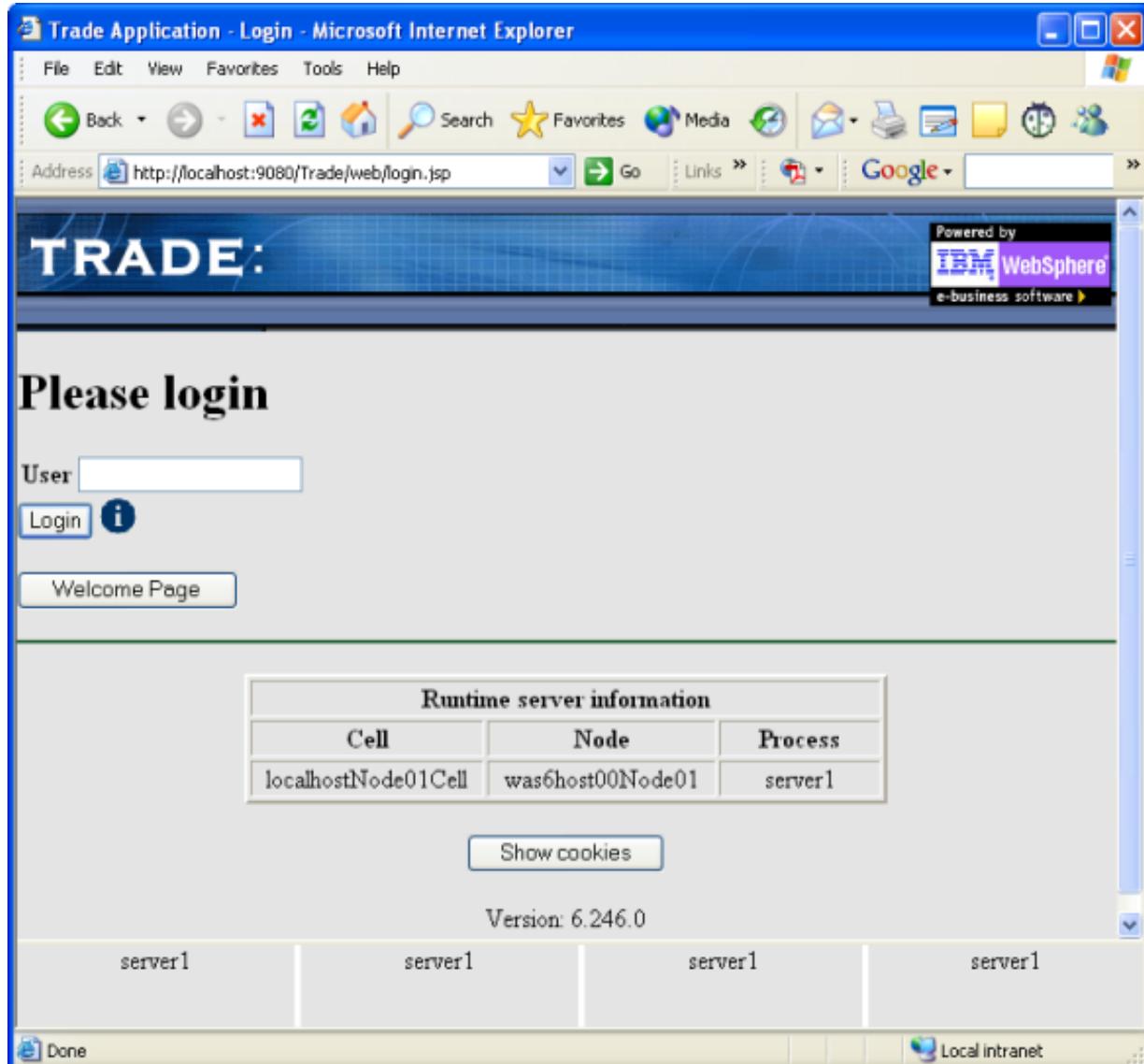
Test the Enterprise Application

Test the application by accessing it via the WebSphere Application Server's HTTP transport.

- ___ 1. Verify the TradeApplication.
 - ___ a. Open a new Web browser, and access the TradeApplication by entering the following address: **http://localhost:9080/Trade/web**.



- __ b. Click **Login**. The Trade login page (**login.jsp**) should be displayed.



A registered user can log in to access their account. You can get quotes for a number of stocks but the labs will primarily use IBM, AOL, CSCO, DELL, HWP, LU and MSFT.

Information: The Login page presents some additional information to you. This information is used for the purpose of education and trouble shooting, and is not really necessary to the application.

The **Runtime server information** shows the cell, node, and process that served the current page. This information is useful when running in a network deployment environment, and you need to know which cluster member serviced your request.

Show cookies is a hyperlink that opens a window and displays any cookies for this session. Useful to troubleshooting session affinity, seeing the ClonelIDs of the application servers, and so forth. Usually for advanced users only.

The section containing the four frames showing the server names is used in a network deployment environment to see how workload is being distributed among cluster members.

- ___ c. Log in to the application. Enter the user **client** and click **Login**.

- __ d. The Account Information page should be displayed.

The screenshot shows a Microsoft Internet Explorer window with the title "Account Information - Microsoft Internet Explorer". The address bar shows the URL <http://localhost:9080/Trade/web/AccountInfo>. The page itself has a blue header with the word "TRADE:" and a "Powered by IBM WebSphere e-business software" banner indicating "Version: 6.246.0". Below the header is a navigation menu with tabs: Welcome, Login, Account Information (which is selected), Holdings, and Transaction History. There are also "Get Quotes" and "i" buttons. The main content area is divided into two columns. The left column contains "Account Information" with fields: Name (Jessie James), AccountId (87652289), Email (jjames@us.ibm.com), Credit Card Number (555444907), and Balance (\$11,558.17). The right column contains "Preferences" with checkboxes for "Use Internet quotes" (selected) and "Broadcast transaction data". Below these are "Home Address" and "Business Address" sections. The "Home Address" is 97 Milton Rd., Pittsburgh, PA 15222, USA. The "Business Address" is 1050 Chum Ave., Markham, ON L5G 1K9, Canada. At the bottom of the page are "Update Account Page" and "i" buttons, and a "Local intranet" link in the status bar.

From this page you can get a stock quote, update account information, view the holdings and transaction history pages, and go back to the welcome and login pages.

Notice from the Account Information page under Preferences the option to Use Internet quotes (internet connection required) is selected. When the option is not selected quotes will come from a database. Until the Quote Web service is installed, leave this option selected.

- ___ e. Select the information icon, or “circle i” icon in the application which is available to give you additional information for the purpose of education or troubleshooting.

Account Information - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Favorites

Address Go Links

TRADE:

Powered by **IBM WebSphere**
e-business software >

Version: 6.246.0

Welcome [Login](#) Account Information Holdings Transaction History

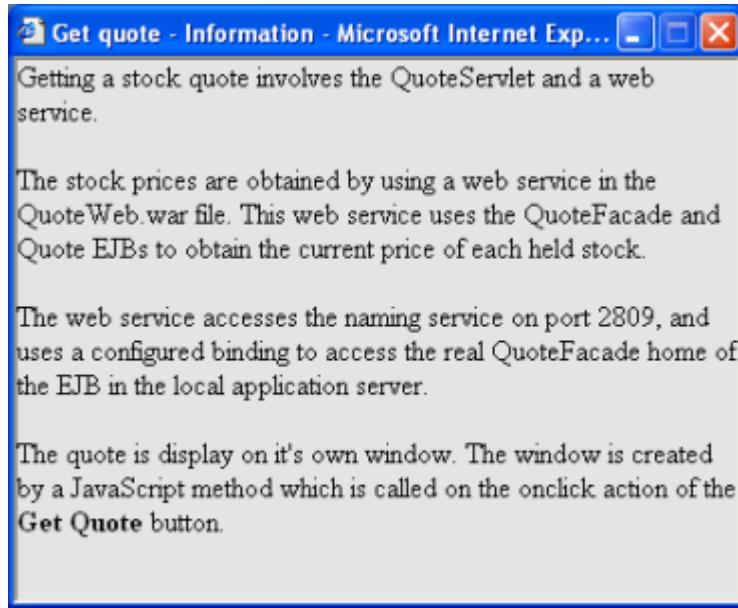
[Get Quotes](#)

Account Information <p>Name: Jessie James AccountId: 87652289 Email: jjames@us.ibm.com Credit Card Number: 555444907 Balance: \$11,558.17</p>	Preferences <p>Use Internet quotes: <input checked="" type="checkbox"/> Broadcast transaction data: <input type="checkbox"/></p>
Home Address <p>97 Milton Rd. Pittsburgh, PA 15222 USA</p>	Business Address <p>1050 Chum Ave. Markham, ON L5G 1K9 Canada</p>

[Update Account Page](#)

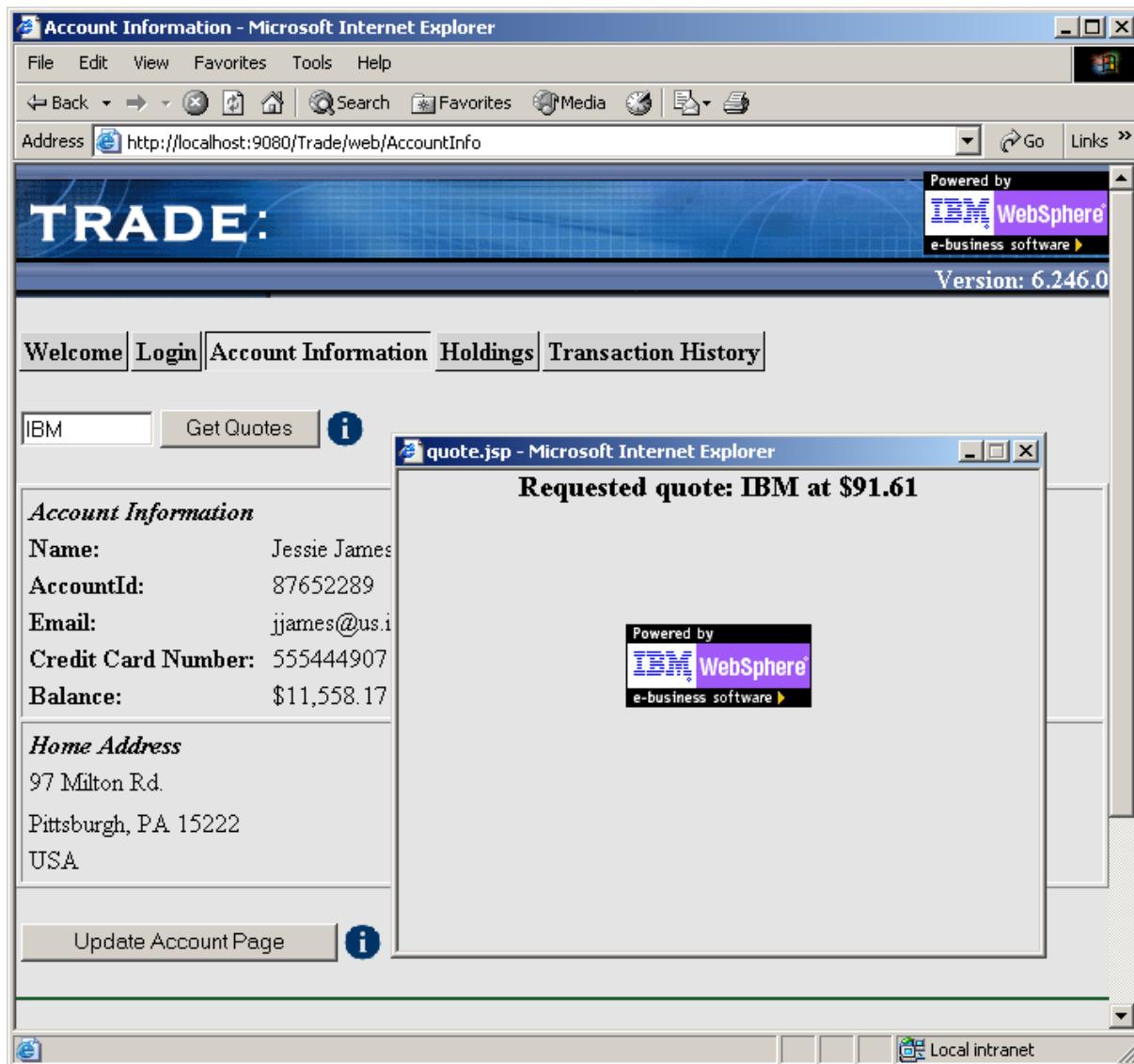
Local Intranet

The information icons appear throughout the application. Selecting an icon shows background information as to how the application performs a particular function. A pop-up window will appear with information for that icon.



- ___ f. Exit the information window.

g. On the Account Information page enter IBM and click Get Quotes.



Close the quote.jsp pop-up window. Feel free to obtain additional quotes.

2. Select the **Holdings** tab. The **Holdings** page shows what stocks are currently in your portfolio. It also displays your balance and allows you to buy and sell stocks.

The screenshot shows a Microsoft Internet Explorer window with the title "Holdings - Microsoft Internet Explorer". The address bar shows the URL <http://localhost:9080/Trade/web/Holdings>. The page itself has a blue header with the word "TRADE" in large letters. Below the header, there is a menu bar with "Welcome", "Login", "Account Information", "Holdings" (which is highlighted), and "Transaction History". A toolbar below the menu includes "Get Quotes" and an information icon. The main content area displays two rows of stock holdings:

Holding	Shares	Latest Price Per Share	Shares to Sell
IBM	3.0	\$92.58	<input type="text"/> SELL
DELL	4.0	\$40.09	<input type="text"/> SELL

Below this, there is a section for "Your current balance is: \$11,558.17" with fields for "Symbol", "Quantity", and a "BUY" button. At the bottom of the page, there is a "Runtime server information" section with tabs for "Cell", "Node", and "Process".

Feel free to buy and sell stock. Notice that as you buy and sell your balance changes accordingly and also your holdings reflect your actions.

3. Experiment with all features of the application to verify that all pages and functions work.
4. You may also logout and register a new user, then log in as the new user and perform the same functions as you did with **client** and verify that all is still well.
5. Close the browser.
6. Logout of the administrative console and close the browser.
7. For extra credits use the DB2 Control Center to verify that the tables in the TRADE database have updated according to your actions.

End of lab

Exercise review and wrap-up

The first part of the exercise looked at installing an application using the WebSphere administrative console. After installation of the TradeApplication, the application was started and tested.

Exercise 6. Problem determination (optional)

What this exercise is about

This lab illustrates how to view and configure log and trace files. In this exercise you use WebSphere tools including Application Server Toolkit (AST) and the Log Analyzer for troubleshooting.

What you should be able to do

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

- Locate and view some log files according to the profile concept of the WebSphere Application Server
- Use the Log Analyzer and AST to view and analyze the activity.log
- Enable tracing and different details of Logging
- Work with Hang Thread Detection (optional)

Introduction

In this lab, you will locate and view log files for the WebSphere Application Server. You will use a text editor to view the log files and use the Log Analyzer and AST tools to view logs and trace. Students also learn how to enable tracing for an application server by using the administrative console.

Required materials

To perform this exercise, you must have a working WebSphere Application Server **server1**, administrative console and a running Default Application and Trade Application installed on **profile1**.

Instructor exercise overview

The first part of this exercise is performed using **server1** on **profile1**. Students change settings and look at log files using the administrative console.

Next in this exercise the students learn how to work with log and trace output using the Log Analyzer first and then the AST. For trace monitoring students use the hitcount application and the Trade Application running **server1** on **profile1**.

The optional part of the lab deals with hung thread detection. Students have to install an application and configure the hung detection policy.

Information about Step 4 using Trace

- Students will enable ConnectionManager for trace
`com.ibm.ejs.j2c.ConnectionManager*=finer`
- three tests with DefaultApplication /hitcount and ivtApp (EJB test)
 1. - first with **Servlet** (HttpSession variable) --> no connection
 2. - second with **/hitcount (using CMP EJB)** --> yes, connections (each request)
 3. - third with **/ivtejb (using Session EJB)** --> no connection

Note: /ivtejb returns a blank screen only!

Information about the optional part (HangThreadLab)

After requesting the address: <http://localhost:9080/HangLab/HangServlet>.

You will not see any output in this browser window. It is just "waiting."

Exercise instructions

Preface

You can use any text editor to view most log files. Unless the instructions tell you differently
- close the editor window after you examine each file.

Step 1 Working with log files of the application server

In WebSphere Application Server V6 the location of the log files is <profile_root>\logs. With the new logging architecture (JSR-047) the configuration of the **log detail level** is separated from JVM and process log settings.

Information: This lab is performed using the **server1** application server from the <profile_root>\profile1 folder. Make sure no other Application servers are **not** running before proceeding.

- ___ 1. Verify that **profile1's server1** is running.
 - ___ a. Navigate to the <profile_root>\profile1\bin folder and execute the **serverStatus server1** command.
 - ___ b. If server1 is not running execute the **startServer server1** command to start the application server.

UNIX: Use a Terminal Window:

- ___ a. Navigate to <profile_root>/profile1/bin.
 - ___ b. At the prompt, type **./serverStatus.sh server1**
 - ___ c. if server is not running **./startServer.sh server1**
- ___ 2. You can change the location, name and other settings of log files using the administrative console.
 - ___ a. Log in to the administrative console:
http://localhost:9060/ibm/console

- ___ b. In the navigation tree, select **Troubleshooting** —> **Logs and Trace** —> **server1**.

The screenshot shows a software interface titled "Logging and Tracing". At the top, there is a blue header bar with the title and a help icon. Below the header, the text "Logging and Tracing > server1" is displayed in a bold, dark blue font. A descriptive text follows: "Select a system log to configure, or specify a log detail level for components and groups of components. Log levels allow you to control which events are processed by Java logging." Underneath this text, there is a section titled "General Properties" with a blue underline. To the left of this section, there is a vertical list of five items, each preceded by a small gray square:

- [Diagnostic Trace](#)
- [JVM Logs](#)
- [Process Logs](#)
- [IBM Service Logs](#)
- [Change Log Detail Levels](#)

Information: You can also reach the configuration area for Logging and Tracing by selecting **Servers** —> **Application Servers** —> **server1**. Select **Logging and Tracing** under the Troubleshooting section.

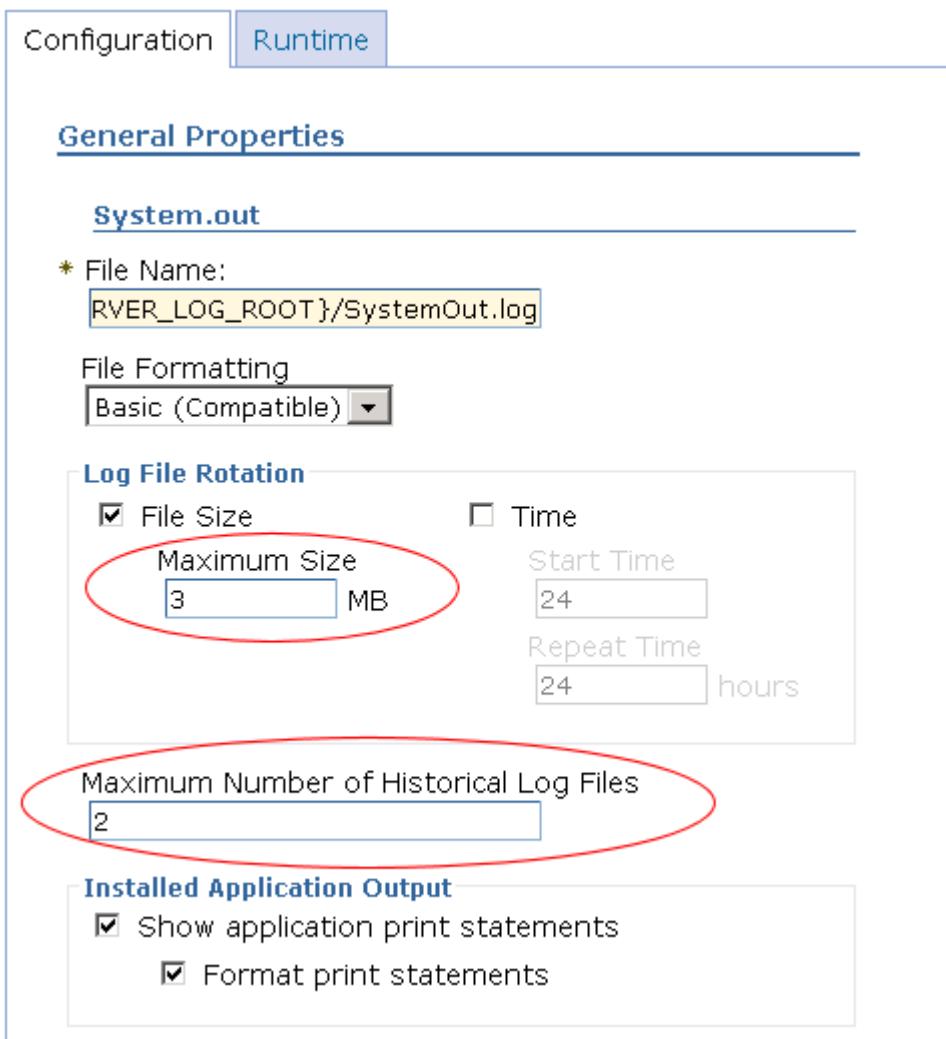
- ___ 3. Change the number of history files and set the maximum size of the log file for "System.out".

- a. Select **JVM Logs**. You can view and modify settings from the Logging and Tracing window for System.out and System.err logs.

The screenshot shows the 'Logging and Tracing' interface with the 'JVM Logs' tab selected. The main content area provides a detailed description of JVM logs, stating they are created by redirecting System.out and System.err streams to independent log files. It mentions that the System.out log monitors the application server's health and the System.err log contains exception stack trace information. Changes on the Configuration panel apply on restart, while Runtime changes apply immediately.

Below the description are two tabs: 'Configuration' (selected) and 'Runtime'. The 'General Properties' section is expanded, showing settings for 'System.out'. Under 'File Name', the value is set to \${SERVER_LOG_ROOT}/Syste... (partially visible). Under 'File Formatting', the dropdown is set to 'Basic (Compatible)'. The 'Log File Rotation' section is expanded, showing 'File Size' checked and 'Time' unchecked. 'Maximum Size' is set to 1 MB, 'Start Time' is 24, and 'Repeat Time' is 24 hours. The 'Maximum Number of Historical Log Files' is set to 1.

- ___ b. Under General Properties for **System.out** set the **Maximum Size** to 3 MB Size and the **Maximum Number of Historical Log Files** to 2.



Click **OK**.

- ___ c. Save the changes to the master configuration.
- ___ 4. View **SystemOut.log** and **SystemErr.log** for server1 using the administrative console.
- ___ a. Select **Troubleshooting** → **Logs and Trace** → **server1** → **JVM Logs** and select the **Runtime** tab. Click **View** to the right of the File Name for SystemOut.

- ___ b. Do not retrieve more than 250 lines in one step. You can modify the number of lines to be retrieved at the top of the Logging and Tracing window.

Logging and Tracing

Logging and Tracing

[Logging and Tracing](#) > [server1](#) > [JVM Logs](#) > [Log file](#)

Display the contents of the Web server log file.

Total: 2464, Filtered total: 250

Retrieve Lines (eg. 250-600)

250

Refresh

- ___ c. Specify the range of lines from 250 to 500 to retrieve and click Refresh.

Logging and Tracing

Logging and Tracing

[Logging and Tracing](#) > [server1](#) > [JVM Logs](#) > [Log file](#)

Display the contents of the Web server log file.

Total: 2464, Filtered total: 200

Retrieve Lines (eg. 250-600)

250-500

Refresh

Information: You can also navigate to the `<profile_root>\profile1\logs\server1` folder to view the logs.

Step 2 Using the Log Analyzer

The Log Analyzer tool is used to view the IBM service log, a binary file also known as the activity log file. This file contains audit, warning, and fatal messages pertaining to the WebSphere environment. Output for this log file cannot be viewed using a simple text editor.

- ___ 1. View information in the IBM Service Log.
- ___ a. Select **Troubleshooting** → **Logs and Trace** → **server1** → **IBM Service Logs**.

[Logging and Tracing](#) > [server1](#) > IBM Service Logs

Configure the IBM service log, also known as the activity log. The IBM service log contains both the WebSphere Application Server messages that are written to the System.out stream and some special messages that contain extended service information that can be important when analyzing problems. There is one service log for all WebSphere Application Server 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.

Configuration

General Properties

- Enable service log
- * File Name:
\${LOG_ROOT}/activity.log
- * Maximum File Size
2

Message Filtering
Log all messages ▾

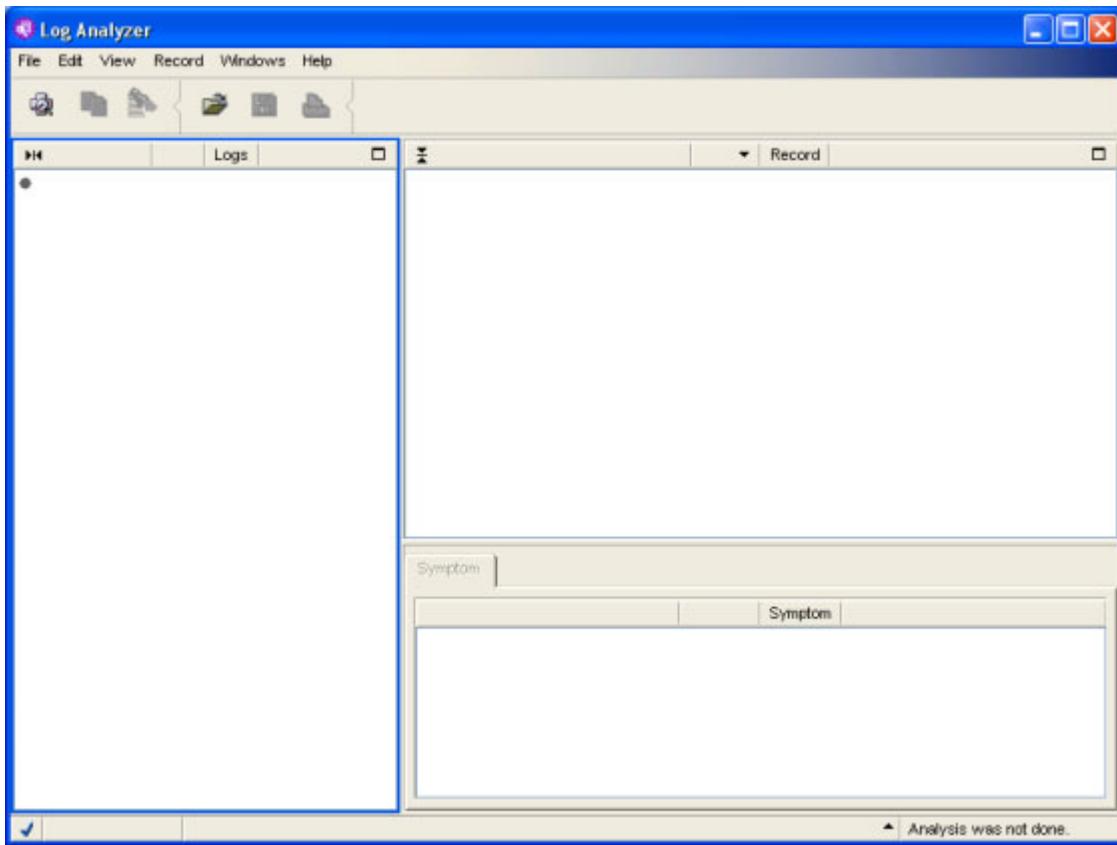
- Enable Correlation ID

___ b. Can the IBM Service log be disabled.? _____

Information: You find the answer in the picture above. For \${LOG_ROOT} you can look to **Environment** —> **WebSphere Variables**.

___ 2. Start the Log Analyzer tool.

- ___ a. In a command prompt navigate to <profile_root>\profile1\bin and enter the **waslogbr** command. The Log Analyzer tool launches and looks like the graphic below.



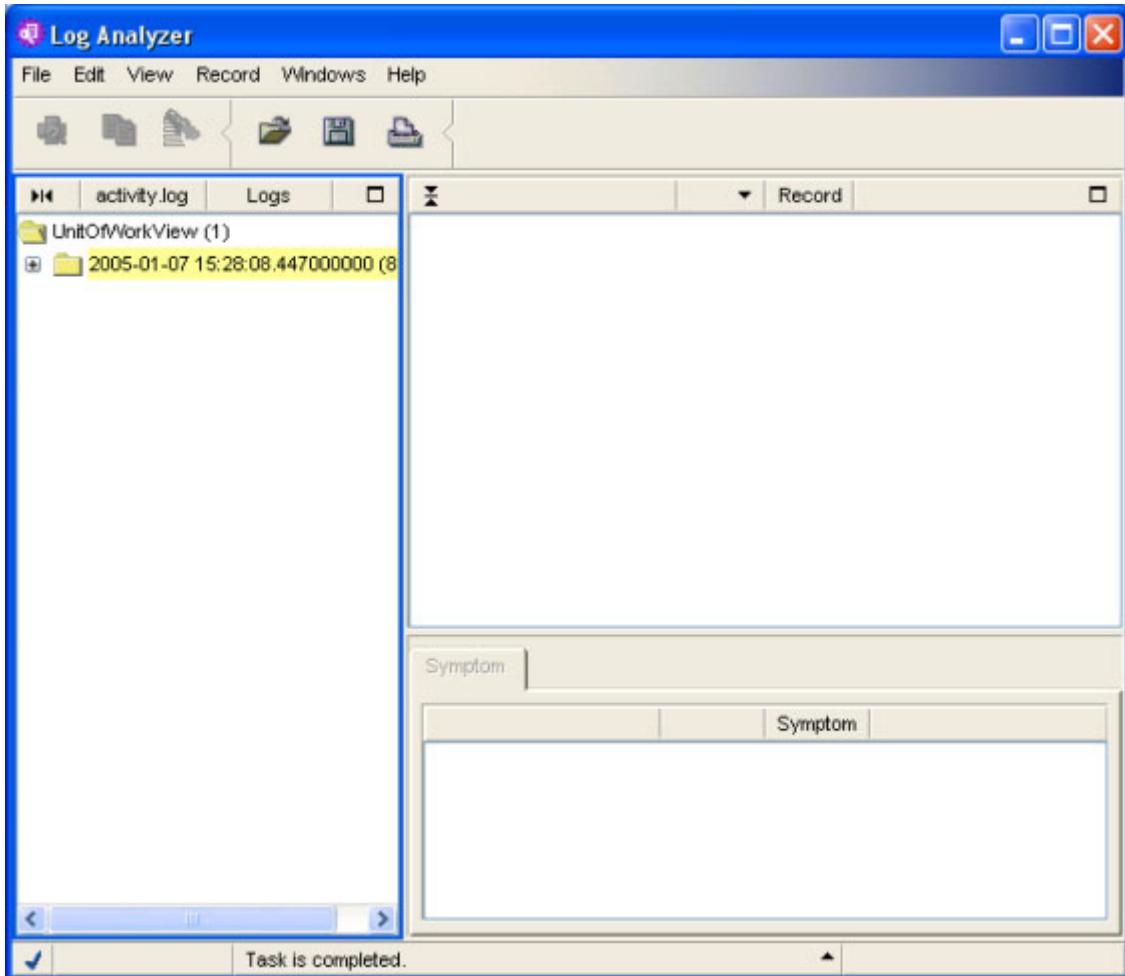
UNIX: <profile_root>/profile1/bin/
enter the command **./waslogbr**

waslogbr uses the Korn shell, RedHat Fedora standard installation does not include ksh.
The lab image has the ksh package installed.

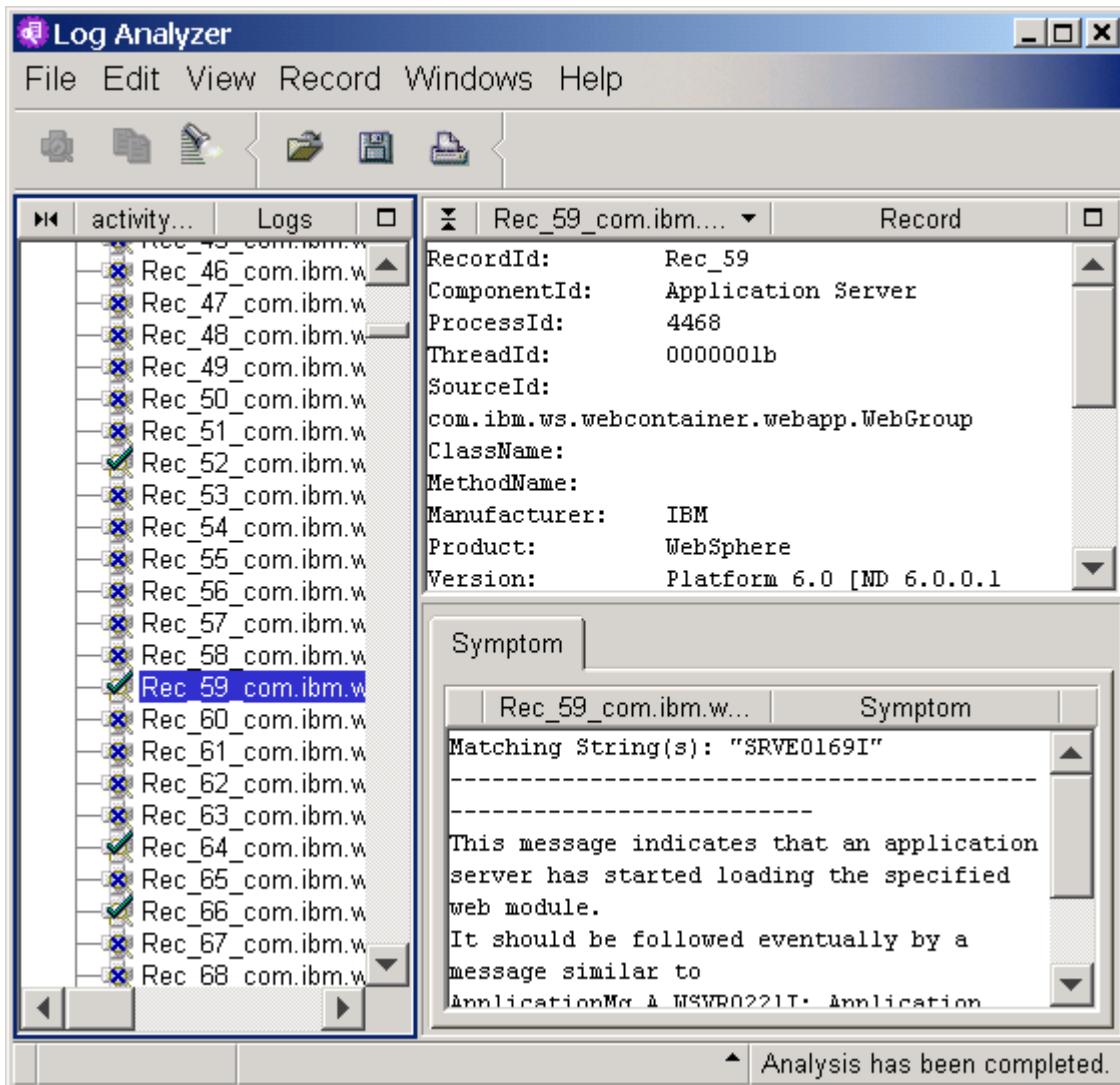
- ___ 3. Open and view the **activity.log** in **Log Analyzer**
- ___ a. From the file menu, select **File** —> **Open**.
- ___ b. Navigate to the <profile_root>\profile1\logs directory. Select the **activity.log** file and click **Open**.

Information: It takes a few moments for the tool to interpret the activity log file and display its output.

- ___ c. The contents of the log file is listed in the left-hand pane of the Log Analyzer. By default, entries are sorted by unit of work.



- ___ d. Expand the list down to the record level and select a record. Details for that record are displayed in the upper pane on the right. If there is any diagnostic information available (from a symptom database) for that record, it displays on the lower right-hand pane (this information shows after analysis has been performed, see step 4 below).

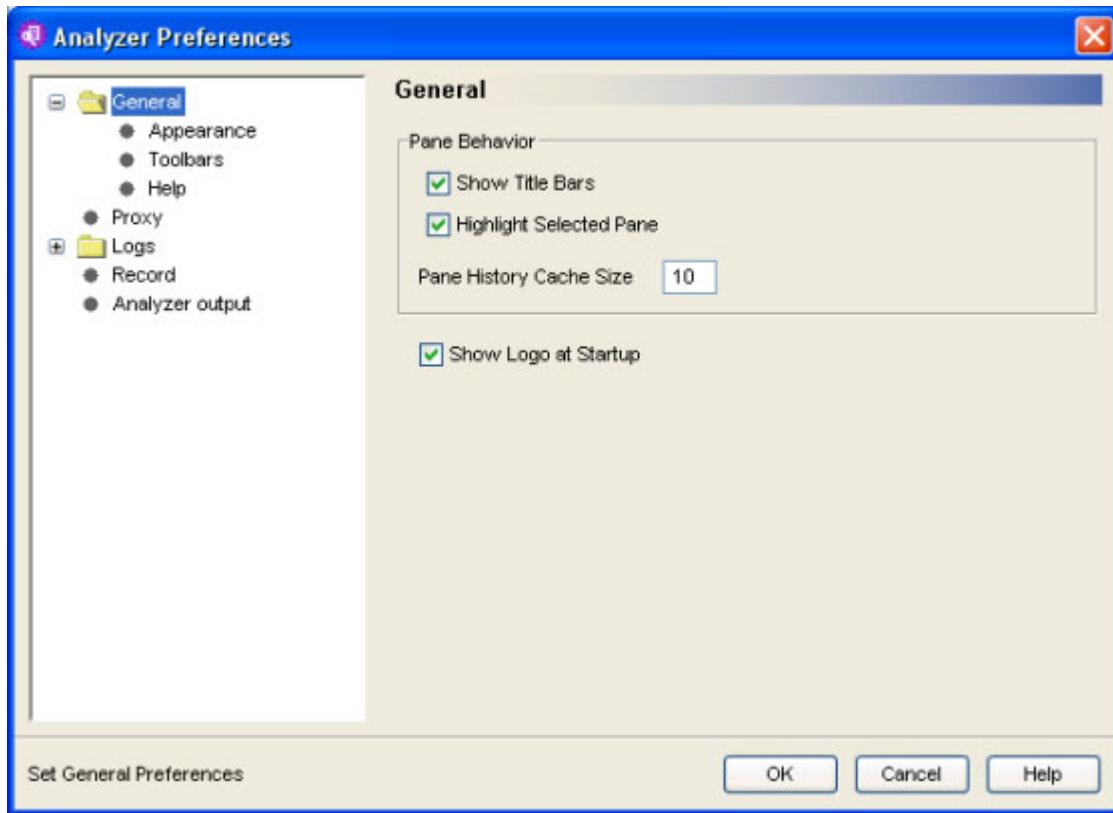


Records are color coded to indicate the level of severity. Records not colored are of information nature.

- 4. Log Analyzer has a feature to download the latest diagnostic information into the symptom database. If there is an active Internet connection perform this step. Select **File** —> **Update database** —> **WebSphere Application Server Network Deployment Symptom Database**.
 - a. After a few moments the latest symptom database is loaded and the tools is now in a position to analyze records on the log. Find a Warning or Severe record and from the menu select **Record** —>**Analyze**. The detailed symptom information is now displayed on the Symptom pane.

Information: It is also possible to start an analysis by right-clicking on an item and choosing **Analyze**.

- ___ 5. To view the Log Analyzer preferences select **File —> Preferences**. Look through the preferences.



- ___ a. Exit from the Analyzer Preferences window when done viewing the preferences.
 ___ 6. Close the Log Analyzer tool.

Step 3 Using AST for viewing service and JVM logs

The Application Server Toolkit (AST) is built using Eclipse technology with features for team programming, debugging, J2EE application deployment and more. The Log and Trace Analyzer is a set of tools used to collect standardized data, it provides a number of views for log file analysis. It also has the ability to combine several log files into a single unit for analysis.

- ___ 1. Start the Application Server Toolkit and change the perspective.
 ___ a. Start AST under Windows: **Start —> Programs —> IBM WebSphere —> IBM WebSphere Application Server Toolkit, V6 —> Application Server Toolkit**. Accept the default Workspace.

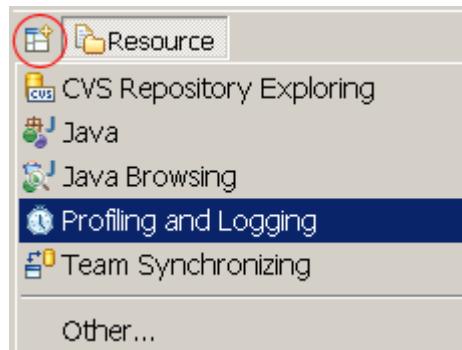
UNIX: Use a Terminal Window: Navigate to **opt/IBM/WebSphere/AST**.

At the prompt, type **./ast**

LINUX: If there is an icon on LINUX desktop: double-click the AST icon.

- ___ b. Switch to the **Profiling and Logging** perspective using the Perspective switch toolbar or menu, which is located in the top-left below the main toolbar. You may

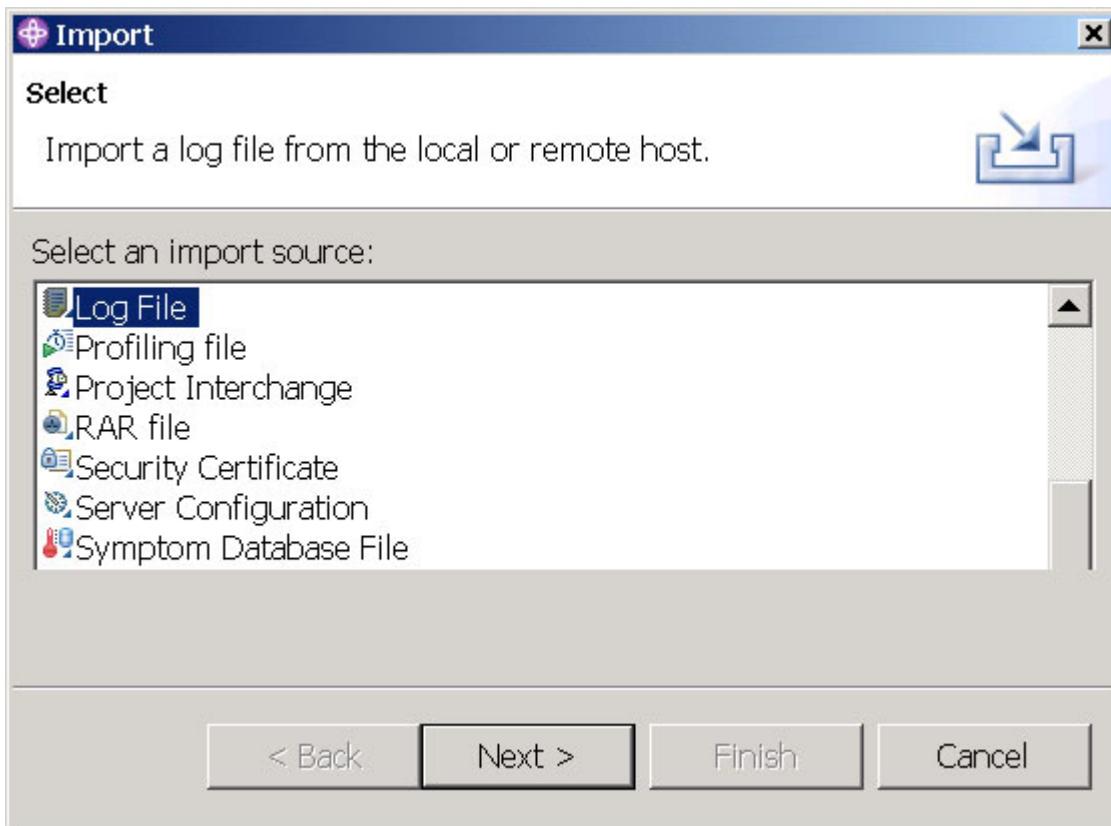
have to click the top-left “Open a Perspective” smart icon.



- ___ c. Another option if the Switch Toolbar is not available, click menu **Window** —> **Open Perspective** —> **Other...** and click **Profiling and Logging**.

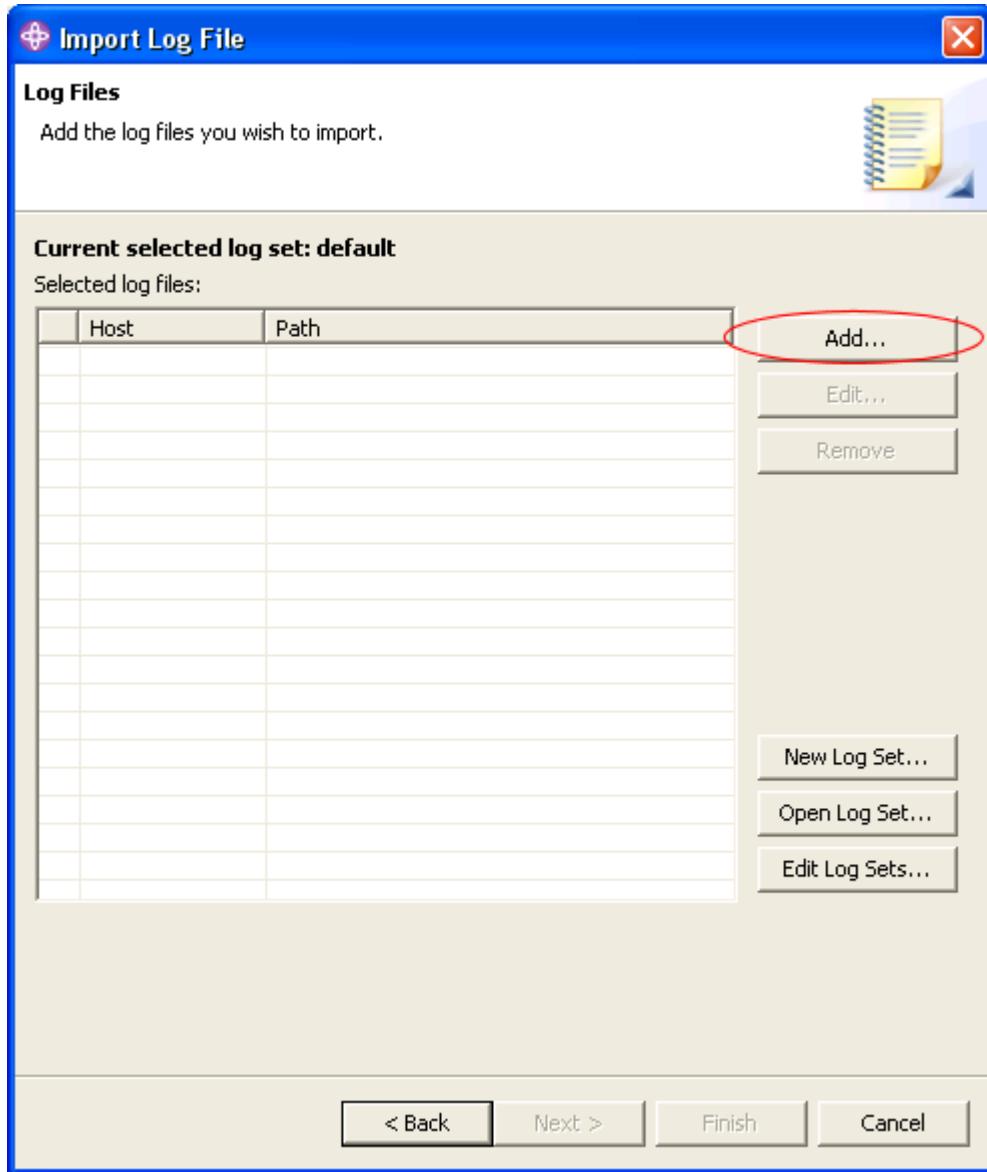
Information: Anytime you have lost some windows or views inside the perspective or want to get the default arrangement of the perspective back, click menu **Window** —> **Reset Perspective**.

- ___ 2. Import the **activity.log** into AST using the import wizard.
- ___ a. In AST on the left-side activate the **Log Navigator**. Right-click with mouse on the empty space inside this view window, select **Import** —> **Log File**.

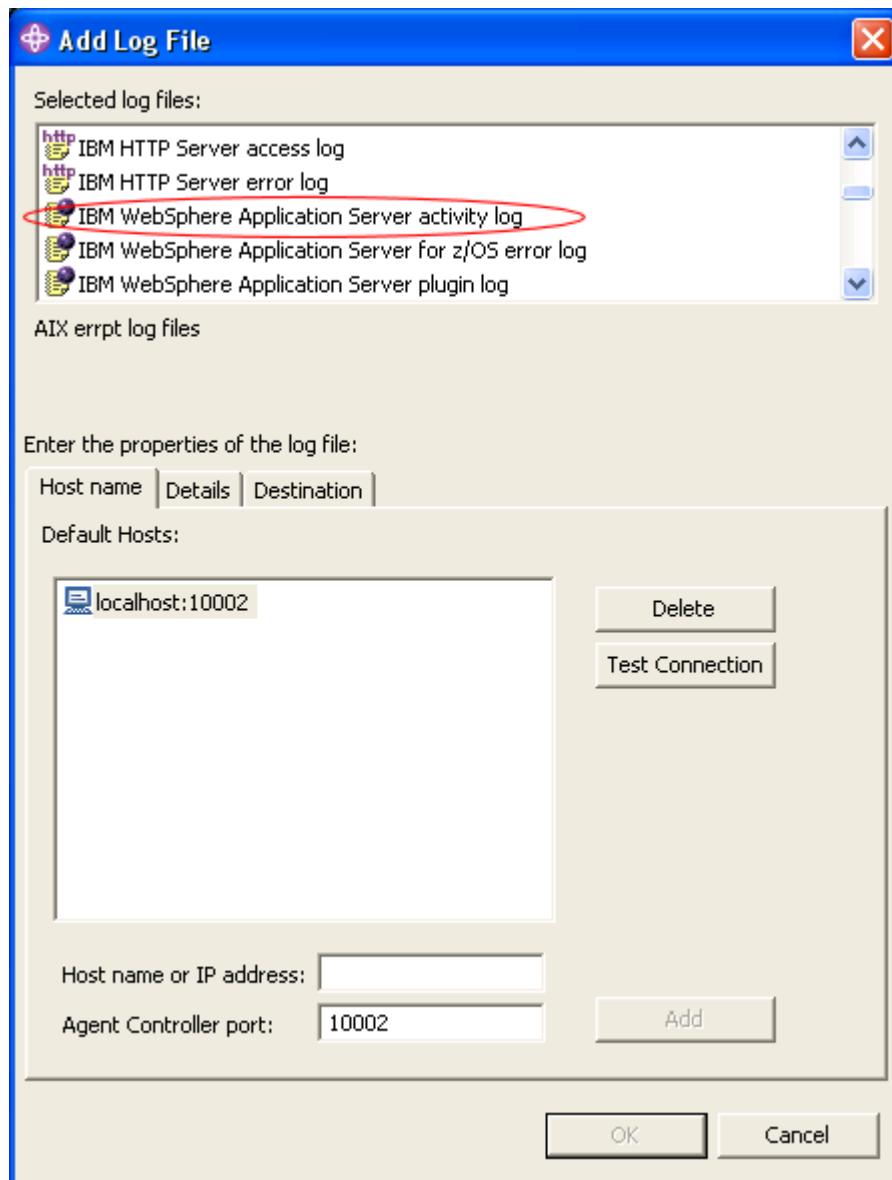


- ___ b. Click **Next**.

- ___ c. On the Import Log File window click **Add...**.

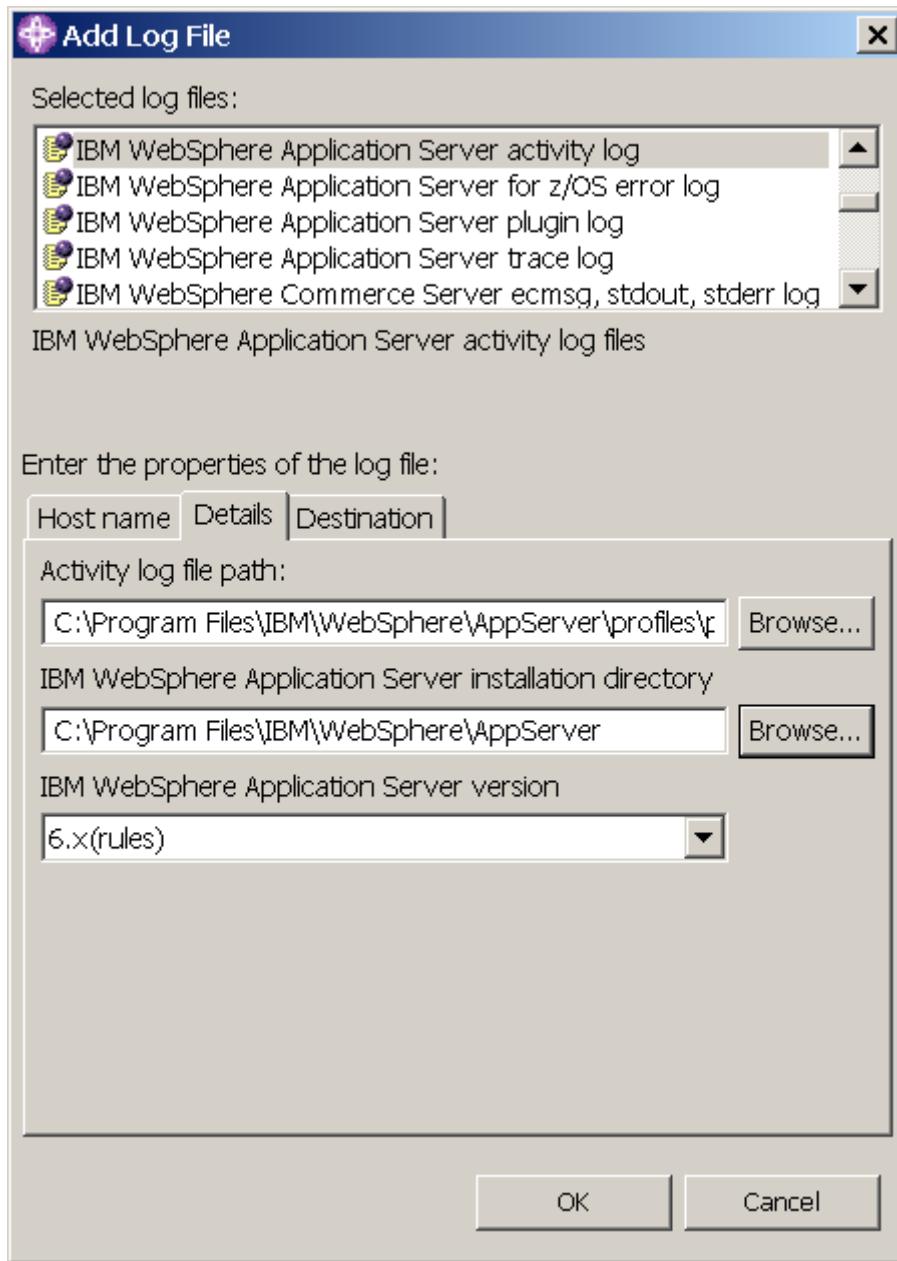


- ___ d. Select **IBM WebSphere Application Server activity log**. Keep the remaining defaults.



- ___ e. Click the **Details** tab under “Enter the properties of the log file.” For the **Activity log file path** navigate to <profile_root>\profile1\logs and select the **activity.log** file. Click **Open**. For the **IBM WebSphere Application Server**

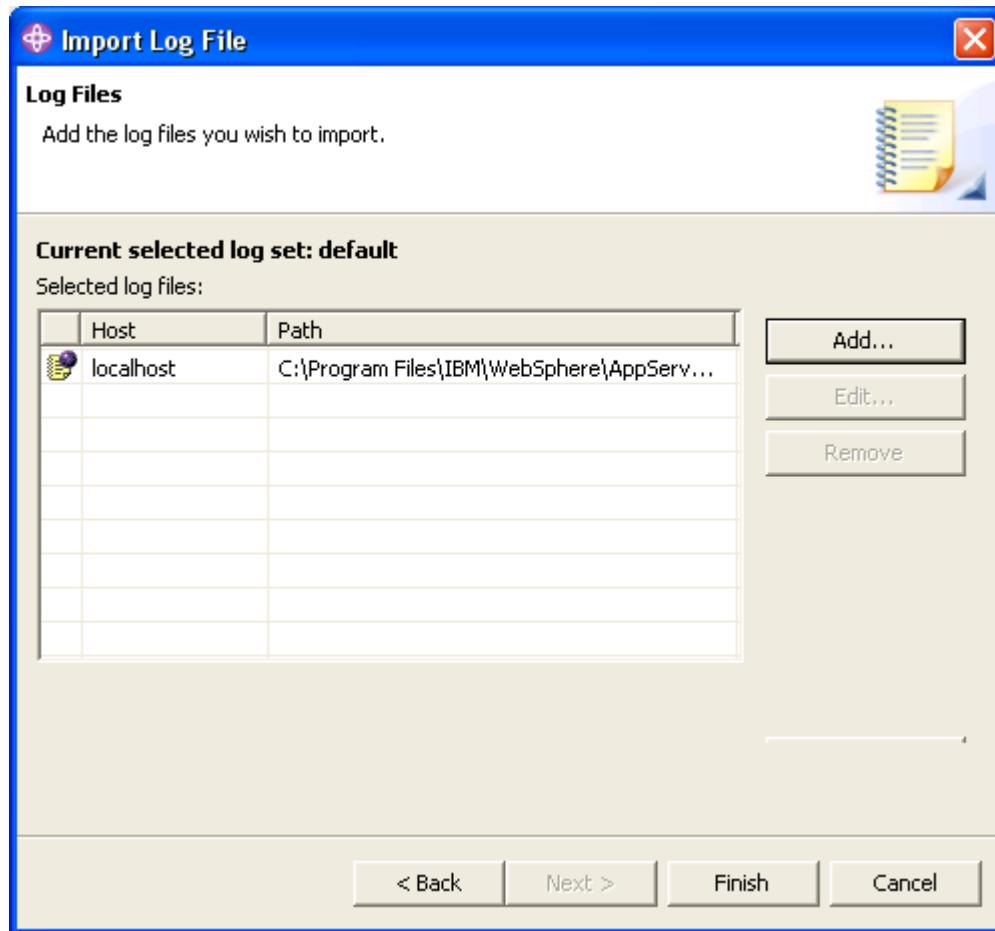
Installation Directory navigate to <was_root> and click **OK**.
Leave the IBM WebSphere Application Server Version as **6.x(rules)**.



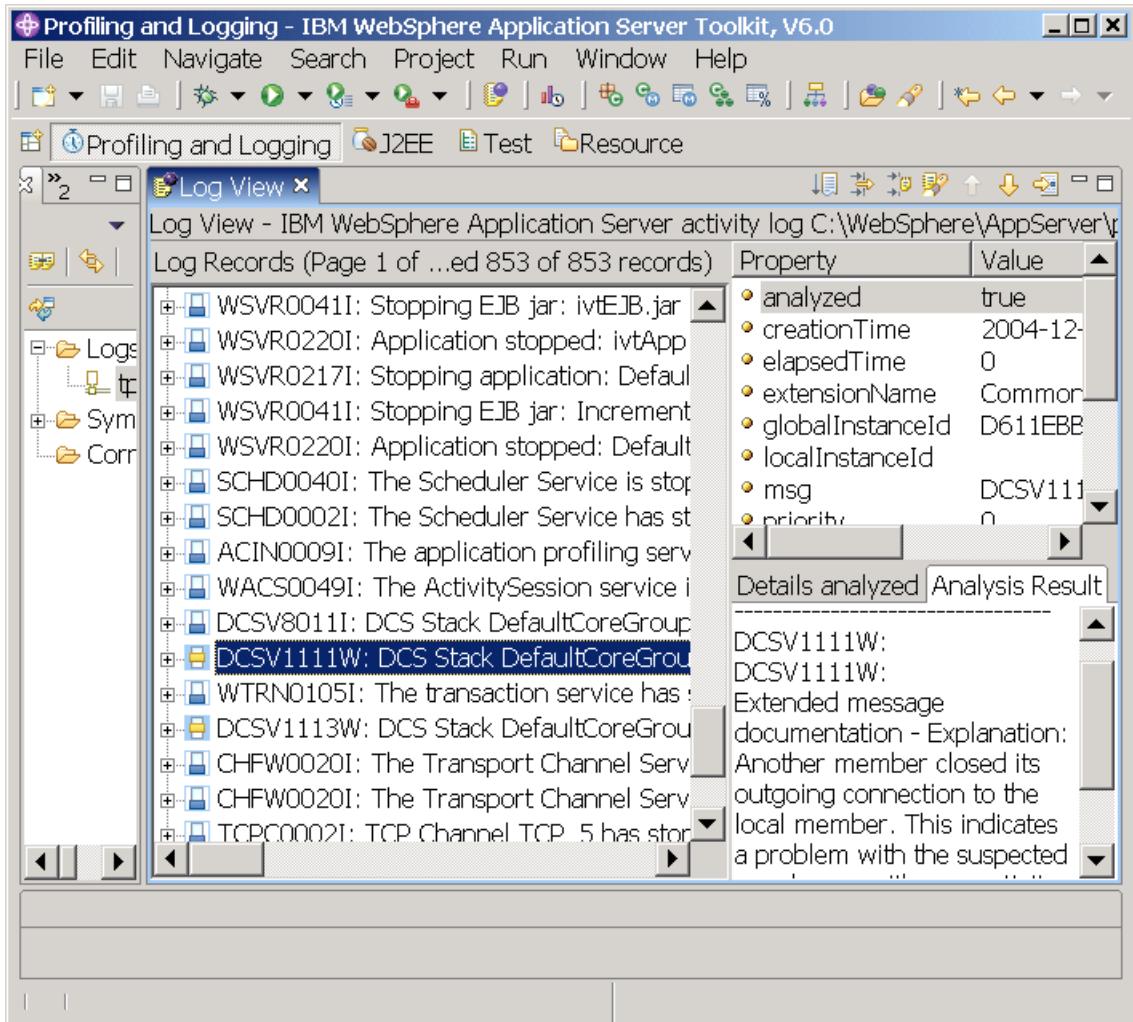
Information: Notice that there are a lot of different types of log files that are supported by the tool, more than WebSphere Application Server, for example, Portal Server, AIX, DB2, z/OS, HTTP Server, WebSphere MQ, and so forth.

UNIX: Use the appropriate directories according to your environment.

- ___ f. Click **OK**. You should have one entry for localhost as Selected log files.

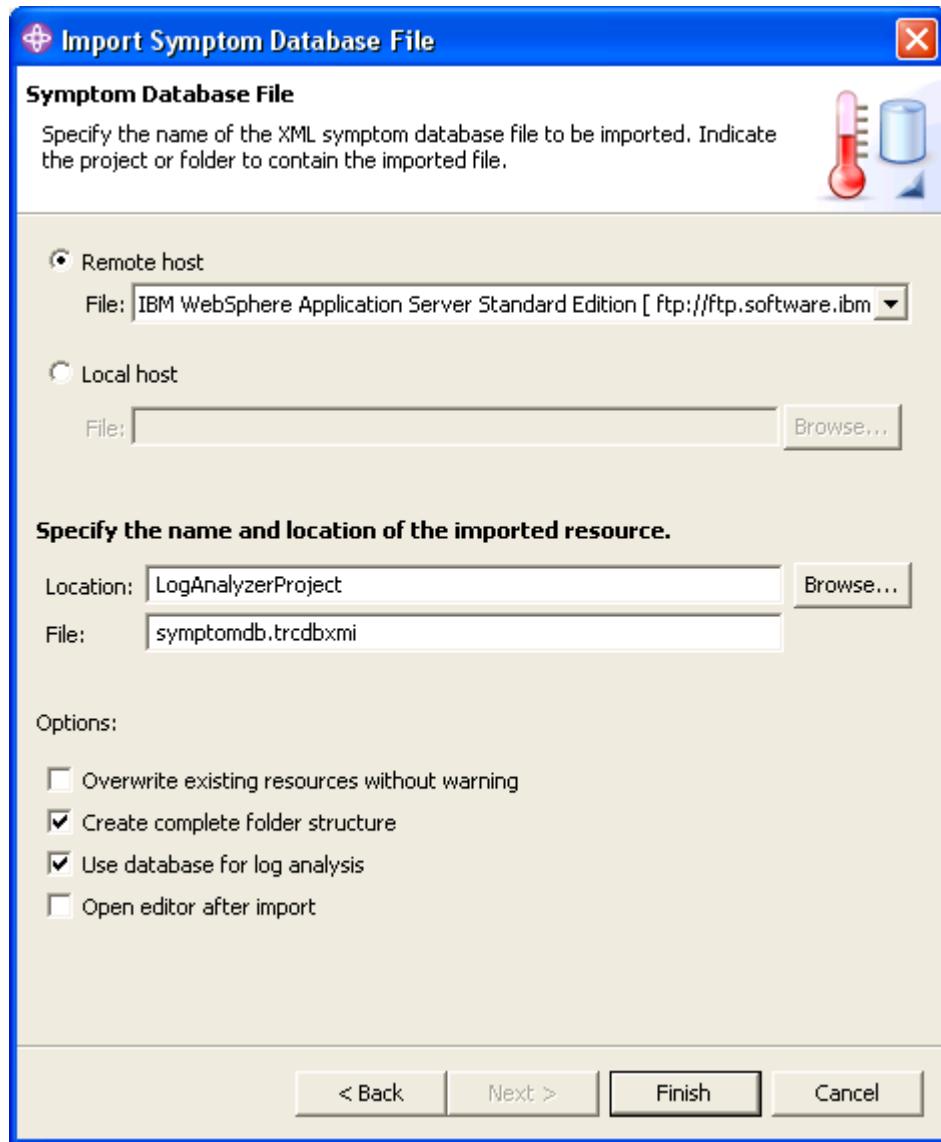


- ___ g. Click in **Finish** to end the import wizard.
___ h. The **Log View** window automatically opens. If not, click AST menu **Window** —> **Show View** —> **Other**. Expand **Profiling and Logging** and click **Log View**.



- ___ 3. Import the Symptom Database into the AST workspace.
 - ___ a. Right-click Symptom Databases in the Log Navigator window. In the menu that appears select **Import —> Symptom Database File**. Click **Next**.

- ___ b. On the Import Symptom Database File window **deselect** the **Open editor after import** option. Keep all remaining defaults. Click **Finish**.

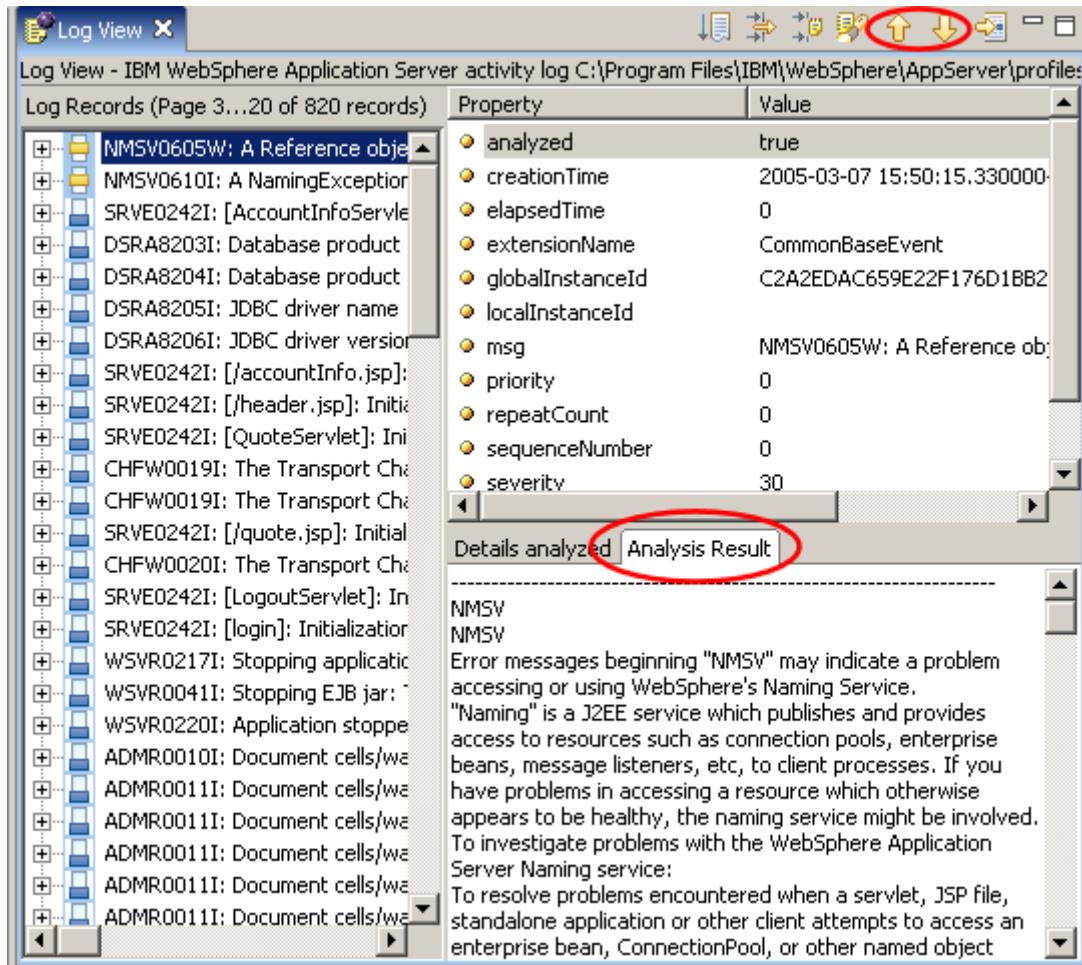


Information: If Internet access is not available during the lab, use a local version of the symptom database contained in the **symptomdb.zip** file located in the **<software_dir>\Troubleshooting** folder. Unzip this file to same folder and follow the description given above to import. This time select **localhost** and browse to the folder with the unzipped **symptomdb.xml**.

- ___ 4. View and analyze the IBM Service Log.
- ___ a. In the Log Record area right-click the first record, select **Analyze All —> Default Log Analyzer**. This takes a minute. After Analyzing you can examine the records.

To view the next page with records use the **Navigate** menu or the smart icons on the top-right on the Log View.

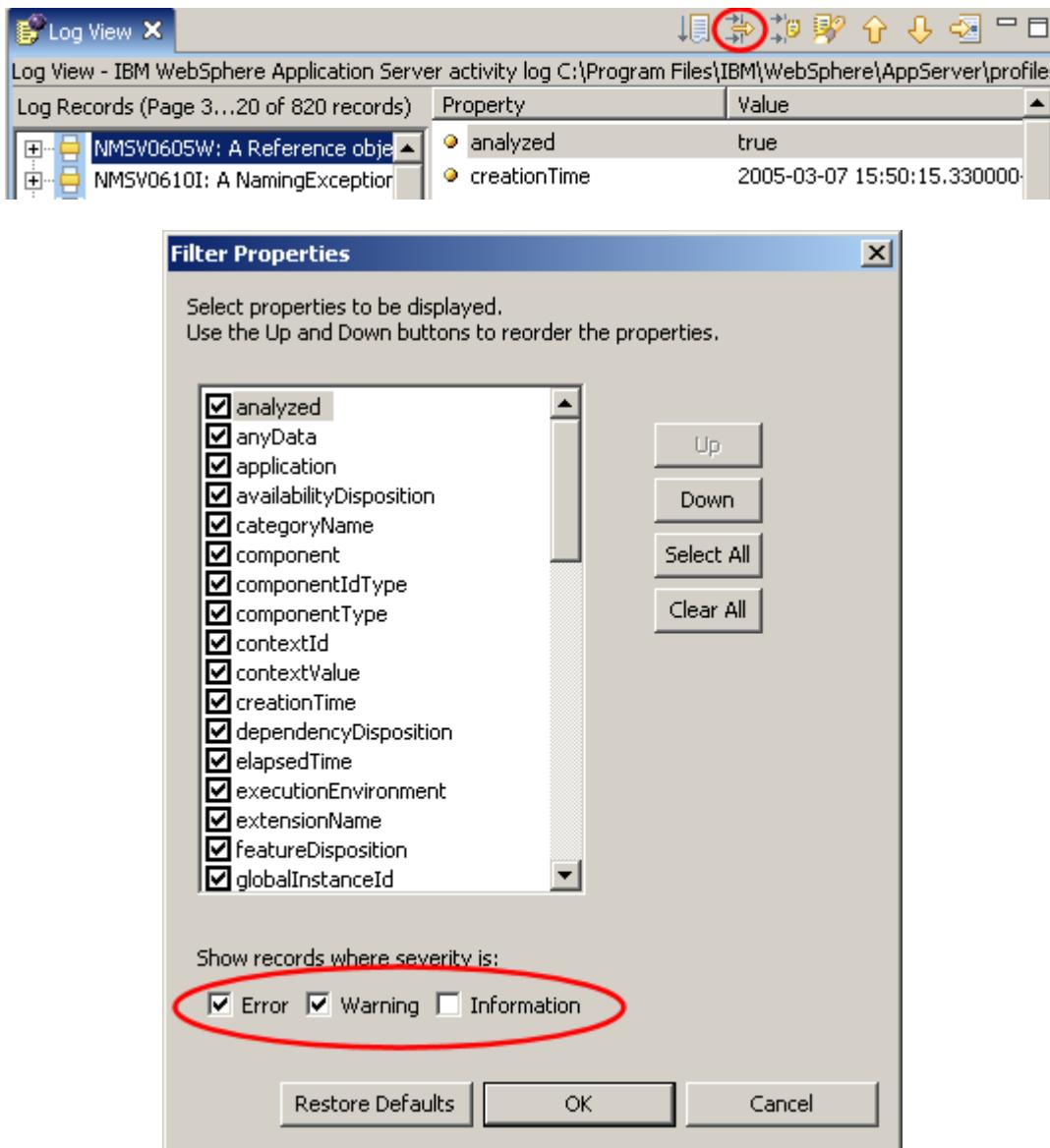
To read the Analysis Result you need to select the **Analysis Result** tab on the right side. Yellow color in the Log Records area indicates an existing Analysis entry.



Information: Filtering in the Log and Trace Analyzer is based on the Common Base Event properties. You can create filters to examine specific data from the application log files. Correlation of one or more log files is also supported. For more information use the eclipse help system (menu Help → Help Contents, click Log and Trace Analyzer).

With AST you can also visit and analyze **JVM Logs**, **Trace** data, which can be located on remote server using IBM Agent Controller. At the end of this lab there is an optional part to use AST for JVM Log and Trace.

- ___ 5. Use the **Filter Properties** icon to view only **Error** and **Warning** records.



- ___ 6. Close the Application Server Toolkit.

Step 4 Enabling tracing on an application server

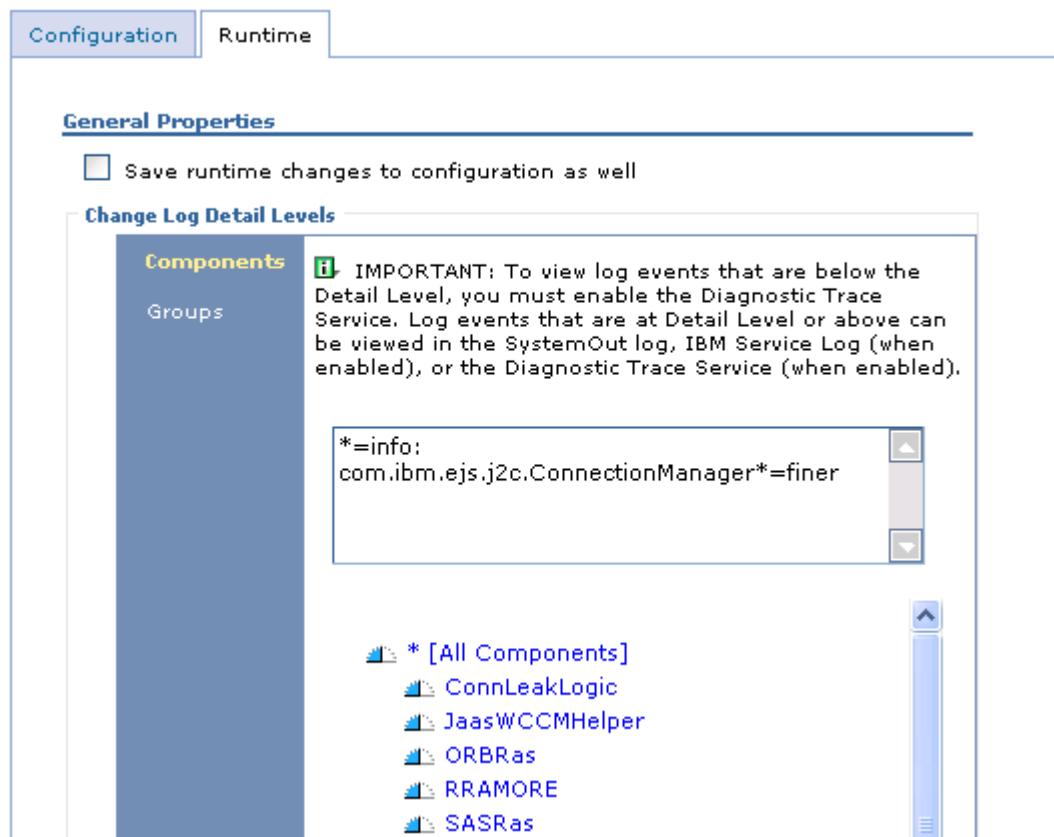
Additional logging can be enabled for events in the WebSphere Application Server by using the tracing features of WebSphere.

Information: Tracing impacts performance, tracing should only be enabled on specific components, if possible, and then disabled when it is no longer needed.

- ___ 1. Verify that server1 in profile1 is running.
 ___ 2. Using the administrative console for profile1. Select **Troubleshooting** → **Logs and Trace** → **server1** → **Diagnostic Trace**.

- ___ a. On this page the **Enable Log** check box is selected by default, so tracing can be started without restarting the server.
 - ___ b. Select trace output type as file.
 - ___ c. Set the size of the file to **25 MB** and the number of historical files to **2**.
 - ___ d. Remember the name and location of the trace **trace.log** file. Click **OK**.
 - ___ e. **Save** the changes to the master configuration.
- ___ 3. Next, change the Trace level. Because of changes to the Logging infrastructure in WebSphere Application Server V6 (Java Logging API) the Trace level is configured on a separate panel.
- ___ a. Select **Troubleshooting** —> **Logs and Trace** —> **server1** —> **Change Log Detail Level** and wait until all components are shown in the browser window.
 - ___ b. Select the **Runtime** tab and wait until all components are visible.
-
- Information:** Changes made on the **Runtime** tab work on actively loaded modules. Changes made on the **Configuration** tab will not take place until the application server is restarted.
- ___ 4. Trace the Java 2 Connectivity with a level of **finer**. To change the Log Detail Levels you have **two possibilities**:

- __ a. Type in the multiple line entry field the following:
***=info: com.ibm.ejs.j2c.ConnectionManager=finer** and click OK.



- __ b. Or expand **com.ibm.ejs.***, **com.ibm.ejs.j2c.***. Click **com.ibm.ejs.j2c.ConnectionManager** and select **finer**. Click OK.

Information: The option of dynamically expanding the components will only display components that have been loaded by the application server. If the application server has not accessed the database since its last restart, the **ConnectionManager** component will **not** be listed under the **com.ibm.ejs.j2c** package.

- __ 5. Verify the trace output using the /hitcount servlet in the DefaultApplication.
- __ a. In a command prompt window navigate to
<profile_root>\profile1\logs\server1.
- Information:** Use the tail utility to monitor the trace output. If there is no tail available you can use Notepad or ASCII editor as tool for to open the trace.log file in this directory.
UNIX: <profile_root>/profile1/logs/server1
- __ b. If tail is available start it: **tail -f trace.log**. In the trace output find the **TRAS0018I** information, that the trace state has changed:
***info:com.ibm.ejs.j2c.ConnectionManager=finer**. Leave the tail window open.

- ___ c. In a Web browser specify the address:
http://localhost:9080/hitcount?selection=SS2
 Use the Reload button in the browser a few times. You should only see information about the servlet initialization (only if the servlet was not already running), but should not see any information about J2C connectivity.
- ___ d. In the same browser of the **Hit Count Demonstration** click **Enterprise JavaBean (CMP) → Global Namespace → Commit → Increment**. The address in the browser should be the following:
http://localhost/hitcount?selection=EJB&lookup=GBL&trans=CMT
- ___ e. There should be a lot of information in the **trace.log** file. Look for **allocateConnection** and **XXXXXXXXX Entry**. If this was the first request of hitcount using CMP EJB you will find the JDBC driver name getting the connection entry in the trace.
- ___ f. Reload the request in the browser. Look again for **allocateConnection** and **XXXXXXXXX Entry**.
 Does the connection use the same entry? _____
 Compare the timestamp and duration of both connections!
- ___ g. In a Web browser specify the address: **http://localhost:9080/ivt/ivtejb** (Only a blank screen appears as response in the browser.)
 Are there any connections? Why not? _____
 Hint: What kind of EJB is used? (Hint: Not Entity EJBs) _____
- ___ 6. Test the connection trace using the TradeApplication.
- ___ a. In a browser window enter **http://localhost:9080/Trade/web** and log in.
 Does the connection use the same entry as for hitcount? _____
 Click TransactionHistory in the browser running the Trade Application.
 Compare the entry with the one used for Login.
- ___ b. If you have Internet access try to get a Quote using the TradeApplication. If you have Internet quotes enabled, you find no connections.
- Information:** Use Trace to obtain detailed information about the execution of WebSphere Application Server components, including application servers, clients, and other processes in the environment. Trace files show the time and sequence of methods called by WebSphere Application Server base classes, and you can use these files to pinpoint the failure.

Trace output is generated as plain text in either basic, advanced or log analyzer format as specified by the user. The basic and advanced formats for trace output are similar to the basic and advanced formats that are available for the JVM message logs. You can also use AST to view and analyze trace output.
- ___ 7. Disable Tracing using the administrative console.
- ___ a. Select **Troubleshooting → Logs and Trace → server1 → Diagnostic Trace**. In the Configuration tab deselect **Enable Log**.

- ___ b. Click **OK**. Save the changes.
- ___ 8. (Optional) Import the **trace.log** into AST. Follow the steps of the previous part of the lab. You can also import **JVM log** files into AST using the Trace log filter of WebSphere Application Server V6.

End of lab

Information: For the optional part of the exercise go to the next page!

Optional exercises: hung thread detection

Step 1 Installing the test application

A common error in J2EE applications is a hung thread. A hung thread can result from a simple software defect (such as an infinite loop) or a more complex cause (for example, a resource deadlock). System resources, such as CPU time, might be consumed by this hung transaction when threads run unbounded code paths, such as when the code is running in an infinite loop. Alternately, a system can become unresponsive even though all resources are idle, as in a deadlock scenario.

Using the WebSphere hang detection policy, you can specify a time that is too long for a unit of work to complete. The thread monitor checks all managed threads in the system (for example, Web container threads and object request broker (ORB) threads).

- ___ 1. Ensure that **server1** within **profile1** is started. Log out from the administrative console.
- ___ 2. Install the HangLabApp enterprise application by issuing a Jython command. The EAR file is located in the **<software_dir>\Troubleshooting\HangThreadLab** folder.
- ___ 3. In a command prompt navigate to the **<profile_root>\profile1\bin** folder and enter the following command (all on one line):

```
wsadmin -lang jython -c
"AdminApp.install(
'c:\software\Troubleshooting\HangThreadLab\hangLabApp.ear' )"
```

UNIX: [all on one line] ./wsadmin.sh -lang jython -c "AdminApp.install(<software_directory>/Troubleshooting/HangThreadLab/hangLabApp.ear)"

- ___ 4. Log in to the administrative console.
- ___ 5. Navigate to **Applications —> Enterprise Applications**. Select the **HangLabApp** using the check box on the left and **Start** the application.

Step 2 Configuring the hang detection policy

Information: Hung thread detection is enabled by default in WebSphere Application Server. In the default configuration, a thread must be active for at least ten minutes before it is marked as potentially hung. In the interest of time, lower that threshold to two minutes by setting a custom property. Set another property to change the interval at which the Thread Monitor polls for hung threads from three minutes to twenty seconds.

- ___ 1. In the administrative console open the Custom Properties panel for your application server.
- ___ a. Select **Servers —> Application servers —> server1**.

- ___ b. Under the Server Infrastructure area expand **Administration** and select **Custom Properties**. You may have to scroll down to see this.
- ___ c. Under Administration, click **Custom Properties**.
- ___ 2. Create a custom property to mark inactive threads as hung threads.
 - ___ a. In the Custom Properties panel, click **New**.
 - ___ b. Under Name enter **com.ibm.websphere.threadmonitor.threshold**.
 - ___ c. Under Value, enter **120**.
 - ___ d. Click **OK**.

Information: This tells the Thread Monitor to mark any thread active for longer than two minutes as hung.

- ___ 3. Create a second custom property to look for hung threads.
 - ___ a. In the Custom Properties panel, click **New**.
 - ___ b. Under Name, enter **com.ibm.websphere.threadmonitor.interval**.
 - ___ c. Under Value, enter **20**.
 - ___ d. Click **OK**. This tells the Thread Monitor to look for hung threads every twenty seconds.
- ___ 4. After clicking **OK**, your Custom Properties list should look like the graphic below.

Select	Name	Value	Description
<input type="checkbox"/>	com.ibm.websphere.threadmonitor.interval	20	
<input type="checkbox"/>	com.ibm.websphere.threadmonitor.threshold	120	
Total 2			

- ___ 5. Save your changes and restart the application server.

Step 3 Detecting hung threads

Now that you have customized the hung detection policy, call a servlet that is intended to hang, so you can see the Thread Monitor in action.

- ___ 1. Make a call to the poorly coded servlet, HangServlet.
 - ___ a. In a Web browser window specify the address:
http://localhost:9080/HangLab/HangServlet.
 - ___ b. You should not see any output in this browser window. It is just "waiting".
- ___ 2. Call HangServlet a second time.
 - ___ a. Open a second Web browser window and specify:
http://localhost:9080/HangLab/HangServlet.
 - ___ b. Like the first window, no output is produced in this window.
- ___ 3. Calling HangServlet twice causes a hang. Since the Thread Monitor threshold was set to 120 seconds, wait two minutes, and then look for the Thread Monitor's output at the end of the **SystemOut.log** file.
 - ___ a. Wait at least two minutes.
 - ___ b. Navigate to <profile_root>\profile1\logs\server1. Open **SystemOut.log** with a text editor.
 - ___ c. You should see a line similar to the following near the end of the file. Use Notepad's Find function (ctrl+F) to search for **ThreadMonitor** if you are having trouble finding it.

```
[8/25/04 11:56:48:234 EDT] 0000002e ThreadMonitor W WSVR0605W:
Thread "Web Container : 0" (610103da) has been active for 260625
milliseconds and may be hung. There are 1 threads in total in the
server that may be hung.
```
- ___ 4. There are some important things to note about this message.
 - ___ a. All log messages generated by the Thread Monitor are printed with the component name "ThreadMonitor", so they are easy to locate.
 - ___ b. The thread name (Web Container: 0) and thread ID (610103da) are printed in the message. These help you locate the hung thread in a Java dump, which is usually the next step in debugging, after you have discovered that there are hanging threads.

Information: The thread name for Web Container threads is different from previous releases. Threads that used to be named `Servlet.Engine.Transports: x` and now named `Web Container: x`.
- ___ 5. Close the browser windows that you opened earlier.

Information: You can configure a hang detection policy to accommodate your applications and environment so that potential hangs can be reported, providing earlier detection of failing servers. When a hung thread is detected, WebSphere Application Server notifies you so that you can troubleshoot the problem.

You can adjust the thread monitor settings by using the wsadmin scripting interface. These changes take effect immediately, but do not persist to the server configuration, and are lost when the server is restarted.

If the thread monitor determines that too many false alarms are issued (determined by the number of pairs of hang and clear messages), it can automatically adjust the threshold. For more information about scripting and false alarms setting visit the Information Center of WebSphere Application Server V6.

End of lab

Exercise review and wrap-up

This exercise examined many of the features of logging and tracing using the administrative console. You looked at the properties and setting of server and tested some of the possibilities with running applications.

Application Server Toolkit (AST) contains a set of tools and wizards to import and analyze log and trace files.

The last (optional) part of the exercise looked at the new Hung Thread Detection feature with a small test application.

Exercise 7. Experimenting with wsadmin

What this exercise is about

This lab covers the use of wsadmin. In this exercise, you will learn to manipulate wsadmin objects from the command line and from scripts. This lab will use both the Jacl and Jython scripting languages.

What you should be able to do

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

- Use wsadmin interactively
- Start wsadmin using Jacl and Jython
- Use wsadmin with a single command or script
- View wsadmin's default scripting language

Introduction

The WebSphere Application Server wsadmin tool provides the ability to execute scripts. You can use the wsadmin tool to manage a WebSphere Application Server V6 installation. This tool uses the Bean Scripting Framework (BSF), which supports a variety of scripting languages to configure and control your WebSphere Application Server installation. The wsadmin tool only supports the Jacl and Jython scripting languages.

The wsadmin launcher makes Java objects available through language specific interfaces. Scripts use these objects for application management, configuration, operational control, and for communication with MBeans running in the WebSphere server processes.

Scripting is a non-graphical alternative that you can use to configure and manage the WebSphere Application Server.

Required materials

To perform this exercise, you must have a working WebSphere Application Server installed on the machine.

Instructor exercise overview

In this short exercise, the students will complete several basic manipulations of wsadmin objects. Initially they will start wsadmin and run single wsadmin commands. Emphasize to the students that wsadmin is case sensitive. \$AdminApp is not the same as \$adminApp. (\$AdminApp is correct). There is a lot of room in this exercise for experimentation. Encourage your students to use the \$Help object to retrieve help on specific objects.

Exercise instructions

The wsadmin tool supports two scripting languages: Jacl and Jython. The first section of the lab explores wsadmin using Jacl commands. Starting wsadmin, getting help, and running various commands.

The next section of the lab explores wsadmin using Jython commands. Starting wsadmin getting help, and running various commands. If you would like to skip the Jacl section and continue with the Jython section continue to **Part 2 Using wsadmin and Jython**.

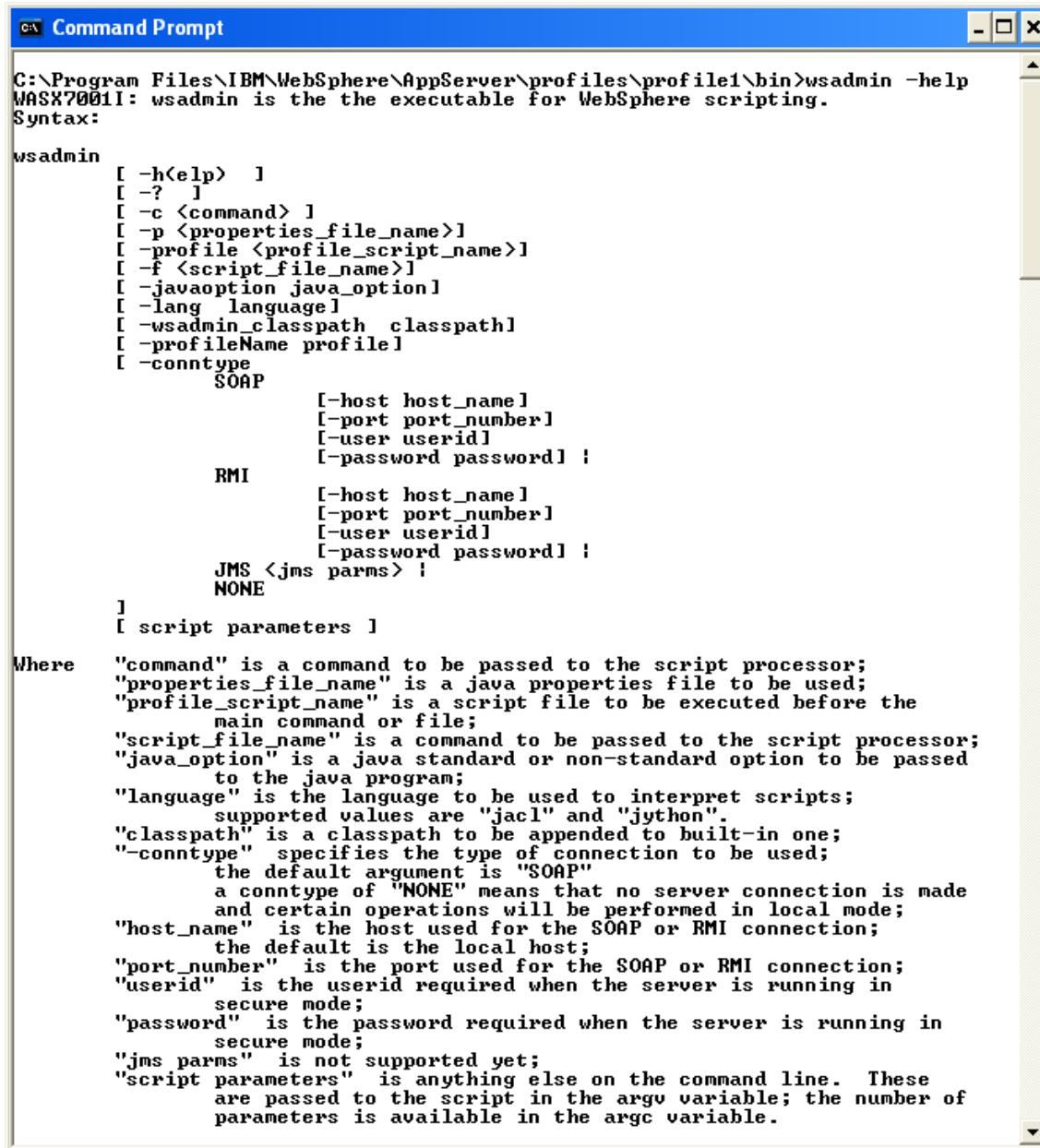
There are solution files for Jacl and Jython commands in <software_dir>\wsadmin. You can continue with the exercise or you can run the appropriate script.

Part 1 Using wsadmin and Jacl

Start wsadmin

- ___ 1. If you would like to skip the Jacl section and continue with the Jython section skip to **Part 2 Using wsadmin and Jython**. Otherwise, continue with the exercise. You can also complete both Part 1 Using wsadmin and Jacl and Part 2 Using wsadmin and Jython.
- ___ 2. Start WebSphere Application Server.
 - ___ a. Open a command prompt window and navigate to the <profile_root>\profile1\bin directory.
 - ___ b. Verify the server is running using the **serverStatus server1** command.
UNIX: Enter ./serverStatus.sh server1
 - ___ c. If the application server is not running start the server by issuing the **startServer server1** command.
UNIX: Enter ./startServer.sh server1
 - ___ d. Do not close the command prompt window.
- ___ 3. List the command line options for wsadmin.

- __ a. Get the command line help for the wsadmin script. From the <profile_root>\profile1\bin directory enter **wsadmin -help**.



```
C:\> Command Prompt
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>wsadmin -help
WSX7001I: 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 classpath ]
  [ -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 ] :
    JMS <jms parms> :
  NONE
  ]
  [ script parameters ]

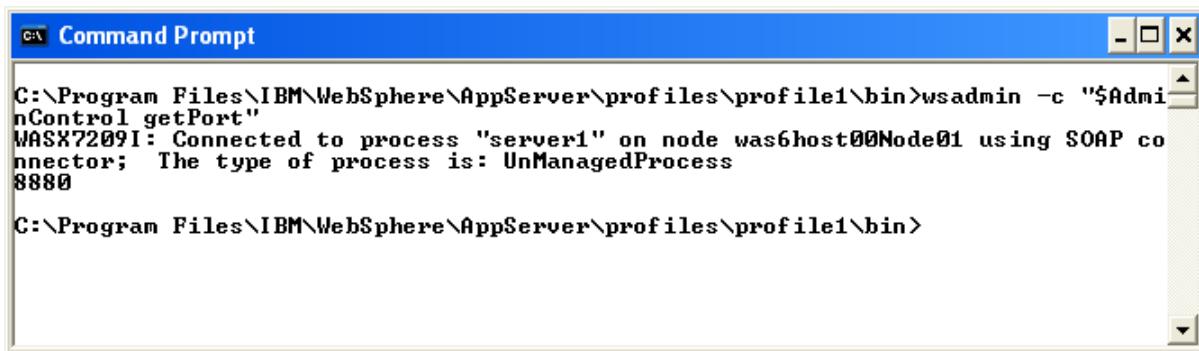
Where "command" is a command to be passed to the script processor;
"properties_file_name" is a java properties file to be used;
"profile_script_name" is a script file to be executed before the
main command or file;
"script_file_name" is a command to be passed to the script processor;
"java_option" is a java standard or non-standard option to be passed
to the java program;
"language" is the language to be used to interpret scripts;
supported values are "jac1" and "jython".
"classpath" is a classpath to be appended to built-in one;
"-conntype" specifies the type of connection to be used;
the default argument is "SOAP"
a conntype of "NONE" means that no server connection is made
and certain operations will be performed in local mode;
"host_name" is the host used for the SOAP or RMI connection;
the default is the local host;
"port_number" is the port used for the SOAP or RMI connection;
"userid" is the userid required when the server is running in
secure mode;
"password" is the password required when the server is running in
secure mode;
"jms parms" is not supported yet;
"script parameters" is anything else on the command line. These
are passed to the script in the argv variable; the number of
parameters is available in the argc variable.
```

UNIX: Enter ./wsadmin.sh -help

The resulting output displays a list of command line options and descriptions of those options.

- __ 4. It is possible to run a single wsadmin command and exit in the Jacl shell. This is not very efficient since a JVM needs to be created every time a command is run. Enter following command:

wsadmin -c "\$AdminControl getPort"



```
C:\ Command Prompt
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>wsadmin -c "$AdminControl getPort"
WASX7209I: Connected to process "server1" on node was6host00Node01 using SOAP connector; The type of process is: UnManagedProcess
8880
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>
```

UNIX: Enter ./wsadmin -c "\$AdminControl getPort"

The port which wsadmin is using to connect to the application server will be displayed.

- ___ 5. From a command prompt (not the wsadmin prompt), you can run wsadmin with the -f option. This option gives you the ability to run a script file from wsadmin. The command to run a script looks like this:

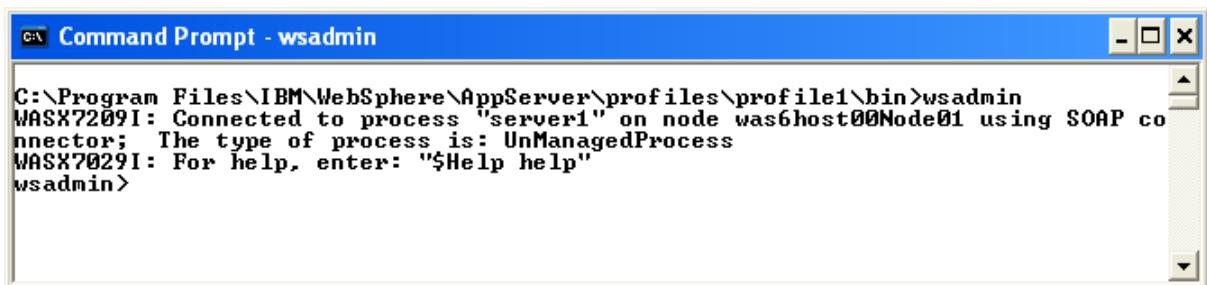
wsadmin -f <software_dir>/wsadmin/jacl/class_sample.jacl

UNIX: ./wsadmin.sh -f <software_dir>/wsadmin/jacl/class_sample.jacl

The class_sample.jacl file is a script file which includes all of the commands from the Part 1 Using wsadmin and Jacl section for Jacl commands. You can copy the commands from the file and paste into the wsadmin prompt or you can continue with the exercise.

UNIX: If using the script it will need to be modified for your environment.

- ___ 6. Start wsadmin from the command line.
 - ___ a. Ensure the command prompt window is still open on the **<profile_root>\profile1\bin** directory. Run the **wsadmin** batch file to open an interactive wsadmin session. In the command prompt window type **wsadmin**.



```
C:\ Command Prompt - wsadmin
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>wsadmin
WASX7209I: Connected to process "server1" on node was6host00Node01 using SOAP connector; The type of process is: UnManagedProcess
WASX7029I: For help, enter: "$Help help"
wsadmin>
```

UNIX: Enter ./wsadmin.sh

Information: When security is enabled a userid and password must be supplied when using wsadmin. From the command prompt issue **wsadmin -username <userid> -password <password>**.

- ___ b. You should receive the wsadmin prompt. At this point, you are ready to begin working with **wsadmin** interactively.

Work with wsadmin objects

- 1. Use the Help object to get information about the AdminApp, AdminControl, AdminConfig and AdminTask objects.

Information: The Help object is used to provide general help for the objects AdminApp, AdminConfig, AdminControl, AdminTask and Help. It is also the interface to obtain information about MBeans (operations, attributes and particular interface information about MBeans).

When referencing wsadmin objects prepend their names with the dollar “\$” sign.

- a. At the wsadmin command prompt enter the following commands. Each command will provide an overview of the object's capabilities and a list of valid commands to use with the object.

```
$Help AdminApp
$Help AdminControl
$Help AdminConfig
$Help AdminTask
```

```
wsadmin>$Help AdminApp
WASX7095I: The AdminApp object allows application objects to
be manipulated -- this includes installing, uninstalling, editing,
and listing. Most of the commands supported by AdminApp operate in two
modes: the default mode is one in which AdminApp communicates with the
WebSphere server to accomplish its tasks. A local mode is also
possible, in which no server communication takes place. The local
mode of operation is invoked by bringing up the scripting client with
no server connected using the command line "--conntype NONE" option
or setting the "com.ibm.ws.scripting.connectionType=NONE" property in
the wsadmin.properties.

The following commands are supported by AdminApp; more detailed
information about each of these commands is available by using the
"help" command of AdminApp and supplying the name of the command
as an argument.

deleteUserAndGroupEntries      Deletes all the user/group information for all
                               the roles and all the username/password information for RunAs
                               roles for a given application.
edit                          Edit the properties of an application
editInteractive               Edit the properties of an application interactively
export                        Export application to a file
exportDDL                     Export DDL from application to a directory
help                          Show help information

install                       Installs an application, given a file name and an option string.
```

- ___ b. More detailed information about each command is available using the “help” command. Enter the following command **\$AdminApp help edit** for more information on the edit object.

```
wsadmin>$AdminApp help edit
WASX7104I: Method: edit

    Arguments: application name, options

    Description: Modifies the application specified by "application name"
                  using the options specified by "options". The user is not prompted for
                  any information.

wsadmin>
```

- ___ 2. Use the AdminApp object to list applications and application module information.

Information: The AdminApp object is used to work with application objects. This includes functions such as installing, uninstalling, listing and editing.

- ___ a. At the wsadmin command prompt enter **\$AdminApp list** and all applications installed on the application server will be listed.

```
wsadmin>$AdminApp list
DefaultApplication
TradeApplication
ivtApp
query
wsadmin>
```

- ___ b. At the wsadmin command prompt, enter **\$AdminApp listModules TradeApplication**. A list of the installed Web and EJB Modules for the TradeApplication enterprise application will be displayed.

```
wsadmin>$AdminApp listModules TradeApplication
TradeApplication#TradeWeb.war+WEB-INF/web.xml
TradeApplication#TradeEJB.jar+META-INF/ejb-jar.xml
wsadmin>
```

- ___ 3. Use the AdminControl object to get information about the domain, cell and host.

Information: The AdminControl object is used to invoke operational commands on “live” running objects. It supports utility methods for tracing, reconnecting the server and converting data types.

- ___ a. At the wsadmin command prompt, enter **\$AdminControl getCell**. The cell name will be displayed.

```
Command Prompt - wsadmin
wsadmin>$AdminControl getCell
was6host00Node01Cell
wsadmin>
```

- ___ b. At the wsadmin command prompt, enter **\$AdminControl getNode**. The node name will be displayed.

```
Command Prompt - wsadmin
wsadmin>$AdminControl getNode
was6host00Node01
wsadmin>
```

- ___ c. At the wsadmin command prompt, enter **\$AdminControl getHost**. The host name of the machine connected to will be displayed.

```
Command Prompt - wsadmin
wsadmin>$AdminControl getHost
localhost
wsadmin>
```

- ___ 4. Use the AdminConfig object to modify the static configuration of a JDBC Provider. You will modify the description of the JDBC Provider in multiple steps using JACL commands, wsadmin commands and one variable.

Information: The AdminConfig object manipulates the static configuration data for a WebSphere installation for all objects EXCEPT applications. There are commands to list, create, remove, display and modify configuration data.

- __ a. Verify the name of the JDBC Provider. At the wsadmin command prompt, enter **\$AdminConfig list JDBCProvider**. The name of the JDBC Provider will be displayed.

```
wsadmin>$AdminConfig list JDBCProvider
"Cloudscape JDBC Provider <XA>(cells/was6host00Node01Cell/nodes/was6host00Node01
/servers/server1/resources.xml#builtin_jdbcprovider)"
"Cloudscape JDBC Provider <XA>(cells/was6host00Node01Cell/resources.xml#builtin_
jdbcprovider)"
"Cloudscape JDBC Provider<cells/was6host00Node01Cell/nodes/was6host00Node01/serv
ers/server1/resources.xml#JDBCProvider_1110204627176>"
Trade(cells/was6host00Node01Cell/applications/TradeApplication.ear/deployments/T
radeApplication/resources.xml#JDBCProvider_1103737547632)
wsadmin>
```

- __ b. At the wsadmin command prompt, enter the following:

set jdbc [\$AdminConfig getid /JDBCProvider:Trade/]

set is a JACL command

jdbc is a variable name

getid is a \$AdminConfig command that retrieves the configuration id of the JDBC Provider object

/JDBCProvider:Trade/ is the hierarchical containment path of the configuration object, including the actual name of the object.

or **set jdbc [\$AdminConfig getid "/JDBCProvider:<name of provider>/"]** if your JDBC Provider name is different from Trade. Quotes are needed if there are spaces in the name of the object.

\$AdminConfig show \$jdbc

show is a \$AdminConfig command that displays the values of the object *jdbc*

```
wsadmin>set jdbc1 [$AdminConfig getid /JDBCProvider:Trade/]
Trade(cells/was6host00Node01Cell/applications/TradeApplication.ear/deployments/T
radeApplication/resources.xml#JDBCProvider_1103737547632)
wsadmin>
wsadmin>$AdminConfig show $jdbc1
{classpath "C:\Program Files\IBM\SQLLIB\java\db2java.zip"}
{description "DB2 Legacy CLI-based Type 2 JDBC Driver <XA>"}
{implementationClassName COM.ibm.db2.jdbc.DB2XADataSource}
{name Trade}
{nativepath {}}
{xa false}
wsadmin>
```

\$AdminConfig modify \$jdbc {{description "Trade Application JDBC Provider (XA)"}}

modify is a \$AdminConfig command that changes the description of the JDBC Provider in the configuration id (which is stored in the variable *jdbc*) to the value “Trade Application JDBC Provider (XA)”

description is an attribute of server objects

\$AdminConfig show \$jdbc

show is a \$AdminConfig command that displays the values of the object *jdbc*

```
wsadmin>$AdminConfig modify $jdbc1 {{description "Trade JDBC Provider (XA)"}}
wsadmin>$AdminConfig show $jdbc1
<classpath "C:\Program Files\IBM\SQLLIB\java\db2java.zip">
<description "Trade JDBC Provider (XA)">
<implementationClassName COM.ibm.db2.jdbc.DB2XADataSource>
<name Trade>
<nativelpath {}>
<xa false>
wsadmin>
```

\$AdminConfig save

This command saves the configuration to the repository.

- ___ c. List the data sources used by the installed applications. At the wsadmin command prompt, enter **\$AdminConfig list DataSource**. The name of the DataSources will be displayed.

```
wsadmin>$AdminConfig list DataSource
"Default Datasource<cells/was6host00Node01Cell/nodes/was6host00Node01/servers/server1/resources.xml#DataSource_1110204628929>"
DefaultEJBTimerDataSource<cells/was6host00Node01Cell/nodes/was6host00Node01/servers/server1/resources.xml#DataSource_1000001>
Trade<cells/was6host00Node01Cell/applications/TradeApplication.ear/deployments/TradeApplication/resources.xml#DataSource_1103737547602>
wsadmin>
```

- ___ 5. Use the AdminTask object to get information about the node and server.

Information: The AdminTask object is used to run an administrative command. Administrative commands are discovered dynamically when you start wsadmin.

- ___ a. At the wsadmin command prompt, enter the following commands to obtain information about the node and server:

set nodes [\$AdminTask listNodes]

set is a JACL command

nodes is a variable name

listNodes is a \$AdminTask command that displays all of the nodes in the cell

\$AdminTask listServerTypes \$nodes

listServerTypes is a \$AdminTask command that lists server types for the value of *nodes*

\$AdminTask listServers

listServer is a \$AdminTask command that lists the servers

```
wsadmin>set nodes [$AdminTask listNodes]
was6host00Node01
wsadmin>$AdminTask listServerTypes $nodes
APPLICATION_SERVER
WEB_SERVER
GENERIC_SERVER
wsadmin>$AdminTask listServers
server1(cells/was6host00Node01Cell/nodes/was6host00Node01/servers/server1!server
.xml)
wsadmin>
```

___ 6. Use various command to get information on the configuration and server.

___ a. At the wsadmin command prompt, enter the following commands:

set cell [\$AdminConfig list Cell]

set is a JACL command

cell is a variable name

list is a \$AdminConfig command that displays the cell

set cellname [\$AdminConfig showAttribute \$cell name]

set is a JACL command

cellname is a variable name

showAttribute is a \$AdminConfig command that displays the value of the object *cell*

set nodes [\$AdminConfig list Node \$cell]

set is a JACL command

nodes is a variable name

list is a \$AdminConfig command that displays all of the nodes in *cell*

set nodename [\$AdminConfig showAttribute \$nodes name]

set is a JACL command

nodename is a variable name

showAttribute is a \$AdminConfig command that displays the value of the attributes for the object *nodes*

\$AdminConfig showall [\$AdminConfig list Node]

showall is a \$AdminConfig command that displays the configuration of the nodes in the cell

set server [\$AdminConfig list Server]

set is a JACL command

server is a variable name

list is a \$AdminConfig command that displays all of the servers

\$AdminTask showServerInfo \$server

nodes is a variable name

showServerInfo is a \$AdminTask command that displays all of the information for the *server*

```

C:\ Command Prompt - wsadmin
wsadmin>set cell [$AdminConfig list Cell]
was6host00Node01Cell<cells/was6host00Node01Cell!cell.xml#Cell_1>
wsadmin>set cellname [$AdminConfig showAttribute $cell name]
was6host00Node01Cell
wsadmin>set nodes [$AdminConfig list Node $cell]
was6host00Node01<cells/was6host00Node01Cell/nodes/was6host00Node01!node.xml#Node_1>
wsadmin>set nodename [$AdminConfig showAttribute $nodes name]
was6host00Node01
wsadmin>$AdminConfig showall [$AdminConfig list Node]
<DiscoveryProtocol TCP>
<HostName was6host00>
<Name was6host00Node01>
<Properties {}>
wsadmin>set svr [$AdminConfig list Server]
server1<cells/was6host00Node01Cell/nodes/was6host00Node01/servers/server1!server.xml#Server_1110204554262>
wsadmin>$AdminTask showServerInfo $svr
<cell was6host00Node01Cell>
<serverType APPLICATION_SERVER>
<com.ibm.websphere.baseProductVersion 6.0.0.1>
<node was6host00Node01>
<server server1>
wsadmin>
```

Work with applications

Use wsadmin and JACL commands to uninstall the ivtApp application. The application must first be stopped. Once an application is stopped you can uninstall the application.

- ___ 1. Enter the following commands that will stop the ivtApp application and uninstall the application.

```
set appManager [$AdminControl queryNames  
type=ApplicationManager,cell=$cellname,node=$nodename,process  
=server1,*]
```

This command gets the name of the application manager Mbean for the applications running on the server.

```
set app [$AdminControl queryNames  
type=Application,cell=$cellname,node=$nodename,process=server1  
,J2EEName=ivtApp,*]
```

This command gets the application detail and assigns it to a variable.

set appName [\$AdminControl getAttribute \$app name]

This command gets the application name and assigns it to a variable.

\$AdminControl invoke \$appManager stopApplication \$appName

This command stops the ivtApp.

\$AdminApp uninstall \$appName

This command uninstalls the ivtApp.

Information: Look for the message indicating the application ivtApp uninstalled successfully.

\$AdminConfig save

Save the configuration to the repository. This command will update the repository with the information about the uninstalled **ivtApp** application.

```
Command Prompt - wsadmin
wsadmin>set appManager [$AdminControl queryNames type=ApplicationManager,cell=$cellname,node=$nodename,process=server1,*]
WebSphere:platform=dynamicproxy,cell=was6host00Node01Cell,version=6.0.0.1,name=ApplicationManager,mbeanIdentifier=ApplicationManager,type=ApplicationManager,node=was6host00Node01,process=server1
wsadmin>
wsadmin>set app [$AdminControl queryNames type=Application,cell=$cellname,node=$nodename,process=server1,J2EEName=ivtApp,*]
WebSphere:name=ivtApp,process=server1,platform=dynamicproxy,node=was6host00Node01,J2EEName=ivtApp,Server=server1,version=6.0.0.1,type=Application,mbeanIdentifier=cells/was6host00Node01Cell/applications/ivtApp.ear/deployments/ivtApp/deployment.xml#ApplicationDeployment_1110204646124,cell=was6host00Node01Cell
wsadmin>set appName [$AdminControl getAttribute $app name]
ivtApp
wsadmin>$AdminControl invoke $appManager stopApplication $appName
wsadmin>$AdminApp uninstall ivtApp
ADMA5017I: Uninstallation of ivtApp started.
ADMA5104I: The server index entry for WebSphere:cell=was6host00Node01Cell,node=was6host00Node01 is updated successfully.
ADMA5102I: The configuration data for ivtApp from the configuration repository is deleted successfully.
ADMA5011I: The cleanup of the temp directory for application ivtApp is complete.
ADMA5106I: Application ivtApp uninstalled successfully.
wsadmin>$AdminConfig save
wsadmin>
```

- ___ 2. Verify the application has been uninstalled by using the administrative console.
 - ___ a. Log in to the administrative console using the userid **admin**.
 - ___ b. Expand **Applications** and select **Enterprise Applications**. The ivtApp application should no longer be listed.

Enterprise Applications

Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Preferences

Start	Stop	Install	Uninstall	Update	Rollout Update	Remove File	Export	Export DDL
<input checked="" type="checkbox"/>								
Select	Name	Status						
<input type="checkbox"/>	DefaultApplication							
<input type="checkbox"/>	TradeApplication							
<input type="checkbox"/>	query							
Total 3								

- ___ 3. Install the ivtApp application and verify that it is running. The ivtApp.ear file can be found in the <was_root>\installableApps directory.

- __ a. Copy the ivtApp.ear file from the <was_root>\installableApps directory to the <profile_root>\profile1\installableApps directory.
- __ b. Enter the following commands that will install the ivtApp application and start the application.

\$AdminApp install ..\installableApps\ivtApp.ear {-appname ivtApp}

This command installs the ivtApp.ear and gives it the name ivtApp.
-appname is an installation option.

Information: Look for the message indicating the application ivtApp installed successfully.

\$AdminConfig save

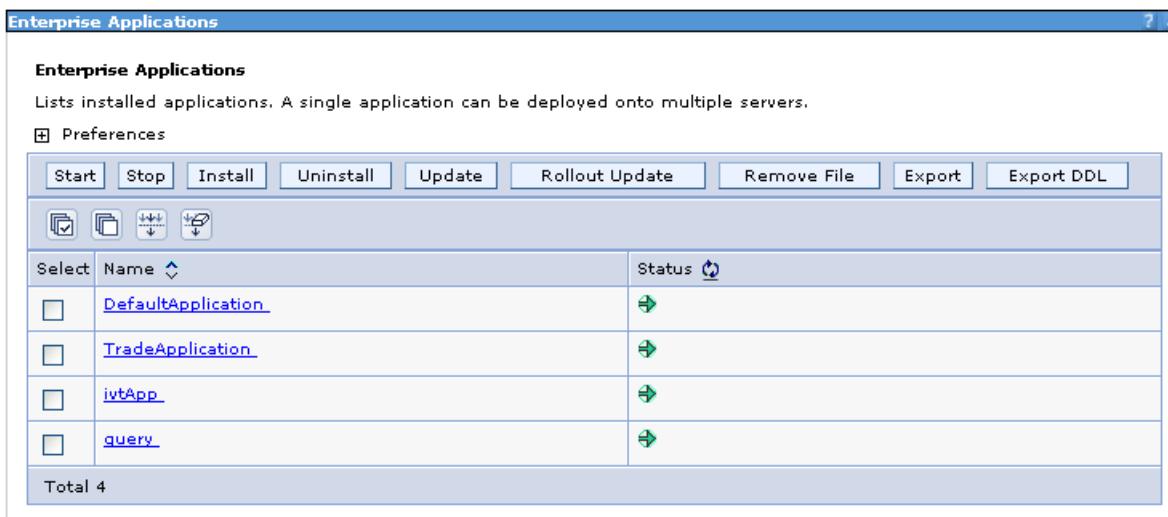
This command saves the configuration to the repository.

\$AdminControl invoke \$appManager startApplication \$appName

This command will start the newly installed application.

- __ 4. Verify the application has been installed by using the administrative console.
 - __ a. Log in to the administrative console, if needed, or refresh the current administrative console.
 - __ b. Expand **Applications** and select **Enterprise Applications**. The ivtApp application should be listed and running.

Enterprise Applications



- __ c. Logout of the administrative console.
- __ d. Exit from the wsadmin prompt.
- __ 5. Look through the Info Center example scripts.
 - __ a. Take a look in the directory <software_dir>\wsadmin\jac\samples for a number of scripts name ex1.jacl through ex12.jacl. These are sample scripts

taken directly from the Info Center. They demonstrate a number of very useful operations that might require scripting. Here is a list of the scripts and a brief description of what they do.

Example	Description
ex1	create server, modify, load app start server
ex2	stop server, uninstall app, remove server
ex3	create cluster, install app on it, create cluster members, sync nodes, start cluster
ex4	setting traces and dumping threads
ex5	various app install options
ex6	display summary information about the installation: servers, apps, and so forth
ex7	use a template to create a JDBCProvider and DataSource; modify DataSource
ex8	J2C security; installing J2CResourceAdapter and creating J2CConnectionFactory
ex9	create JDBCProvider using template; create CMPConnectionFactory
ex10	create WebSphere variable; create URLProvider and URLs using this variable
ex11	modify WebSphere ports in the serverindex.xml file.
ex12	modifying attributes that are lists

- ___ b. Using a text editor, go through each of these examples. These examples will not only demonstrate various WebSphere commands, but also expose some of the Jacl structures.
- ___ 6. The default scripting language for wsadmin is Jacl. If Jython is preferred, the following command would be used for a one time switch to use Jython instead of Jacl:

wsadmin -lang jython

Information: To always use Jython as the scripting language edit the **wsadmin.properties** file found in the <profile_root>\profile1\properties directory.

Look for the entry that defines **com.ibm.ws.scripting.defaultLang=jacl** and set it equal to Jython. This change will be global to this profile. It is also possible to define a property file for individual users or through an environment variable.

UNIX: Enter **./wsadmin.sh -lang jython**

Part 2 Using wsadmin and Jython

Start wsadmin

- ___ 1. Start WebSphere Application Server.
 - ___ a. Open a command prompt window and navigate to the **<profile_root>\profile1\bin** directory.
 - ___ b. Verify the server is running using the **serverStatus server1** command.
UNIX: Enter **./serverStatus.sh server1**
 - ___ c. If the application server is not running start the server by issuing the **startServer server1** command.
UNIX: Enter **./startServer.sh server1**
 - ___ d. Do not close the command prompt window.
- ___ 2. List the command line options for wsadmin.

- ___ a. Get the command line help for the wsadmin script. From the <profile_root>\profile1\bin directory enter **wsadmin -help**

```
C:\> Command Prompt
C:\> Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>wsadmin -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 classpath ]
[ -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] !
    JMS <jms parms> !
    NONE
]
[ script parameters ]

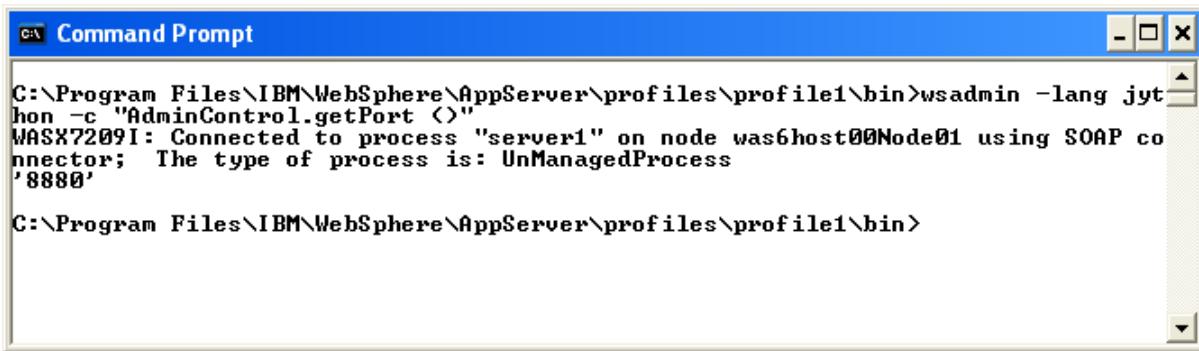
Where "command" is a command to be passed to the script processor;
"properties_file_name" is a java properties file to be used;
"profile_script_name" is a script file to be executed before the
main command or file;
"script_file_name" is a command to be passed to the script processor;
"java_option" is a java standard or non-standard option to be passed
to the java program;
"language" is the language to be used to interpret scripts;
supported values are "jacl" and "jython".
"classpath" is a classpath to be appended to built-in one;
"-conntype" specifies the type of connection to be used;
the default argument is "SOAP"
a conntype of "NONE" means that no server connection is made
and certain operations will be performed in local mode;
"host_name" is the host used for the SOAP or RMI connection;
the default is the local host;
"port_number" is the port used for the SOAP or RMI connection;
"userid" is the userid required when the server is running in
secure mode;
"password" is the password required when the server is running in
secure mode;
"jms parms" is not supported yet;
"script parameters" is anything else on the command line. These
are passed to the script in the argv variable; the number of
parameters is available in the argc variable.
```

UNIX: Enter ./wsadmin.sh -help

The resulting output displays a list of command line options and descriptions of those options.

- ___ 3. It is possible to run a single wsadmin command and exit in the Jython shell. This is not very efficient since a JVM needs to be created every time a command is run.
Enter following command:

wsadmin -lang jython -c "AdminControl.getPort()"



```
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>wsadmin -lang jython -c "AdminControl.getPort ()"
WASX7209I: Connected to process "server1" on node was6host00Node01 using SOAP connector; The type of process is: UnManagedProcess
'8880'

C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>
```

UNIX: Enter ./wsadmin.sh -lang jython -c "AdminControl.getPort()"

The port which wsadmin is using to connect to the application server will be displayed.

- ___ 4. From a command prompt (not the wsadmin prompt), you can run wsadmin with the -f option. This option gives you the ability to run a script file from wsadmin. The command to run a script looks like this:

```
wsadmin -lang jython -f
<software_dir>/wsadmin/jython/class_sample.py
```

UNIX: /wsadmin.sh -lang jython -f <software_dir>/wsadmin/jython/class_sample.py

The class_sample.pyl file is a script file which includes all of the commands from the Part 2 Using wsadmin and Jython section for Jython commands. You can copy the commands from the file and paste into the wsadmin prompt or you can continue with the exercise.

UNIX: If you use the script, it will need to be modified for your environment.

- ___ 5. Start wsadmin from the command line using Jython.
 - ___ a. Ensure the command prompt window is still open on the <profile_root>\profile1\bin directory. The default scripting language for wsadmin is Jacl. Run the **wsadmin** batch file and change the language to start an interactive wsadmin session. If Jython is preferred, the following command would be used for a one time switch to use Jython instead of Jacl:

wsadmin -lang jython

```
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>wsadmin -lang jython
WASX7209I: Connected to process "server1" on node was6host00Node01 using SOAP connector; The type of process is: UnManagedProcess
WASX7031I: For help, enter: "print Help.help()"
wsadmin>
```

UNIX: Enter ./wsadmin.sh -lang jython

Information: To always use jython as the scripting language follow these steps. Using a text editor, edit the **wsadmin.properties** file found in the **<profile_root>\profile1\properties** directory.

Look for the entry that defines **com.ibm.ws.scripting.defaultLang=jacl** and set it equal to jython. This change will be global to this profile. It is also possible to define a property file for individual users or through an environment variable.

Information: When security is enabled a userid and password must be supplied when using wsadmin. From the command prompt issue **wsadmin -username <userid> -password <password>**.

- ___ b. You should receive the wsadmin prompt. At this point, you are ready to begin working with **wsadmin** interactively.

Work with wsadmin objects

- 1. Use the Help object to get information about the AdminApp, AdminControl, AdminConfig and AdminTask objects.

Information: The Help object is used to provide general help for the objects AdminApp, AdminConfig, AdminControl, AdminTask and Help. It is also the interface to obtain information about MBeans (operations, attributes and particular interface information about MBeans).

- a. At the wsadmin command prompt enter the following commands. Each command will provide an overview of the object's capabilities and a list of valid commands to use with the object.

```
print Help.AdminApp()
print Help.AdminControl()
print Help.AdminConfig()
print Help.AdminTask()
```

```
wsadmin>print Help.AdminApp()
WASX7095I: The AdminApp object allows application objects to
be manipulated -- this includes installing, uninstalling, editing,
and listing. Most of the commands supported by AdminApp operate in two
modes: the default mode is one in which AdminApp communicates with the
WebSphere server to accomplish its tasks. A local mode is also
possible, in which no server communication takes place. The local
mode of operation is invoked by bringing up the scripting client with
no server connected using the command line "-connectionType NONE" option
or setting the "com.ibm.ws.scripting.connectionType=NONE" property in
the wsadmin.properties.

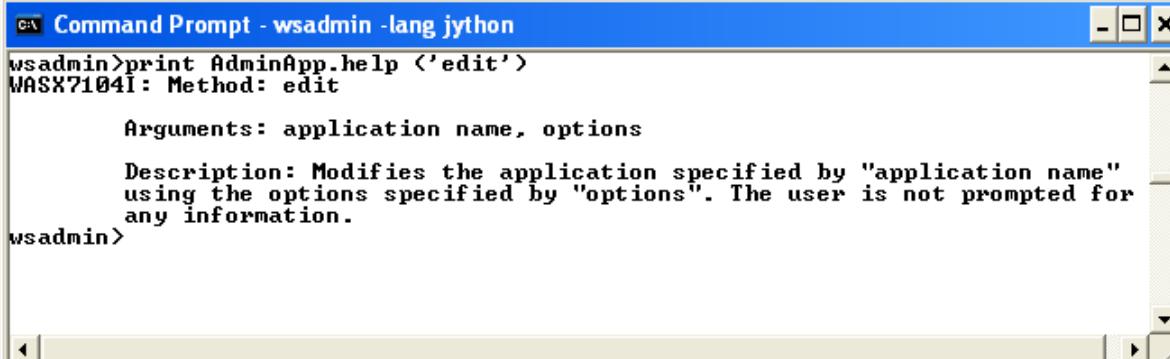
The following commands are supported by AdminApp; more detailed
information about each of these commands is available by using the
"help" command of AdminApp and supplying the name of the command
as an argument.

deleteUserAndGroupEntries
    Deletes all the user/group information for all
    the roles and all the username/password information for RunAs
    roles for a given application.
edit
    Edit the properties of an application
editInteractive
    Edit the properties of an application interactively
export
    Export application to a file
exportDDL
    Export DDL from application to a directory
help
    Show help information

install
    Installs an application, given a file name and an option string.

installInteractive
    Installs an application in interactive mode, given a
    file name and an option string.
list
    List all installed applications
listModules
    List the modules in a specified application
options
    Shows the options available, for a given file, application,
    or in general.
publishWSDL
    Publish WSDL files for a given application
searchJNDIReferences
```

- ___ b. More detailed information about each command is available using the “help” command. Enter the following command **print AdminApp.help('edit')** for more information on the edit object.



```
Command Prompt - wsadmin -lang jython
wsadmin>print AdminApp.help ('edit')
WASX7104I: Method: edit

    Arguments: application name, options

    Description: Modifies the application specified by "application name"
                  using the options specified by "options". The user is not prompted for
                  any information.

wsadmin>
```

- ___ 2. Use the AdminApp object to list applications and application module information.

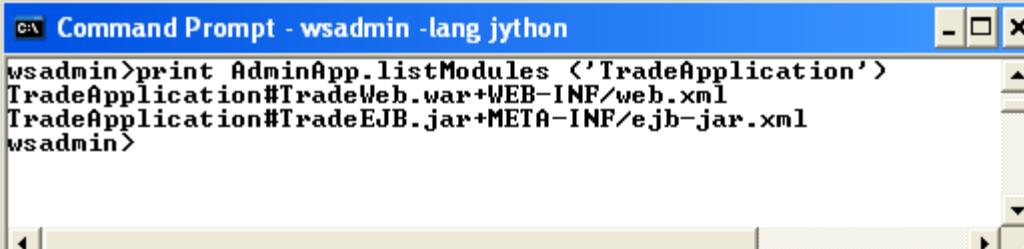
Information: The AdminApp object is used to work with application objects. This includes functions such as installing, uninstalling, listing and editing.

- ___ a. At the wsadmin command prompt enter **print AdminApp.list()** and all applications installed on the application server will be listed.



```
Command Prompt - wsadmin -lang jython
wsadmin>print AdminApp.list()
DefaultApplication
TradeApplication
ivtApp
query
wsadmin>
```

- ___ b. At the wsadmin command prompt, enter **print AdminApp.listModules('TradeApplication')**. A list of the installed Web and EJB Modules for the TradeApplication enterprise application will be displayed.



```
Command Prompt - wsadmin -lang jython
wsadmin>print AdminApp.listModules ('TradeApplication')
TradeApplication#TradeWeb.war+WEB-INF/web.xml
TradeApplication#TradeEJB.jar+META-INF/ejb-jar.xml
wsadmin>
```

- ___ 3. Use the AdminControl object to get information about the domain, cell and host.

Information: The AdminControl object is used to invoke operational commands on "live" running objects. It supports utility methods for tracing, reconnecting the server and converting data types.

- ___ a. At the wsadmin command prompt, enter **print AdminControl.getCell()**. The cell name will be displayed.

```
C:\ Command Prompt - wsadmin -lang jython
wsadmin>print AdminControl.getCell()
was6host00Node01Cell
wsadmin>
```

- ___ b. At the wsadmin command prompt, enter **print AdminControl.getNode()**. The node name will be displayed.

```
C:\ Command Prompt - wsadmin -lang jython
wsadmin>print AdminControl.getNode()
was6host00Node01
wsadmin>
```

- ___ c. At the wsadmin command prompt, enter **print AdminControl.getHost()**. The host name of the machine connected to will be displayed.

```
C:\ Command Prompt - wsadmin -lang jython
wsadmin>print AdminControl.getHost()
localhost
wsadmin>
```

- ___ 4. Use the AdminConfig object to modify the static configuration of a JDBC Provider. You will modify the description of the JDBC Provider in multiple steps using Jacl commands, wsadmin commands and one variable.

Information: The AdminConfig object manipulates the static configuration data for a WebSphere installation for all objects EXCEPT applications. There are commands to list, create, remove, display and modify configuration data.

- __ a. Verify the name of the JDBC Provider. At the wsadmin command prompt, enter **print AdminConfig.list('JDBCProvider')**. The name of the JDBC Provider will be displayed.

```
C:\ Command Prompt - wsadmin -lang jython
wsadmin>print AdminConfig.list('JDBCProvider')
"Cloudscape JDBC Provider <XA>(cells/was6host00Node01Cell/nodes/was6host00Node01
/servers/server1/resources.xml#builtin_jdbcprovider)"
"Cloudscape JDBC Provider <XA>(cells/was6host00Node01Cell/resources.xml#builtin_
jdbcprovider)"
"Cloudscape JDBC Provider<cells/was6host00Node01Cell/nodes/was6host00Node01/serv
ers/server1/resources.xml#JDBCProvider_1110204627176>"
Trade(cells/was6host00Node01Cell/applications/TradeApplication.ear/deployments/T
radeApplication/resources.xml#JDBCProvider_1103737547632)
wsadmin>
```

- __ b. At the wsadmin command prompt, enter the following:

jdbc=AdminConfig.getid('/JDBCProvider:Trade/')

jdbc is a variable name

getid is an AdminConfig command that retrieves the configuration id of the JDBC Provider object

/JDBCProvider:Trade/ is the hierarchical containment path of the configuration object, including the actual name of the object.

or **jdbc=AdminConfig.getid('/JDBCProvider:<name of provider>/')** if your JDBC Provider name is different from Trade.

print AdminConfig.show(jdbc)

show is an AdminConfig command that displays the value of the object *jdbc*

```
C:\ Command Prompt - wsadmin -lang jython
wsadmin>jdbc1=AdminConfig.getId('/JDBCProvider:Trade/')
wsadmin>print AdminConfig.show(jdbc1)
[classpath "C:\Program Files\IBM\SQLLIB\java\db2java.zip"]
[description "DB2 Legacy CLI-based Type 2 JDBC Driver <XA>"]
[implementationClassName COM.ibm.db2.jdbc.DB2XADataSource]
[name Trade]
[nativepath []]
[xa false]
wsadmin>
```

AdminConfig.modify(jdbc, [['description', "Trade Application JDBC Provider (XA)']])

modify is an AdminConfig command that changes the description of the JDBC Provider in the configuration id (which is stored in the variable *jdbc*) to the value "Trade Application JDBC Provider (XA)"

description is an attribute of server objects

Information: If you completed Part 1 Using wsadmin and JACL the description may already be set to "Trade Application JDBC Provider (XA)". Modify the description to the value "Trade App JDBC Provider (XA)".

print AdminConfig.show(jdbc)

show is an AdminConfig command that displays the value of the object *jdbc*

```
Command Prompt - wsadmin -lang jython
wsadmin>AdminConfig.modify <jdbc1, [[{'description': 'Trade JDBC Provider (XA)'}]]
>,
wsadmin>print AdminConfig.show <jdbc1>
[classpath "C:\Program Files\IBM\SQLLIB\java\db2java.zip"]
[description "Trade JDBC Provider (XA)"]
[implementationClassName COM.ibm.db2.jdbc.DB2XADataSource]
[name Trade]
[nativepath []]
[xa false]
wsadmin>
```

AdminConfig.save()

This command saves the configuration to the repository.

- ___ c. List the data sources used by the installed applications. At the wsadmin command prompt, enter **print AdminConfig.list('DataSource')**. The name of the data sources will be displayed.
- ___ 5. Use the AdminTask object to get information about the node and server.

Information: The AdminTask object is used to run an administrative command. Administrative commands are discovered dynamically when you start wsadmin.

- ___ a. At the wsadmin command prompt, enter the following commands to information about the node and server:

nodes=AdminTask.listNodes()

nodes is a variable name

listNodes is an AdminTask command that displays all of the nodes in the cell

print AdminTask.listServerTypes(nodes)

listServerTypes is an AdminTask command that lists server types for the value of *nodes*

print AdminTask.listServers()

listServer is an AdminTask command that lists the servers

```
wsadmin>nodes=AdminTask.listNodes ()
wsadmin>print AdminTask.listServerTypes (nodes)
APPLICATION_SERVER
WEB_SERVER
GENERIC_SERVER
wsadmin>print AdminTask.listServers ()
server1<cells/was6host00Node01Cell/nodes/was6host00Node01/servers/server1!server.xml>
wsadmin>
```

___ 6. Use various command to get information on the configuration and server.

___ a. At the wsadmin command prompt, enter the following commands:

cell=AdminConfig.list('Cell')

print cell

cell is a variable name

list Cell is an AdminConfig command that displays the cell

cellname=AdminConfig.showAttribute(cell, 'name')

print cellname

cellname is a variable name

listAttribute is an AdminConfig command that displays all of the attributes of the cell

nodes=AdminConfig.list('Node', cell)

print nodes

nodes is a variable name

list is an AdminConfig command that displays all of the nodes in the cell

nodename=AdminConfig.showAttribute(nodes, 'name')

print nodename

nodename is a variable name

showAttribute is an AdminConfig command that displays all of the attributes of the nodes in the cell

print AdminConfig.showall(AdminConfig.list ('Node'))

node is a variable name

list is an AdminConfig command that displays all of the nodes in the cell

```
server=AdminConfig.list('Server')
```

print server

server is a variable name

/list is an AdminConfig command that displays the servers in the cell

AdminTask.showServerInfo(server)

showServerInfo is an AdminTask command that shows the information on the server

```
Command Prompt - wsadmin -lang python
wsadmin>cell=AdminConfig.list('Cell')
wsadmin>print cell
was6host00Node01Cell<cells/was6host00Node01Cell:cell.xml#Cell_1>
wsadmin>cellname=AdminConfig.showAttribute<cell, 'name'>
wsadmin>print cellname
was6host00Node01Cell
wsadmin>nodes=AdminConfig.list('Node', cell)
wsadmin>print nodes
was6host00Node01<cells/was6host00Node01Cell/nodes/was6host00Node01 :node.xml#Node_1>
wsadmin>nodename=AdminConfig.showAttribute<nodes, 'name'>
wsadmin>print nodename
was6host00Node01
wsadmin>print AdminConfig.showall(AdminConfig.list ('Node'))
[discoveryProtocol TCP]
[HostName was6host00]
[name was6host00Node01]
[properties []]
wsadmin>server=AdminConfig.list('Server')
wsadmin>print server
server1<cells/was6host00Node01Cell/nodes/was6host00Node01/servers/server1:server.xml#Server_1110204554262>
wsadmin>AdminTask.showServerInfo(server)
'[ [node was6host00Node01] [serverType APPLICATION_SERVER] [com.ibm.websphere.ba
seProductVersion 6.0.0.1] [cell was6host00Node01Cell] [server server1] ]'
wsadmin>
```

Work with applications

Use wsadmin and Jython commands to uninstall the ivtApp application. The application must first be stopped. Once an application is stopped you can uninstall the application.

- ___ 1. Enter the following commands that will stop the ivtApp application and uninstall the application.

```
appManager=AdminControl.queryNames('type=ApplicationManager,
cell='+cellname+',node='+nodename+',process=server1,*')
```

print appManager

This command gets the name of the application manager Mbean for the applications running on the server.

```
app=AdminControl.queryNames('type=Application,cell='+cellname+
',node='+nodename+',process=server1,J2EEName=ivtApp,*')
print app
```

This command gets the application detail and assigns it to a variable.

```
appName=AdminControl.getAttributes(app, 'name')
```

```
print appName
```

This command gets the application name and assigns it to a variable.

```
AdminControl.invoke(appManager, 'stopApplication', appName)
```

This command stops the ivtApp.

```
AdminApp.uninstall (appName)
```

This command uninstalls the ivtApp.

Information: Look for the message indicating the application ivtApp uninstalled successfully.

```
AdminConfig.save()
```

Save the configuration to the repository. This command will update the repository with the information about the uninstalled **ivtApp** application.

```
Command Prompt - wsadmin -lang python
wsadmin>appManager=AdminControl.queryNames('type=ApplicationManager,cell='+cellname+
',node='+nodename+',process=server1,*')
wsadmin>print appManager
WebSphere:platform=dynamicproxy,cell=was6host00Node01Cell,version=6.0.0.1,name=A
pplicationManager,mbeanIdentifier=ApplicationManager,type=ApplicationManager,nod
e=was6host00Node01,process=server1

wsadmin>app=AdminControl.queryNames('type=Application,cell='+cellname+',node='+
nodename+',process=server1,J2EEName=ivtApp,*')
wsadmin>print app
WebSphere:name=ivtApp,process=server1,platform=dynamicproxy,node=was6host00Node0
1,J2EEName=ivtApp,Server=server1,version=6.0.0.1,type=Application,mbeanIdentifie
r=cells/was6host00Node01Cell/applications/ivtApp.ear/deployments/ivtApp/deployme
nt.xml#ApplicationDeployment_1110302262499,cell=was6host00Node01Cell

wsadmin>appName=AdminControl.getAttribute(app, 'name')
wsadmin>print appName
ivtApp

wsadmin>AdminControl.invoke(appManager, 'stopApplication', appName)
,'

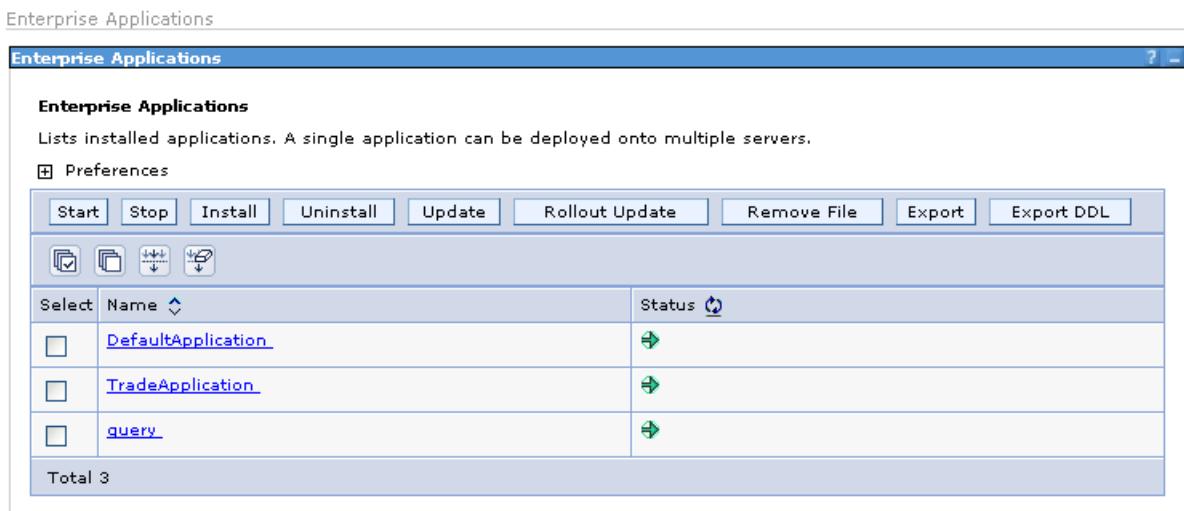
wsadmin>AdminApp.uninstall(appName)
ADMA501I: Uninstallation of ivtApp started.
ADMA5104I: The server index entry for WebSphere:cell=was6host00Node01Cell,node=w
as6host00Node01 is updated successfully.
ADMA5102I: The configuration data for ivtApp from the configuration repository i
s deleted successfully.
ADMA5011I: The cleanup of the temp directory for application ivtApp is complete.

ADMA5106I: Application ivtApp uninstalled successfully.
,'

wsadmin>AdminConfig.save()
,'

wsadmin>
```

2. Verify the application has been uninstalled by using the administrative console.
- __ a. Log in to the administrative console using the userid **admin**.
 - __ b. Expand **Applications** and select **Enterprise Applications**. The ivtApp application should no longer be listed.



3. Install the ivtApp and verify that it is running. The ivtApp.ear file can be found in the <was_root>\installableApps directory.
- __ a. Copy the ivtApp.ear file from the <was_root>\installableApps directory to the <profile_root>\profile1\installableApps directory.
 - __ b. Enter the following commands that will install the ivtApp application and start the application.

```
AdminApp.install('..../installableApps/ivtApp.ear', ['-appname ivtApp'])
```

This command installs the ivtApp.ear and gives it the name ivtApp.
The -appname is an installation option.

Information: Look for the message indicating the application ivtApp installed successfully.

AdminConfig.save()

This command saves the configuration to the repository.

AdminControl.invoke(appManager, 'startApplication', appName)

This command will start the newly installed application.

4. Verify the application has been installed by using the administrative console.
- __ a. Log in to the administrative console, if needed, or refresh the current administrative console.

- ___ b. Expand **Applications** and select **Enterprise Applications**. The ivtApp application should be listed and running.

Enterprise Applications

The screenshot shows a software interface titled "Enterprise Applications". At the top, there's a toolbar with buttons for Start, Stop, Install, Uninstall, Update, Rollout Update, Remove File, Export, and Export DDL. Below the toolbar is a toolbar with icons for file operations like Open, Save, Print, and Copy/Paste. The main area is a table with columns "Select", "Name", and "Status". The table contains four rows corresponding to the applications listed above. At the bottom of the table, it says "Total 4".

Select	Name	Status
<input type="checkbox"/>	DefaultApplication	
<input type="checkbox"/>	TradeApplication	
<input type="checkbox"/>	ivtApp	
<input type="checkbox"/>	query	

Total 4

- ___ c. Logout of the administrative console.
 ___ d. Exit from the wsadmin prompt.
 ___ 5. Look through the Info Center example scripts.
 ___ a. Take a look in the directory <software_dir>\wsadmin\jython\samples for a number of scripts name ex1.py through ex12.py. These are sample scripts taken directly from the Info Center. They demonstrate a number of very useful operations that might require scripting. Here is a list of the scripts and a brief description of what they do.

Example	Description
ex1	create server, modify, load app start server
ex2	stop server, uninstall app, remove server
ex3	create cluster, install app on it, create cluster members, sync nodes, start cluster
ex4	setting traces and dumping threads
ex5	various app install options
ex6	display summary information about the installation: servers, apps, and so forth
ex7	use a template to create a JDBCProvider and DataSource; modify DataSource
ex8	J2C security; installing J2CResourceAdapter and creating J2CConnectionFactory
ex9	create JDBCProvider using template; create CMPConnectionFactory
ex10	create WebSphere variable; create URLProvider and URLs using this variable

Example	Description
ex11	modify WebSphere ports in the serverindex.xml file.
ex12	modifying attributes that are lists

- ___ b. Using a text editor, go through each of these examples. These examples will not only demonstrate various WebSphere commands, but also expose some of the python structures.

End of lab

Exercise review and wrap-up

In this lab, you learned to manipulate wsadmin objects from the command line using both Jacl and Jython languages..

Exercise 8. Use WebSphere Rapid Deployment to install the QuoteWS Application

What this exercise is about

This lab covers the installation of the QuoteWS Application using WebSphere Rapid Deployment. In this exercise you use WRD to quickly deploy one of the component of the Trade Application, which supplies database quotes, to WebSphere Application Server V6.

What you should be able to do

At the end of the lab, you should be able to use WRD to perform the installation of an Enterprise Application packaged in an enhanced EAR file. The EAR file contains all the application modules, as any other EAR file would, and also contains the definition of other resources required by the application.

Introduction

This lab is used to install the QuoteWS Application enhanced EAR file. Enhanced EAR files are not part of the J2EE specification and are, at this time, only recognized by WebSphere Application Server V6.

Required materials

To do this lab you require a properly set up computer with WebSphere Application Server V6 installed as well as the necessary startup and program files.

Instructor exercise overview

In this lab the students install the QuoteWS.ear file using WRD in autoappinstall style. Since the enhanced ear file already contains the data source definition required for the application to run, there is no need for them to use console to define it. If the students get JNDI failures, or DB2 type exceptions, go back to the ear file, and using AST, make sure the data source definition is correct.

Make sure the Environment variable that defines the DB2_JDBC_DRIVER_PATH points to the actual folder where the db2java.zip file exists on the machine. This path varies with the operating system.

Exercise instructions

Preface

The QuoteWS application provides database quotes for the Trade application. It is a Web services provider and comes into play if the user of the Trade application has decided not to use Internet quotes, or as a failover mechanism if the user has selected to use Internet quotes but that Web service is unreachable.

In this exercise you use WRD to install the QuoteWS application to a running WebSphere Application Server V6 server.

The application's EAR file is dropped on to an **autoappinstall** WRD monitored workspace. Installation of the application and its required resources is automatic.

Step 1 Setup the WRD workspace with style autoappinstall

Before WRD can start monitoring the workspace it needs to know which directory is to be monitored and what style of monitoring is to take place.

- ___ 1. Set up the WRD workspace using style of **autoappinstall** in the **<software_dir>\WRD\QuoteWSWrk** directory.
 - ___ a. Open a command prompt window.
 - ___ b. Change to the **<profile_root>\profile1\bin** folder.
 - ___ c. Set the WORKSPACE environment variable using the following command:

```
set WORKSPACE=<software_dir>\WRD\QuoteWSWrk
```



```
C:\ Profile 1
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>set WORKSPACE=C:\Software\WRD\QuoteWSWrk
```

UNIX: Enter: `export WORKSPACE=<software_dir>/WRD/QuoteWSWrk`

- ___ d. Configure WRD deployment automation project to use the **autoappinstall** style by entering the following command:

```
wrd-config -project "QuoteWSApp" -style "autoappinstall"
```

UNIX: Enter: `./wrd-config.sh -project "QuoteWSApp" -style "autoappinstall"`

Accept all defaults presented to you by pressing **Enter** at each prompt. Wait for the **WebSphere Rapid Deployment configuration** completed message before proceeding.

```
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>wrd-config.bat -project "QuoteWSApp" -style "AutoAppInstall"
Launching WebSphere Rapid Deployment configuration. Please wait...
Starting Workbench...

Initializing the workspace...

Parameter Configuration Settings
Press ENTER to accept defaults
The * symbol denotes required input

Enter the server name* < server1 > :
Enter the server JMX host name* < localhost > :
Enter the server JMX port number* < 8880 > :
Enter your server username < -- > :
Enter your server password < -- > :

Configuring the workspace...
Building the workspace...
WebSphere Rapid Deployment configuration completed.
```

- ___ 2. Ensure that server1 in profile1 is up and running.
- ___ 3. Start monitoring the new WRD project by entering the following command:

wrd -monitor

```
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>WRD -monitor_
Launching WebSphere Rapid Deployment. Please wait...
Starting Workbench...

WebSphere Rapid Deployment ready for e-business...

Type 'q', 'quit', or 'exit' to shut down WebSphere Rapid Deployment processes
```

Information: This window remains active until you terminate the wrd process

UNIX: Enter: ./wrd.sh -monitor

- ___ 4. Copy the Enterprise application archive **QuoteWS.ear** from the:

<software_dir>\EARs and WARS

into the following folder:

<software_dir>\WRD\QuoteWSWrk\QuoteWSApp

5. Observe the output in the WRD monitor window. You should see that WRD detects that the file has been added to the workspace, it then proceeds to install the Enterprise application on server1 and finally it starts the application.

```
C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\bin>WRD -monitor_
Launching WebSphere Rapid Deployment. Please wait...
Starting Workbench...
WebSphere Rapid Deployment ready for e-business...
Type 'q', 'quit', or 'exit' to shut down WebSphere Rapid Deployment processes
WebSphere Rapid Deployment ready for e-business...

Type 'q', 'quit', or 'exit' to shut down WebSphere Rapid Deployment pr
[03:24:50 PM] [/QuoteWSApp/QuoteWS.ear] Added
[03:24:50 PM] !INSTALL_EAR_FILE QuoteWS.ear!
[03:24:50 PM] Publishing QuoteWS to server_510658053
[03:24:53 PM] Installing New Application: QuoteWS
[03:24:57 PM] ADMA5016I: Installation of QuoteWS started.
[03:25:01 PM] ADMA5058I: Application and module versions validated w
ns of deployment targets.
[03:25:01 PM] ADMA5005I: The application QuoteWS is configured in th
e Application Server repository.
[03:25:01 PM] ADMA5053I: The library references for the installed o
kage are created.
[03:25:01 PM] ADMA5005I: The application QuoteWS is configured in th
e Application Server repository.
[03:25:01 PM] ADMA5001I: The application binaries are saved in C:\Pr
s\IBM\Rational\SDP\6.0\runtimes\base_v6/profiles/default\wstemp\1025aa
space\cells\PJAKABNode01Cell\applications\QuoteWS.ear\QuoteWS.ear
[03:25:01 PM] ADMA5005I: The application QuoteWS is configured in th
e Application Server repository.
[03:25:01 PM] SECJ0400I: Successfully updated the application QuoteWS
appContextIDForSecurity information.
[03:25:01 PM] ADMA5011I: The cleanup of the temp directory for appli
teWS is complete.
[03:25:02 PM] Installation Completed Sucessfully: QuoteWS
[03:25:06 PM] Starting Application: QuoteWS
[03:25:08 PM] Application Started Sucessfully: QuoteWS
```

6. Verify that the QuoteWS Application is up and running on the server.
- a. Open a Web browser and enter the administrative console's address.

<http://localhost:9060/ibm/console>

- ___ b. Verify that the **QuoteWS** application has been installed and it is running.

		Start	Stop	Install	Uninstall	Update	Rollout Update	Remove File	Export	Export DDL
Select	Name	Status								
<input type="checkbox"/>	DefaultApplication									
<input type="checkbox"/>	IBMMUTC									
<input checked="" type="checkbox"/>	QuoteWS									
<input type="checkbox"/>	TradeApplication									
<input type="checkbox"/>	ivtApp									
<input type="checkbox"/>	query									
Total 6										

- ___ 7. Stop the wrd monitor by entering **quit** in its command window.
- ___ 8. Test the **TradeApplication** application to verify that the QuoteWS module is working.
- ___ a. On a Web browser window enter the following address to access the application:

<http://localhost:9080/Trade/web>



Step 2 Test the Trade Application

In the next few steps you test the Trade Application to become familiar with its function.

- ___ 1. Log in to the application
 - ___ a. Click **Login** to gain access to the application.
 - ___ b. At the login page enter **client** as the User.

The screenshot shows the 'Please login' screen of the Trade Application. At the top, there is a blue header bar with the word 'TRADE:' in large white letters. In the top right corner, it says 'Powered by IBM WebSphere e-business software'. Below the header, the main content area has a light gray background. It features a large bold title 'Please login'. Underneath the title is a form field labeled 'User' containing the value 'client'. To the right of the user field is a blue rectangular button with the word 'Login' and a small blue information icon inside. Below the login button is another blue rectangular button labeled 'Welcome Page'. A horizontal green line separates this section from the one below. The second section contains a table with the caption 'Runtime server information'. The table has three columns: 'Cell', 'Node', and 'Process'. The data rows show 'was6host00Cell01' in the Cell column, 'was6host00Node01' in the Node column, and 'server1' in the Process column. Below the table is a blue rectangular button labeled 'Show cookies'. Further down, the text 'Version: 6.246.0' is displayed. At the very bottom of the page, there is a horizontal bar divided into four equal-width segments, each containing the text 'server1'.

2. After login you are placed on the **Account Information** page. From this page you can get a quote or initiate an update to your account information.

TRADE:		Powered by IBM WebSp e-business software
Welcome Login Account Information Holdings Transaction History		
<input type="button" value="Get Quotes"/>		
Account Information <p>Name: Jessie James AccountId: 87652289 Email: jjames@us.ibm.com Credit Card Number: 555444907 Balance: \$11,558.17</p>		Preferences Use Internet quotes: <input checked="" type="checkbox"/> Broadcast transaction data: <input type="checkbox"/>
Home Address 97 Milton Rd. Pittsburgh, PA 15222 USA		Business Address 1050 Chum Ave. Markham, ON L5G 1K9 Canada
<input type="button" value="Update Account Page"/>		

Information: Note in the Preferences that Internet quotes are enabled. In this condition requesting a quote will attempt to obtain a real time quote, if the quote Web service is not available then the QuoteWS Web service is used to provide the quote (if the symbol exists in the database).

3. If you are connected to the Internet try to get a quote for a stock that is also contained in the quote database. This time you will get a real time quote. Try some of the symbols below and record their current value.



Symbol	Value
IBM	
DELL	
MSFT	
AOL	

- ___ 4. To force database quotes, change the user preferences. Click **Update Account Page**.

The screenshot shows the TRADE: application interface. At the top right, it says "Powered by IBM WebSphere e-business software" and "Version: 6.246.0". On the left, there's a "Get Quotes" button and an information icon (a blue circle with an 'i'). The main area is divided into sections:

- Account Information:** Name: Jessie James, AccountId: 87652289, Email: jjames@us.ibm.com, Credit Account Number: 555444907, Balance: \$11,558.17.
- Preferences:** A red oval highlights two checkboxes: "Use Internet quotes:" and "Broadcast transaction data:". Both checkboxes are currently unchecked.
- Home Address:** Address Line 1: 97 Milton Rd., Address Line 2: (empty), City: Pittsburgh, State: PA, Postal Code: 15222, Country: USA.
- Business Address:** Address Line 1: 1050 Chum Ave., Address Line 2: (empty), City: Markham, State: ON, Postal Code: L5G 1K9, Country: Canada.

At the bottom left is a "Update Account" button.

From this page you update your personal information, decide whether you want to get internet quotes (internet connection required), or whether quotes come from a database.

You also choose whether you wish to broadcast your transaction information to the main transaction processor. At this time do not select this option since the required setup to send asynchronous messages has not yet been done.

You can also modify your balance from this page (not shown on screen capture, scroll down the page to see where this is done).

- ___ 5. Uncheck **Use Internet quotes**.
___ 6. Click **Update Account**.

- ___ 7. Get the same quotes as you did in the step above. The values are now coming from the database and are NOT current.



Symbol	Value
IBM	
DELL	
MSFT	
AOL	

- ___ 8. If you are connected to the Internet, consider enabling Internet quotes again.
___ 9. Click **Login** to sign off the application.

End of lab

Exercise review and wrap-up

The first part of the exercise you used WRD in **autoappinstall** style to install the TradeApplication.ear file. This file contained a JDBC provider, a data source and a JAAS alias definition. Having these resources defined in the enhanced EAR file makes installing the application extremely easy and without having to resort to the console to configure these resources.

In the second part of the exercise you tested and exercised the application to become familiar with its function.

Exercise 9. Federating a cell

What this exercise is about

This lab demonstrates the process of creating a WebSphere Cell through the generation of a Deployment Manager profile followed 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
- Federate an application server profile into a cell
- Use backupconfig to create backups of a profile configurations
- Create a Custom profile and federate it

Introduction

This lab goes through the process of creating and federating a cell. The initial steps include creating 2 additional profiles, the first of which will be a Deployment Manager profile. Once the Deployment Manager profile is created, profile1 will be federated into the cell. Then a Custom profile will be created and federated at the same time.

This lab not only demonstrates the process of creating a cell, but it also prepares the lab environment for other important steps including creating a node to manager a remote Web server and clustering an application server.

Required materials

To perform this exercise, you must have a working WebSphere Application Server **server1**, administrative console and a running Default Application and Trade Application installed on **profile1**.

Instructor exercise overview

The first part of this exercise creates a deployment manager profile called DmgrProfile. After backing up both profiles' configuration (profile1 and DmgrProfile), the original profile1 from the earlier labs will be federated. The student then uses the Profile Creation Wizard to create an additional profile called profile2. This is created as a custom profile so that it can

be automatically federated during creation and so that there will be no server1 created for it.

This cell and these profiles are then required for future labs. Profile1 and profile2 will be used to do horizontal scaling, and a new node (ihsnode) will be added to manage the IHS server.

NOTE: Horizontal scaling implies that cluster members are spread across different physical boxes. Clearly these labs are limited to only one machine (although instructors are encouraged to see if their students might attempt to connect multiple machines into their cells). But, the configurations used for these labs, with having multiple profiles (even though they are on a single physical box), is logically the same as would be done in a true Horizontal scaling situation. But using this configuration, the stand alone student can complete all the labs and no network is required. Again, if students wish to map the federation, IHS and clustering labs to separate machines, and the instructor is comfortable with that approach, the students should be encouraged.

NOTE: It is important since all these profiles will end up being on the same machine that the machine name is consistent. During the profile creation, the host name is used. It is important that the host name is the same during the creation of all the profiles. WebSphere tries to make sure that when new profiles are created that ports are not reused. But if WebSphere thinks two profiles are on different hosts, it does not need to make sure the ports are unique. The problem is that when the logic for the port conflicts is executed, the host names are compared as simple strings. This means if one profile was created using was6host00 and another with was6host00.ibm.com, WebSphere does not take into account that they are on the same hosts.

Exercise instructions

Step 1 Create a deployment manager profile using the Profile creation wizard

The profile creation wizard is a GUI tool for creating WebSphere profiles. Using the profile wizard you can create an application server profile, deployment manager profile or custom profile. First create a deployment manager profile.

- ___ 1. Start the profile wizard. This is the same wizard you used earlier to create profile1.
 - ___ a. Click **Start** —> **Programs** —> **IBM WebSphere** —> **Application Server Network Deployment v6** —> **Profile creation wizard**

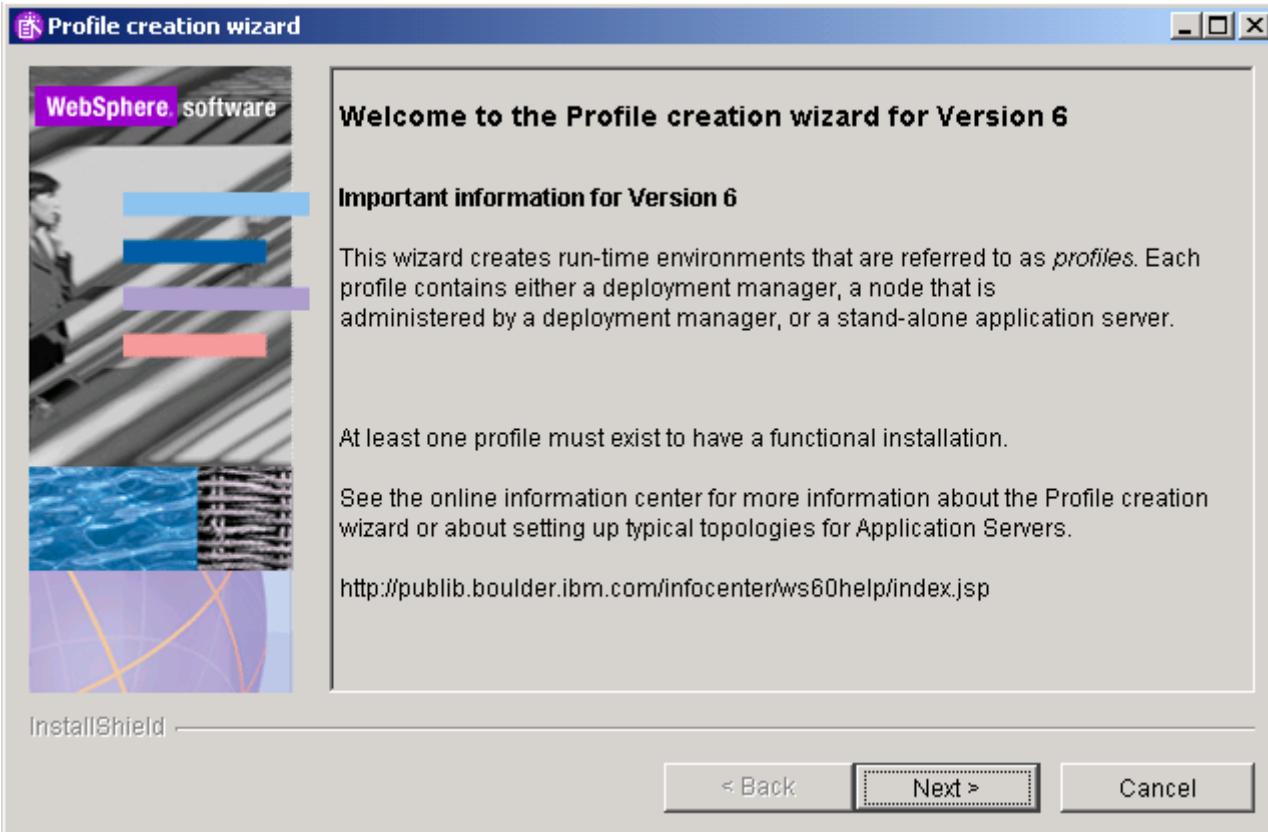
Information: It is also possible to create profiles from the command line using the wasprofile -create script located in the <was_root>\bin directory.

For example:

```
wasprofile -create -profileName profile2  
-profilePath "<profile_root>\profile2"  
-templatePath "<was_root>\profileTemplates\default"  
-nodeName was6hostXXNode02 -cellName was6hostXXCell02  
-hostName was6hostXX
```

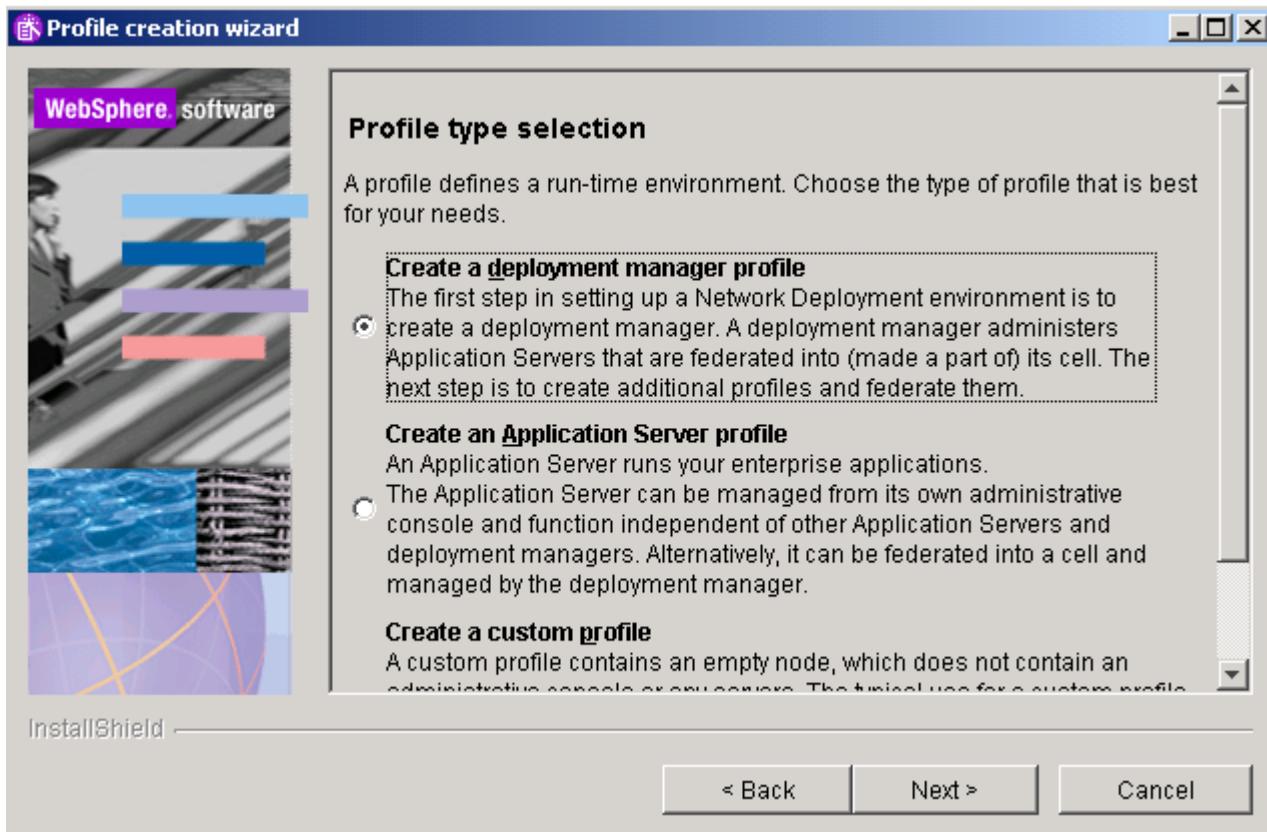
UNIX: To start the Profile creation wizard on UNIX, run the appropriate binary in <was_root>/bin/ProfileCreator. For example, pctLinux.bin.

- __ b. The Welcome screen for the Profile creation wizard appears.

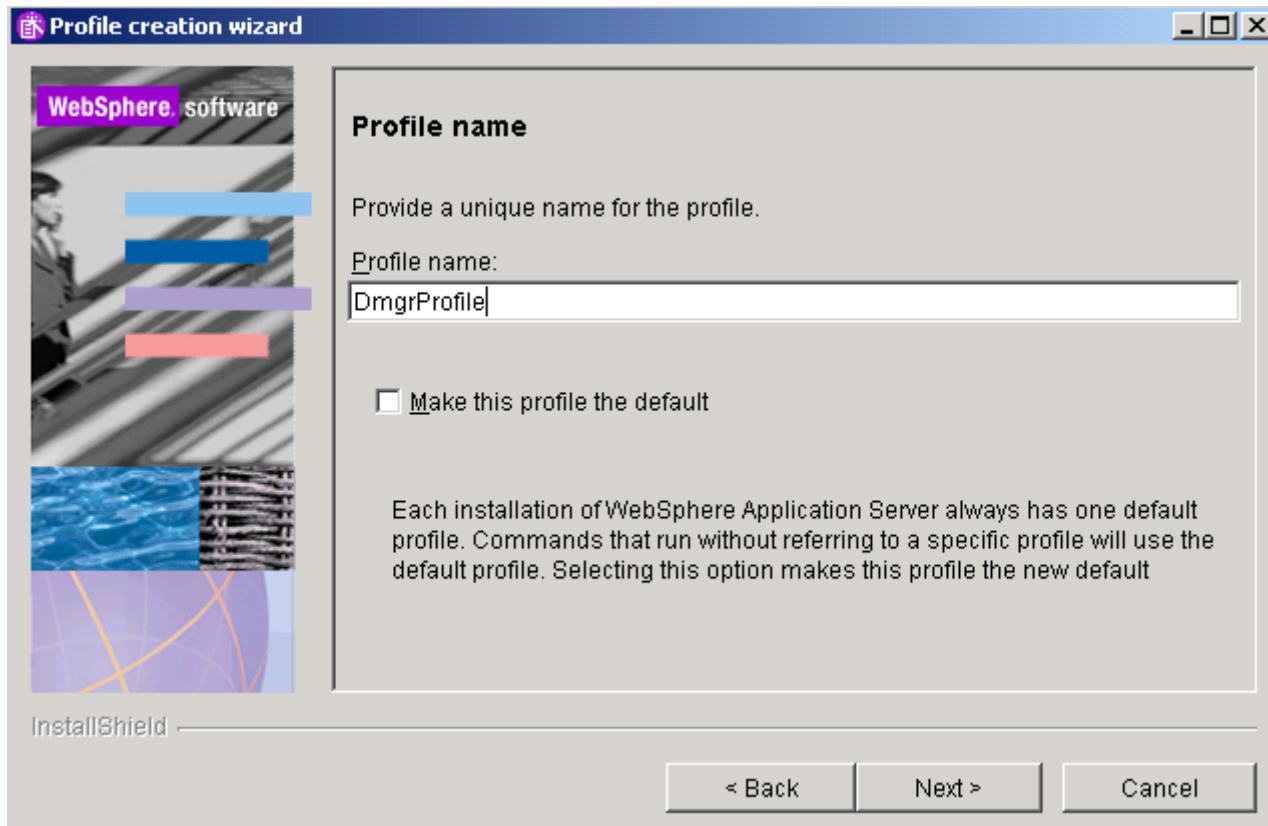


- __ c. Click **Next**.
__ 2. Create a deployment manager profile called **DmgrProfile**.

- __ a. Select the **Create a deployment manager profile** option and click **Next**.



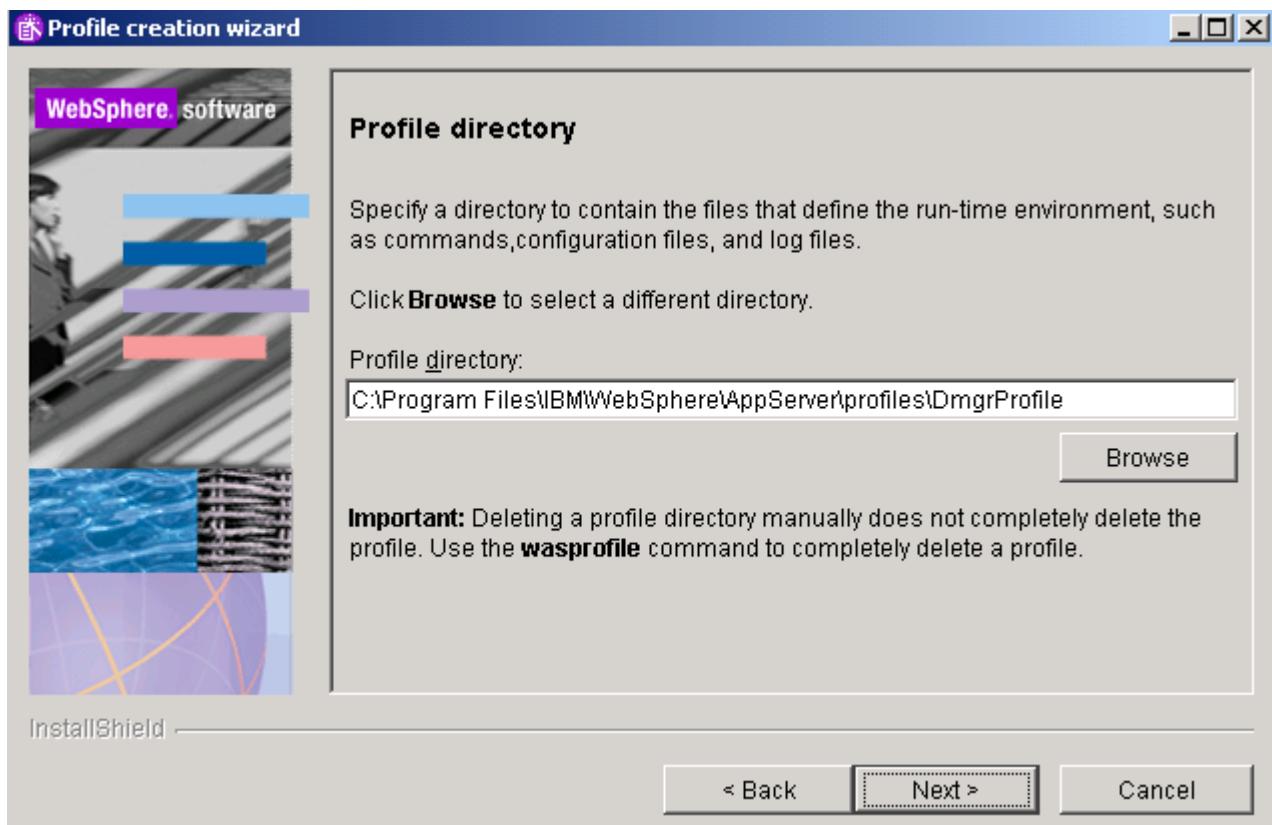
- __ b. Specify **DmgrProfile** for the deployment manager profile name. Do not select **Make this profile the default** option. Click **Next**.



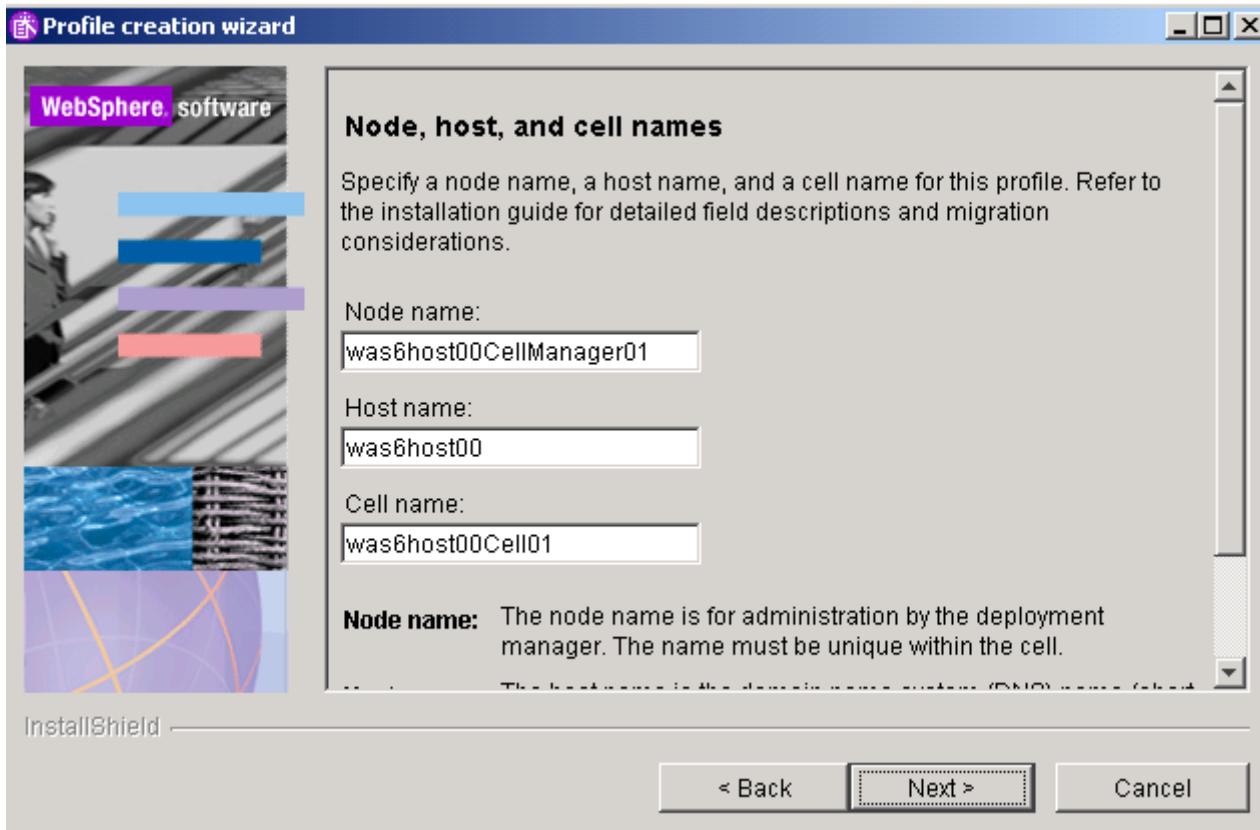
Information: The default profile will initially be the first profile created. It is also possible to change which profile is designated as the default with the Profile creation wizard or the wasprofile command.

When running commands from the <was_root>\bin directory, commands will be executed against the runtime defined by the default profile. It is also possible to specify a particular profile using the -profileName argument.

- ___ c. Next specify a directory for the files for the deployment manager profile. Use the default directory <profile_root>\DmgrProfile. Click **Next**.



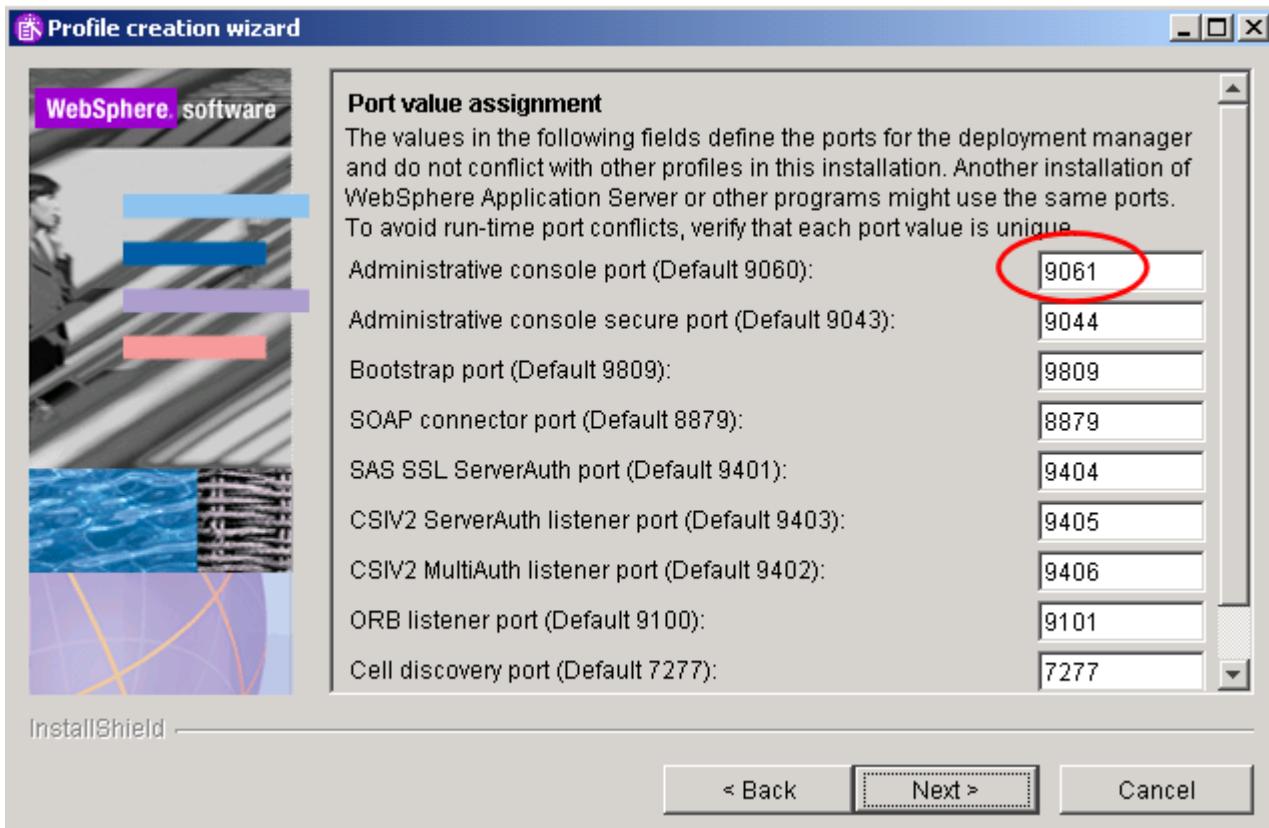
- ___ d. The next window sets the node, host and cell names. Default values are filled in based on the detected TCP/IP host name for your machine. **Ensure the correct names are entered.** Check with your instructor if you are unsure.



Information: Make sure that only the host name of the computer is used, NOT the fully qualified host name, which includes the domain suffix. Use was6hostXX, not was6hostXX.ibm.com.

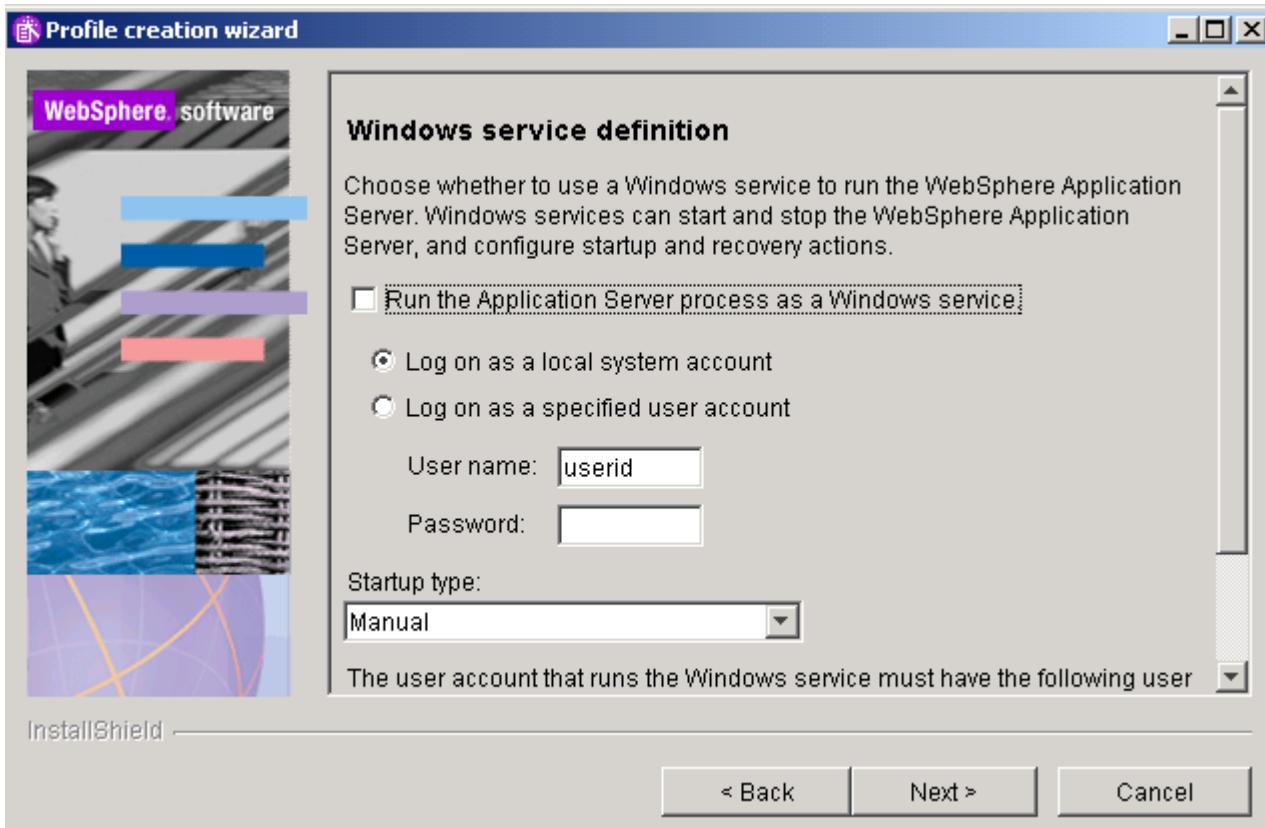
- ___ e. Click **Next**.

- ___ f. The next window allows you to set any ports for the deployment manager so there are no conflicts with other profiles. Keep all the defaults and click **Next**.

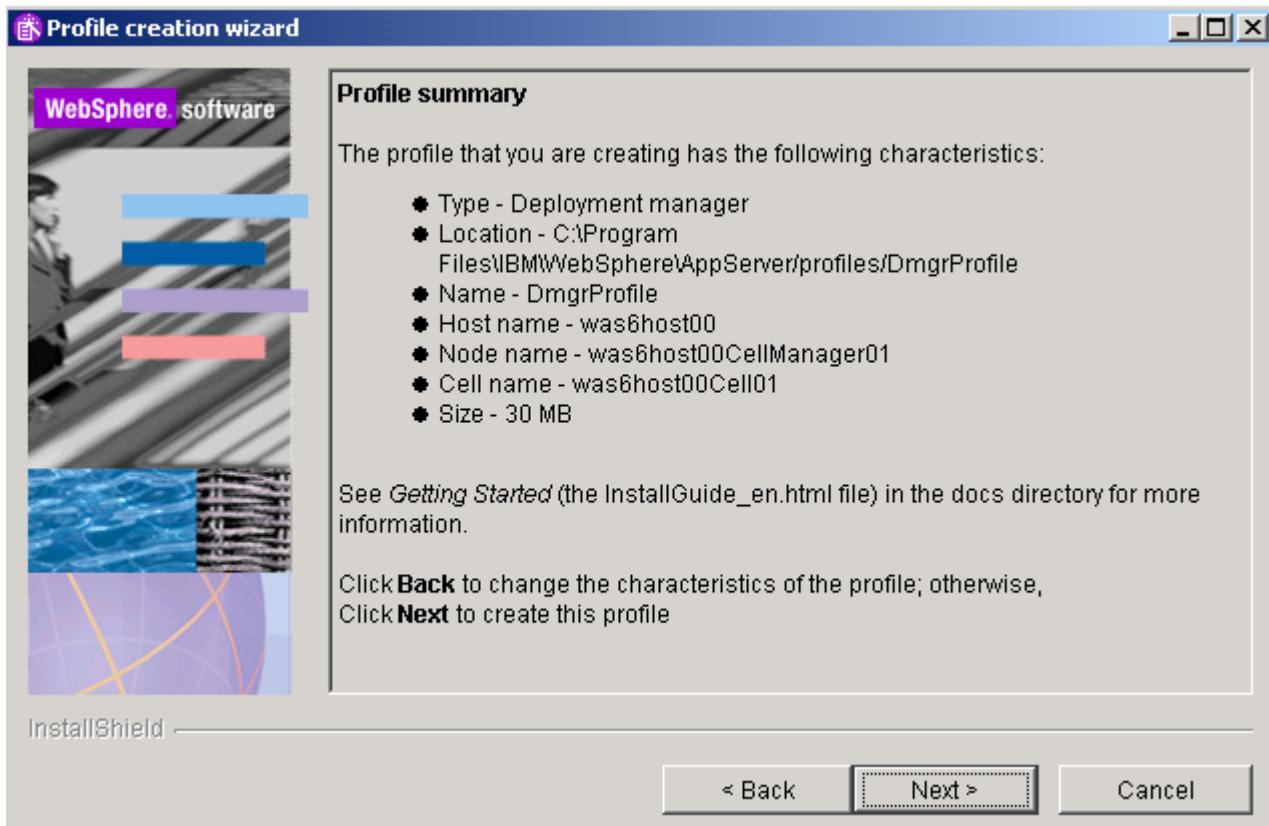


Information: Notice the administrative console port for the deployment manager. This will be used later in this exercise.

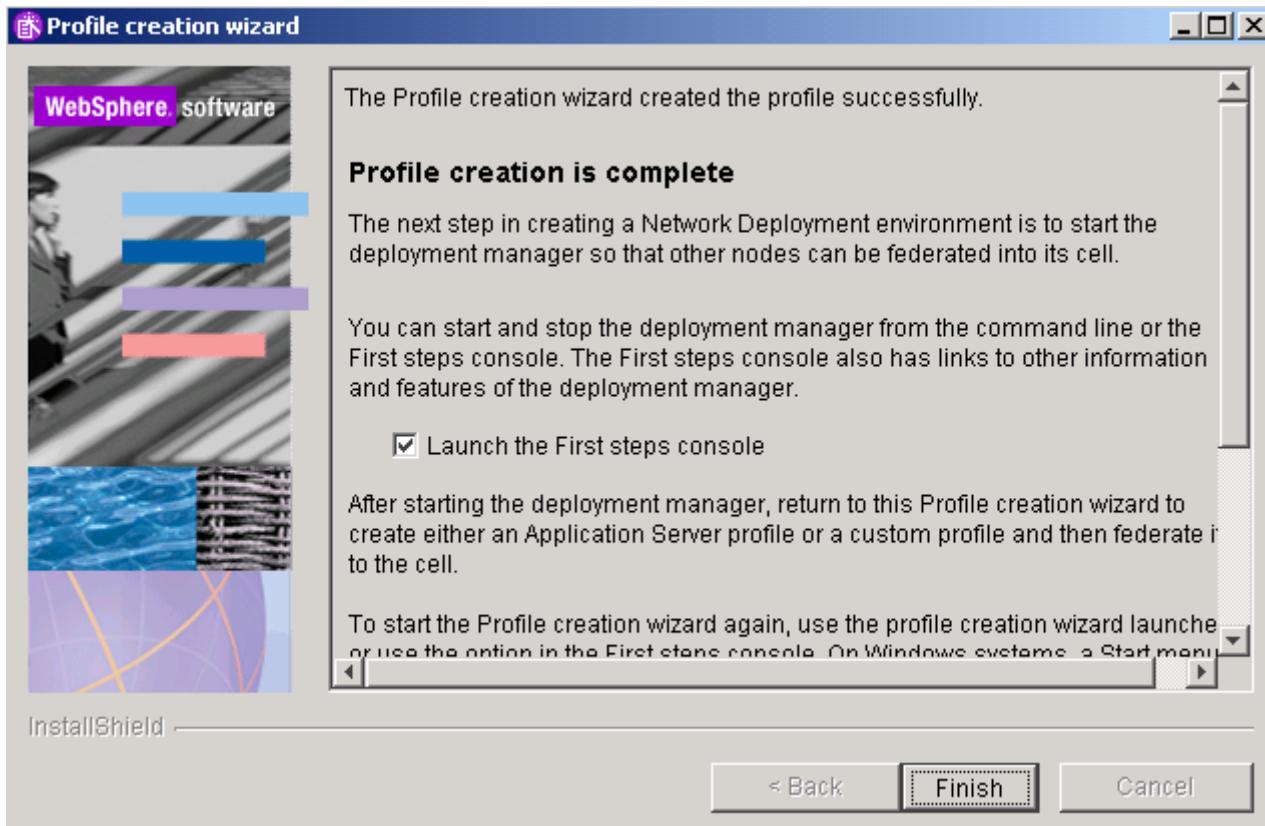
- g. For this lab, do not run as a Windows service. Unselect the box for **Run the Application Server process as a Windows service**. Click **Next**.



- ___ h. The profile summary appears with all of the choices you have made on previous screens. Double-check your information with what is shown below. Click **Next**.



- ___ i. The profile creation completes and profile **DmgrProfile** is created. Notice that the **Launch the First steps console** box is selected. Click **Finish** and the First Steps program will launch.



3. This First steps console is associated with the deployment manager profile, DmgrProfile, that was just created. Each profile has its own First steps console. Select **Installation verification** from the console.

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.

Profile creation wizard
Create a profile.

Information center for WebSphere Application Server
Learn more about WebSphere Application Server.

Migration wizard
Migrate WebSphere Application Server V4 or V5 to V6.0.

Exit

- ___ 4. The installation verification test tool runs and display messages to indicate its status. You should see the message "IVTL00801: Installation Verification is complete". Use the scroll bar to scroll to the bottom to see all the messages.

```

First steps output - Installation verification
Node name is: was6host00\was6host00
Current encoding is: Cp1252
Server port number is: 9061
IVTL0020I: Could not connect to Application Server, waiting for server to start
IVTL0010I: Connecting to the WebSphere Application Server was6host00 on port: 9061
IVTL0020I: Could not connect to Application Server, waiting for server to start
Start running the following command:cmd.exe /c "C:\Program Files\IBM\WebSphere\AppServer\profiles\dmgr\bin>ADMU0116I: Tool information is being logged in file C:\Program
>      Files\IBM\WebSphere\AppServer\profiles\dmgr\logs\dmgr\startServer.log
>ADMU0128I: Starting tool with the DmgrProfile profile
>ADMU3100I: Reading configuration for server: dmgr
>ADMU3200I: Server launched. Waiting for initialization status.
>ADMU3000I: Server dmgr open for e-business; process id is 1572
IVTL0015I: WebSphere Application Server was6host00 is running on port 9061 for profile DmgrProfile
IVTL0035I: Scanning the file C:\Program Files\IBM\WebSphere\AppServer\profiles\dmgr\logs\dmgr\startServer.log
IVTL0040I: 0 errors/warnings were detected in the file C:\Program Files\IBM\WebSphere\AppServer\profiles\dmgr\logs\dmgr\startServer.log
IVTL0070I: IVT Verification Succeeded
IVTL0080I: Installation Verification is complete

```

- ___ 5. Close the First steps installation verification output window.
___ 6. Exit the First steps console.

Step 2 Backup the DmgrProfile configuration

Before continuing, it would probably be a good idea to backup the configuration for the DmgrProfile that was just created.

- ___ 1. Using the backup command line tool, create a backup for the DmgrProfile.
- ___ a. In a command prompt window, change to the <profile_root>\DmgrProfile\bin folder.
 - ___ b. Create the backup by entering the following command:

```
backupConfig
```

```
C:\Program Files\IBM\WebSphere\AppServer\profiles\dmgrProfile\bin>backupConfig
ADMU0116I: Tool information is being logged in file C:\Program
Files\IBM\WebSphere\AppServer\profiles\dmgrProfile\logs\backupConfig.log
ADMU0128I: Starting tool with the dmgrProfile profile
ADMU5001I: Backing up config directory C:\Program
Files\IBM\WebSphere\AppServer\profiles\dmgrProfile\config to file
C:\Program
Files\IBM\WebSphere\AppServer\profiles\dmgrProfile\bin\WebSphereConfig_2005-02-23.zip
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: dmgr
ADMU2010I: Stopping all server processes for node was6host00CellManager01
ADMU0510I: Server dmgr is now STOPPED
ADMU5002I: 99 files successfully backed up
```

UNIX: The command would be ./backupConfig.sh

- ___ 2. backupConfig will create a backup file called WebSphereConfig_<date>.zip using the current date and place it in the <profile_root>\DmgrProfile\bin directory.
 - ___ a. For safe keeping, copy the **WebSphereConfig_<date>.zip** file from the <profile_root>\DmgrProfile\bin\ directory in <software_dir>\Backups and rename it to backup_DmgrProfile_prefederation.zip.
- ___ 3. Verify that the deployment manager is stopped by using the **serverStatus -all** command from the <profile_root>\DmgrProfile\bin\ directory.
- ___ 4. Since profile1 will shortly be federated, create a backup for it as well. Make sure to perform the backupConfig command from the <profile_root>\profile1\bin directory. Copy the backup file to <software_dir>\Backups and rename it to backup_Profile1_prefederation.zip
- ___ 5. Start the deployment manager.
 - ___ a. In a command prompt window navigate to <profile_root>\DmgrProfile\bin and run the **startManager** command to start the deployment manager.

UNIX: The start command on a UNIX machine would be ./startManager.sh

Information: You can also start the deployment manager using the Start menu by clicking **Start —> Programs —> IBM WebSphere —> Application Server Network Deployment v6 —> Profiles —> DmgrProfile —> Start the deployment manager.**

Step 3 Federate Profile1 into the Deployment Manager's cell

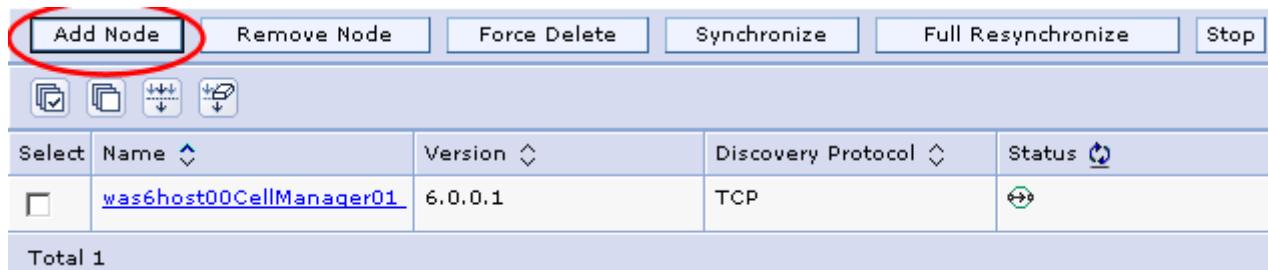
This step Federates profile1 into the cell that is defined by the Deployment Manager profile that was created earlier in this lab.

- ___ 1. Verify that **profile1's server1** is running.
 - ___ a. Navigate to the <profile_root>\profile1\bin folder and execute the **serverStatus server1** command.
 - ___ b. If server1 is not running execute the **startServer server1** command to start the application server.
- ___ 2. Verify that the Deployment Manager is running. This should have been started in the previous section of this lab.
- ___ 3. Open a console to the Deployment Manager.
 - ___ a. Open a Web browser and specify the following address:

`http://localhost:9061/ibm/console`

Information: You can also start the deployment manager's console using the Start menu by clicking **Start** → **Programs** → **IBM WebSphere** → **Application Server Network Deployment v6** → **Profiles** → **DmgrProfile** → **Administrative console**

- ___ b. Log in to the Deployment Manager's console as userid **admin**.
- ___ 4. Federate a node into the cell.
 - ___ a. From the administrative console select **System administration** → **Nodes**.
 - ___ b. Click **Add Node**



Select	Name	Version	Discovery Protocol	Status
<input type="checkbox"/>	was6host00CellManager01	6.0.0.1	TCP	
Total 1				

- ___ c. Select the option to create a **Managed node**. A managed node contains a WebSphere application server and a node agent. The application server runs as part of the network deployment environment. Click **Next**.

Add Node

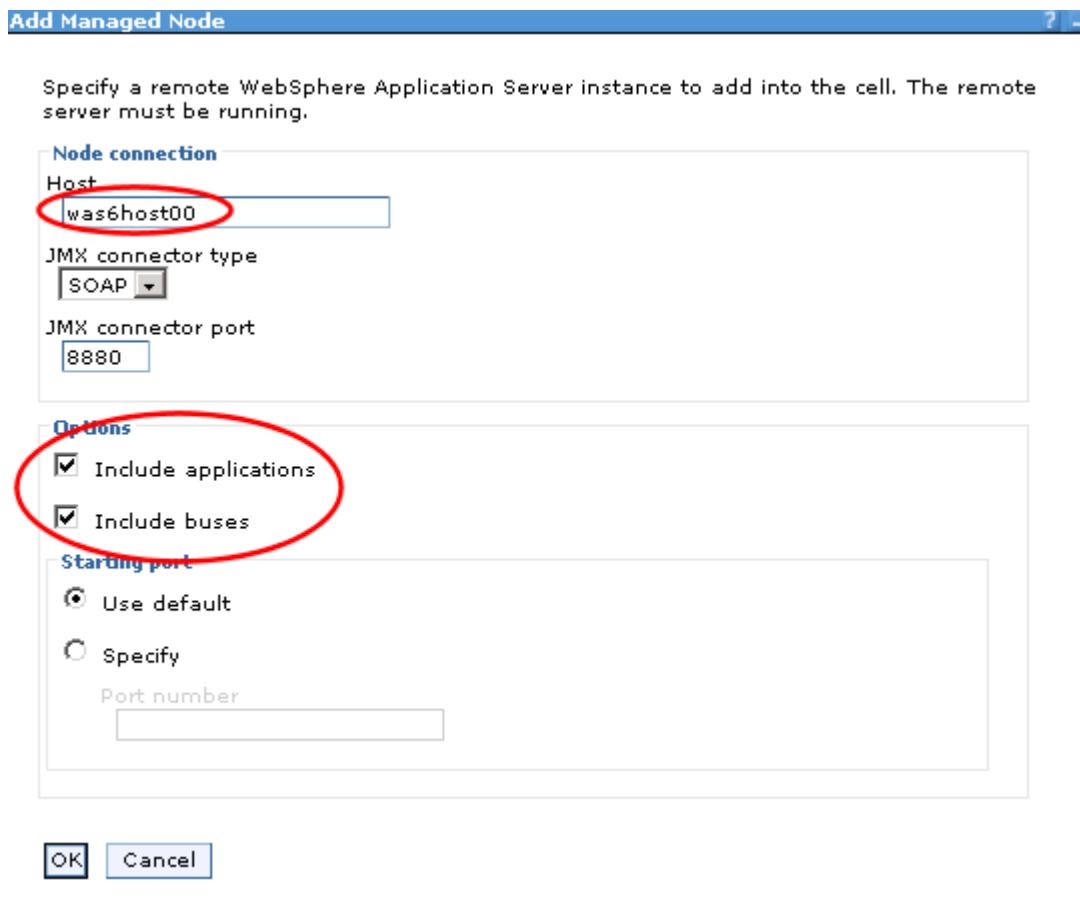
Specify to create a managed or an unmanaged node.

Managed node
A managed node contains a WebSphere application server that runs as part the network deployment environment. It has a node agent process which maintains the node's configuration and operation. Choosing this option will result in running the add node utility.

Unmanaged node
An unmanaged node represents a node in the topology that does not have a node agent process for management when running in the network deployment environment. Unmanaged nodes are used for defining web servers in the topology.

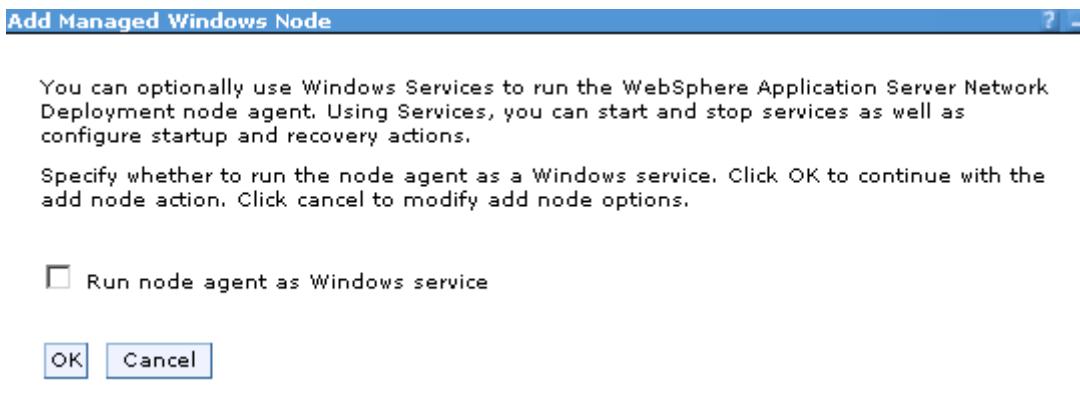
Next **Cancel**

- ___ d. Specify your host name **was6hostXX** for the host and select the options for **Include applications** and **Include buses**. Keep all remaining defaults. Click **OK**.



Information: The port number above points to the node you want add to the cell.

- ___ e. On the next window do not select the node agent to run as a Windows service. Click **OK**.



- ___ f. Verify that the node was added to the cell configuration. You should see the message “Congratulations! Your node was6hostXXNode02 has been successfully incorporated into the cell.”

ADMU0300I: Congratulations! Your node was6host00Node01 has been successfully incorporated into the was6host00Cell01 cell.

ADMU9990I:

ADMU0306I: Be aware:

ADMU0302I: Any cell-level documents from the standalone was6host00Node01Cell configuration have not been migrated to the new

ADMU0307I: You might want to:

ADMU0303I: Update the configuration on the was6host00Cell01 Deployment Manager with values from the old cell-level documents.

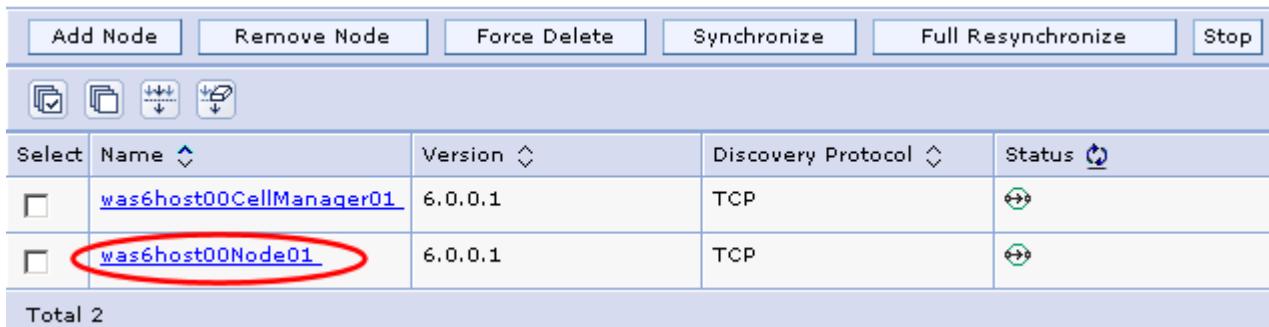
ADMU9990I:

ADMU0003I: Node was6host00Node01 has been successfully federated.

The new node will not be available in the console until you log in again

[Logout from the WebSphere Administrative Console](#)

- ___ g. **Logout** of the administrative console. The new node will not be available in the administrative console until you **log in** again.
- ___ 5. Verify the cell configuration.
- ___ a. Log in to the administrative console using the userid **admin**.
- ___ b. Select **System administration —> Nodes**. Two nodes should be listed. The deployment manager and the was6XXNode01 that was just added.



Add Node	Remove Node	Force Delete	Synchronize	Full Resynchronize	Stop
Select	Name	Version	Discovery Protocol	Status	
<input type="checkbox"/>	was6host00CellManager01	6.0.0.1	TCP		
<input type="checkbox"/>	was6host00Node01	6.0.0.1	TCP		
Total 2					

- ___ c. Verify that the node agent on was6hostXXNode01 has started. Select **System administration** —> **Node agents**. The status of the node agent is Started.

Select	Name	Node	Version	Status
<input type="checkbox"/>	nodeagent	was6host00Node01	6.0.0.1	
Total 1				

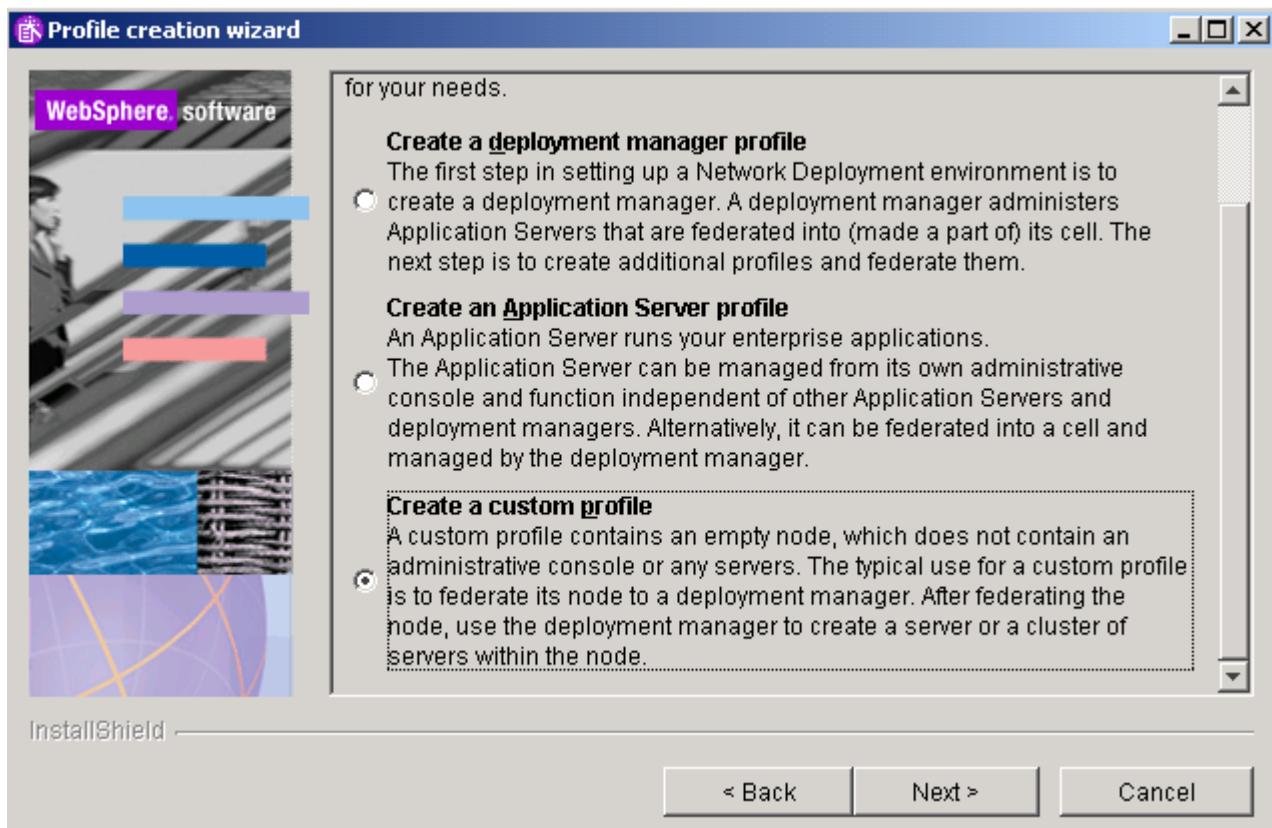
- ___ 6. Start the application server and test the snoop servlet.
- ___ a. Select **Servers** —> **Application servers**. Select the box in front of **server1** and click **Start**. The status for **server1** should now be started.
 - ___ b. Verify the DefaultApplication is running. Select **Application** —> **Enterprise Applications**. DefaultApplication should be running.
 - ___ c. Open another browser window and type in the following address:
`http://localhost:9080/snoop`
- ___ 7. Close the browser window.

Step 4 Create a custom profile and federate into the Deployment Manager's cell

This step creates profile2 as a custom profile and automatically federates it into the cell. A custom profile is useful because it doesn't create any servers on the node, it only creates the configuration and the node agent. This means that no server1 will be created on that node.

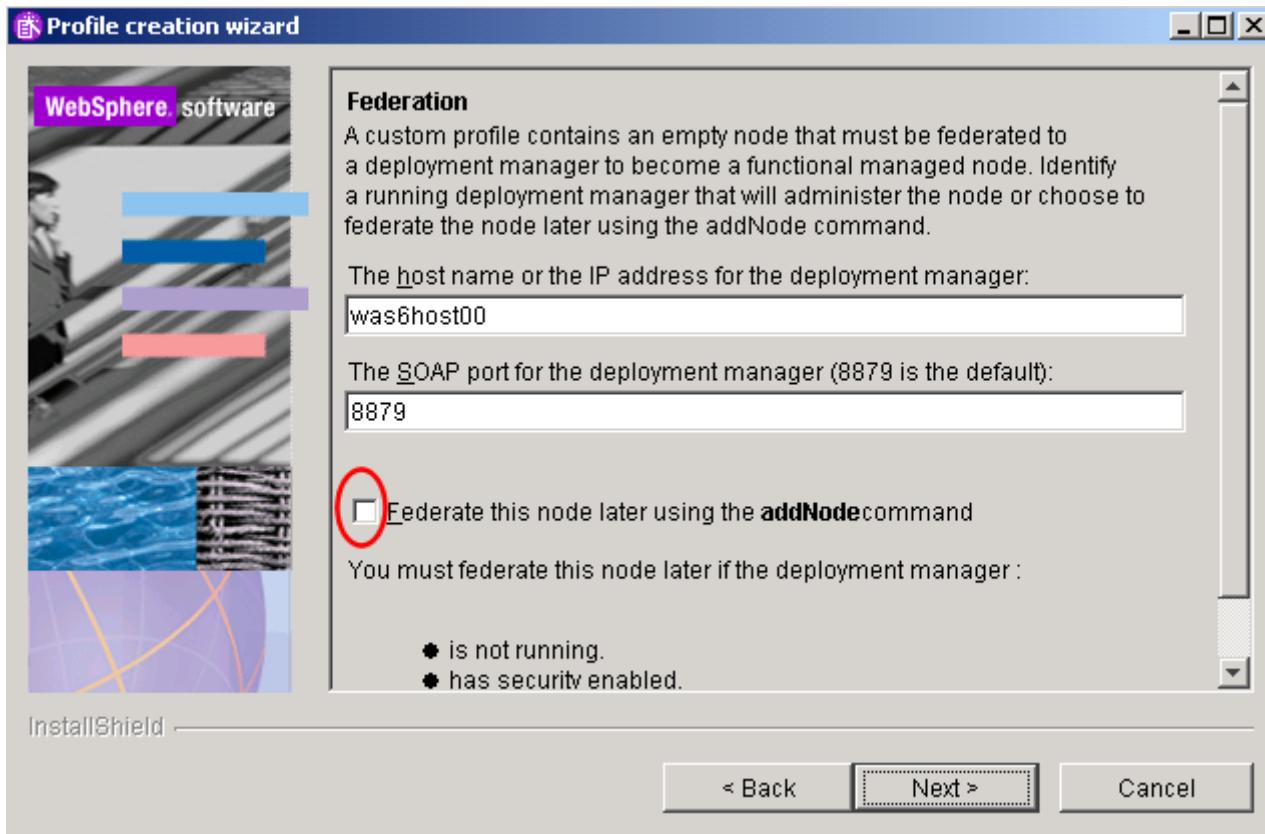
- ___ 1. Start the profile creation wizard.
 - ___ a. Click **Start** —> **Programs** —> **IBM WebSphere** —> **Application Server Network Deployment v6** —> **Profile creation wizard**.
- UNIX:** To start the Profile creation wizard on UNIX, run the appropriate binary in <was_root>/bin/ProfileCreator. For example, pctLinux.bin.
- ___ b. The Welcome screen for the Profile creation wizard appears. Click **Next**.
 - ___ 2. Create a custom profile called **profile2** and federate to the deployment manager configuration.

- ___ a. On the Profile creation wizard window select the **Create a custom profile** option and click **Next**.



- ___ b. On the Federation window enter **was6hostXX** for the host name and keep all other defaults. Be sure that the **Federate this node later using the addNode**

command option is not selected. The node should automatically be federated to the configuration. Click **Next**.



Information: Not selecting to federate the node until **later**, using the **addNode** command manually, causes the node to be federated **now** as part of the process defined by the wizard.

- ___ c. Specify **profile2** for the unique name for the profile. Do not select to make this profile the default. Click **Next**.
 - ___ d. Keep the defaults for the profile directory on the Profile directory window. The directory should be **<profile_root>\profile2**. Click **Next**.
 - ___ e. Default values will appear on the Node and host names windows. Verify a node name of **was6hostXXNode02** and a host name **was6hostXX**. Click **Next**.
 - ___ f. The next window allows you to set any ports to be used so there are no conflicts with other profiles. Keep all the defaults and click **Next**.
 - ___ g. The profile summary will appear with all of the choices made on previous screens. Double-check your choices against what is shown. Click **Next**.
 - ___ h. The profile creation is now complete. On the Profile creation wizard window unselect the **Launch the First steps console** option. Do not use the First steps console for profile2. Click **Finish** to exit the wizard.
- ___ 3. The node **was6hostXXNode02** is automatically federated into the deployment manager configuration. Verify these new configuration changes.

- ___ a. Using the deployment managers administrative console, list the nodes. Select **System administration —> Nodes**. You should see that the node **was6hostXXNode02** had been federated.

Information: Using a Custom profile does not create a server1 instance. This is useful when adding nodes to an existing cell since 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 the both nodeagents have been started.
 - ___ a. If any of the nodeagents need to be started, use the command **<profile_root>\profileX\bin\startNode.bat** to start them.
- UNIX:** The command to start the nodeagent would be **<profile_root>/profileX/bin/startNode.sh**
- ___ 5. Define the **DB2_JDBC_DRIVER_PATH** WebSphere environment variable for node **was6hostXXNode02**.

Information: The application servers on the new node will require database access. This access is done through the data source defined within the ear and uses the WebSphere environment variable to define the specific path to the DB2 driver zip file. It is therefore necessary to define that variable on each node. Alternatively, the variable could be defined at the cluster scope.

- ___ a. From the Administrative console, select **Environment —> WebSphere Variables**.
 - ___ b. Click **Browse nodes**.
 - ___ c. Select **was6hostXXNode02**.
 - ___ d. Select **DB2_JDBC_DRIVER_PATH** and enter the appropriate path in the value field. On Windows the path would be **C:\Program Files\IBM\SQLLIB\java**.
- UNIX:** The value for the driver on UNIX systems will be **<db2>/java**, where <db2> is the DB2 installation directory specified in the appendix for your platform.
- ___ e. Click **OK** and **save** the changes.

End of lab

Exercise review and wrap-up

This exercise goes through the process of creating a cell by producing a Deployment Manager profile. It then federates two additional profiles (or nodes) into the cell. The first profile which was federated was profile1 which was created during the WebSphere installation lab. The second profile was created as a Custom profile.

Exercise 10. IBM Http Server installation

What this exercise is about

This lab covers the installation of IBM Http Server. The IBM Http Server V6 and its plug-ins are installed and enabled to work with WebSphere Application Server V6. The DefaultApplication is mapped to the Web server and the plug-in configuration file is regenerated.

After installation, test the product to ensure the IBM Http Server was installed successfully and the Web server is working with the application server.

What you should be able to do

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

- Describe the installation process for IBM Http Server
- Verify the installation of IBM Http Server
- Describe the directories and configuration files for WebSphere Application Server with profiles
- Describe how to use WebSphere Application Server command utilities

Introduction

In this exercise you install the IBM Http Server.

A Windows administrator user ID has been created for you. Use this ID to log in to Windows and to configure services and database access.

- Userid: **userid**
- Password: **was1edu**

Information: Microsoft Windows 2000 passwords are case sensitive.

UNIX: In the UNIX environments, you will be using:

- Userid: **root**
- Password: **was1edu**

Required materials

To complete this exercise, you need WebSphere Application Server Network Deployment installed on the machine.

Instructor exercise overview

In this exercise the students learn how to install IBM Http Server and the WebSphere plug-ins. Students also map modules to the Web server and regenerate and propagate the plug-in configuration file.

For the Fedora image, it will not meet the system prerequisites. Instead, the user is prompted with two options: “Click Cancel to stop the installation and to install a supported operating system” or “Click Next to continue the installation.” Students should click Next and continue with the installation. Installation will complete successfully.

At the end of this exercise IBM Http Server and the WebSphere plug-ins are installed. The DefaultApplication is mapped to the Web server.

Exercise instructions

Install IBM Http Server

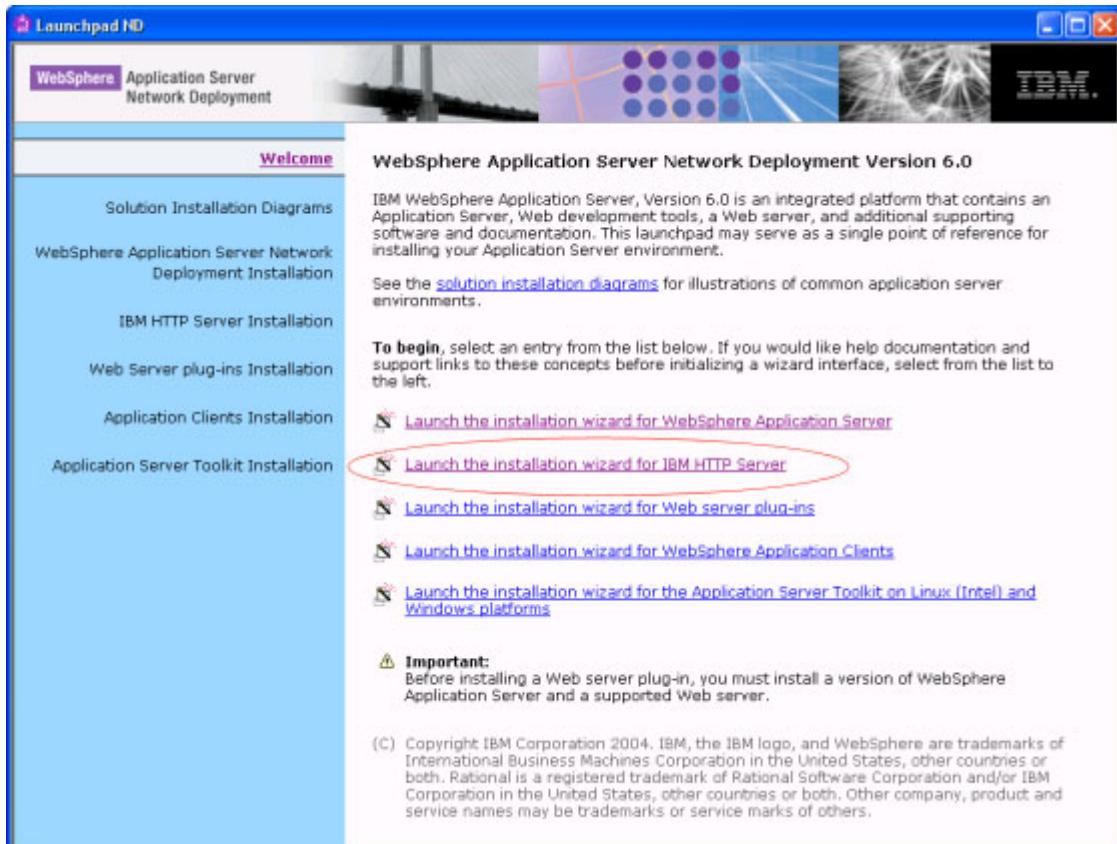
Install the IBM Http Server and its plug-ins.

- ___ 1. Invoke the WebSphere Application Server Launchpad. You may already have the Launchpad running from earlier in the labs.
 - ___ a. Navigate to <software_cds>\WAS6\.
 - ___ b. Run (Double-click) **launchpad**.

UNIX:

- ___ a. Open a terminal window and navigate to <software_cds>/WAS6/.
- ___ b. Enter **./launchpad.sh**

- ___ 2. On the Welcome panel click **Launch the installation wizard for IBM HTTP Server** to install IBM Http Server.

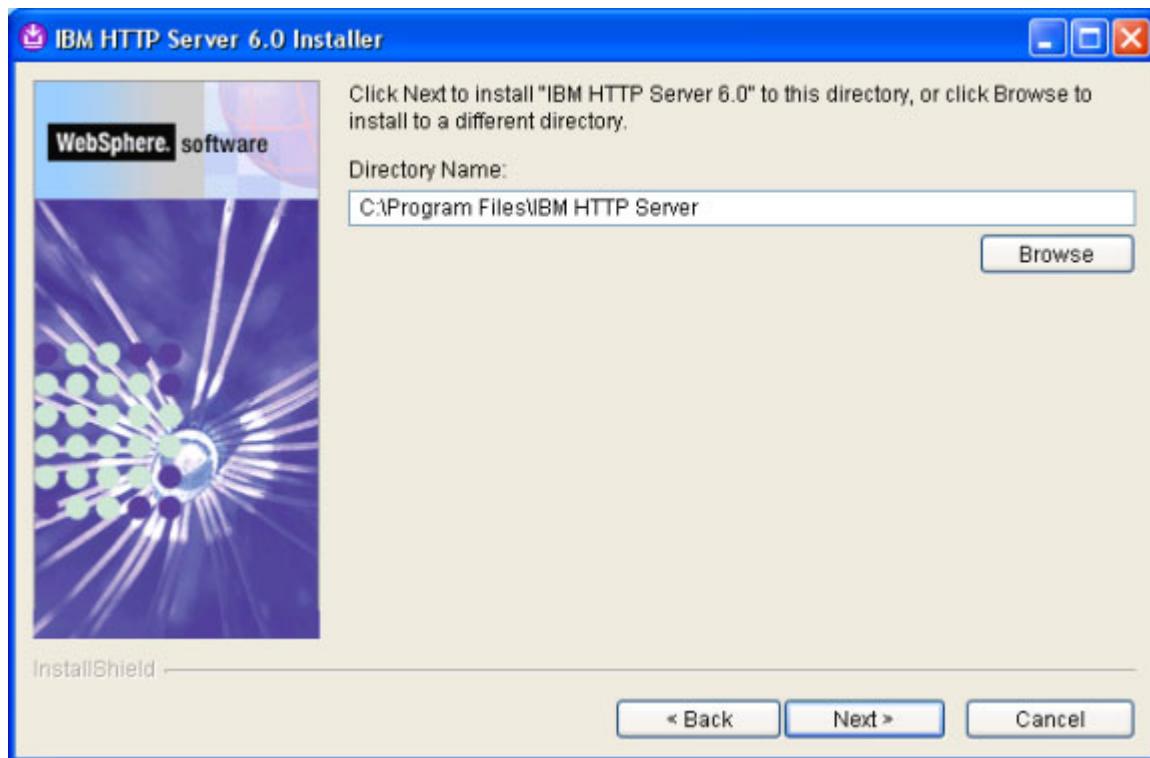


- __ a. The Welcome screen for the IBM Http Server installation appears. Click **Next**.



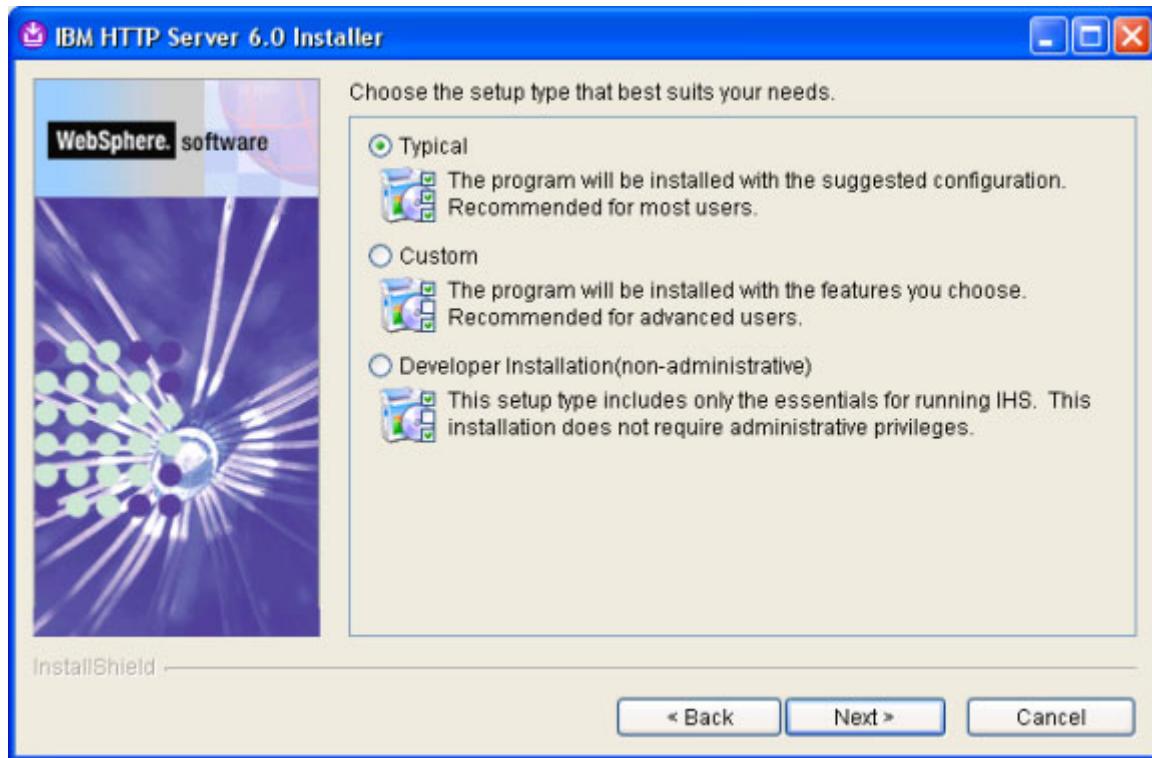
- __ b. Accept the license agreement. Click **Next**.

- c. The next window lets you change the installation directory. Keep the default for the installation directory. Click **Next**.



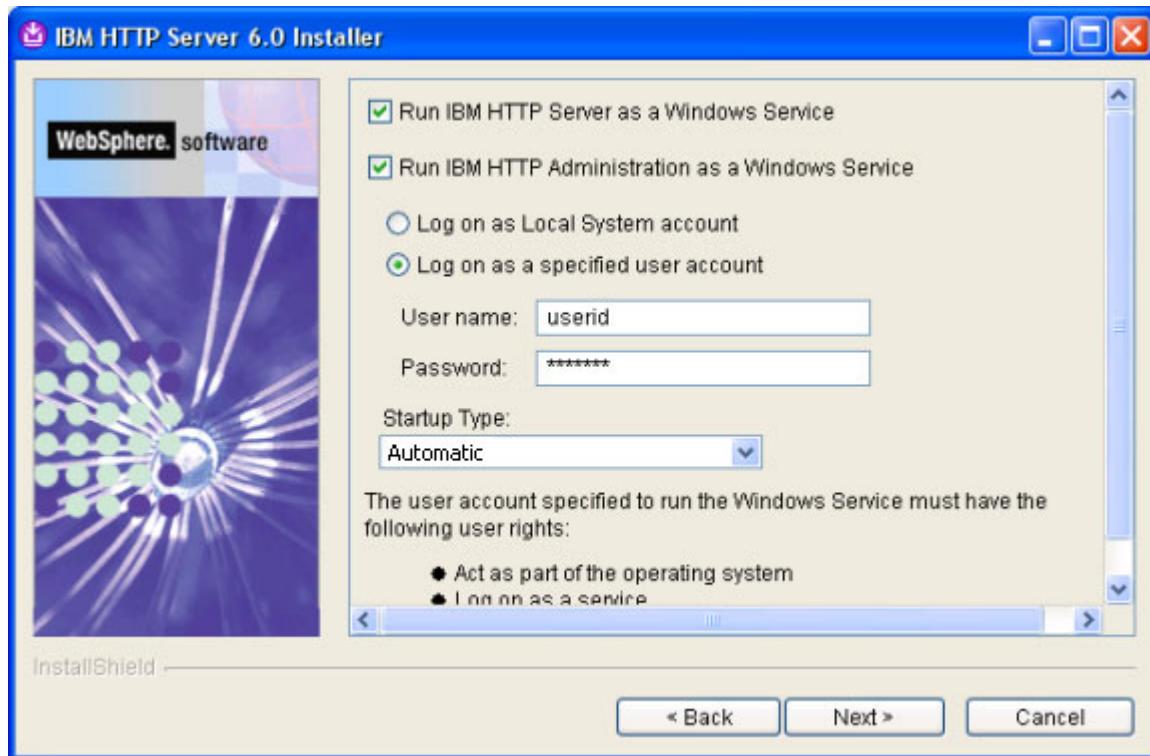
UNIX: In UNIX based systems, the directory structures will be different. Refer to the appropriate [\[Appendix\]](#) for the appropriate directory structures. The install wizard should give the appropriate default values for these directories.

- __ d. Select a **Typical** installation and click **Next**.



- __ e. Select if you want IBM Http Server to be run as a Windows service. If you do, you must specify the user it runs as. Ensure both check boxes are selected and type in the User name **userid** and a password of **was1edu**.

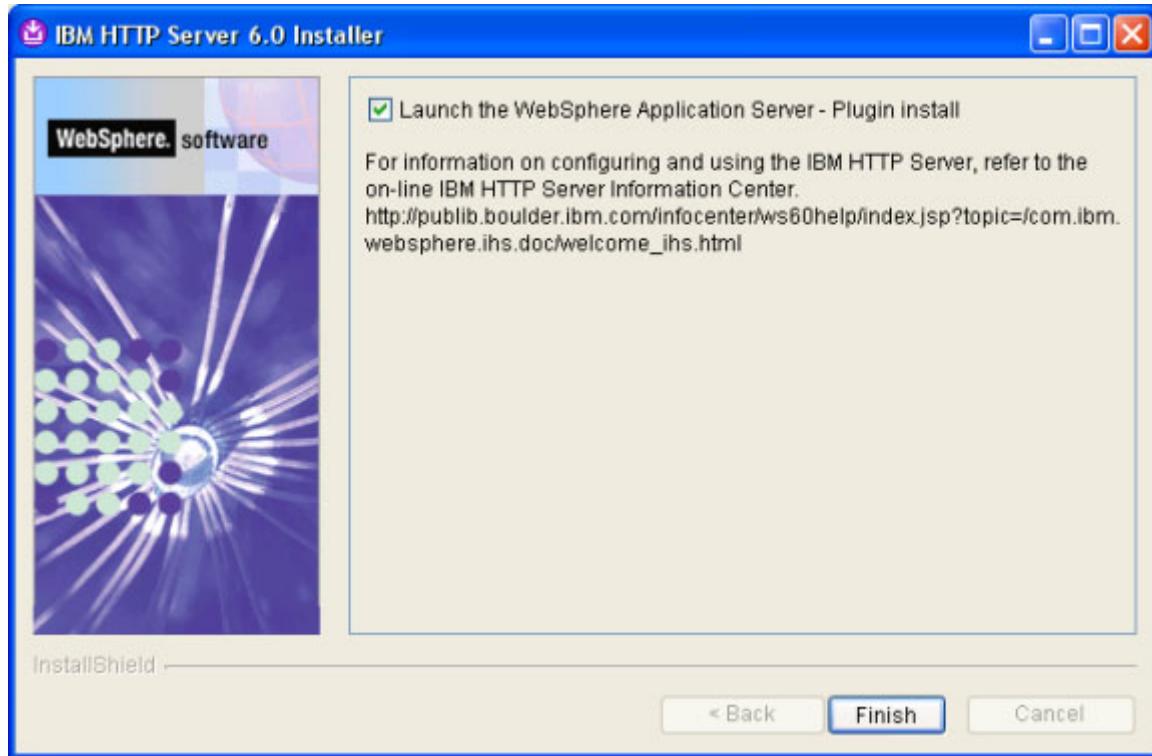
keep the Startup Type of **Automatic**. Click **Next**.



UNIX: On UNIX based platforms, you will not have this option. This screen will not be available.

- ___ f. On the summary window click **Next** to complete the installation.
- ___ g. Once the installation is complete a window appears indicating the installation was successful. Click **Next** to continue with the wizard.
- ___ 3. Install the WebSphere Application Server plug-in.

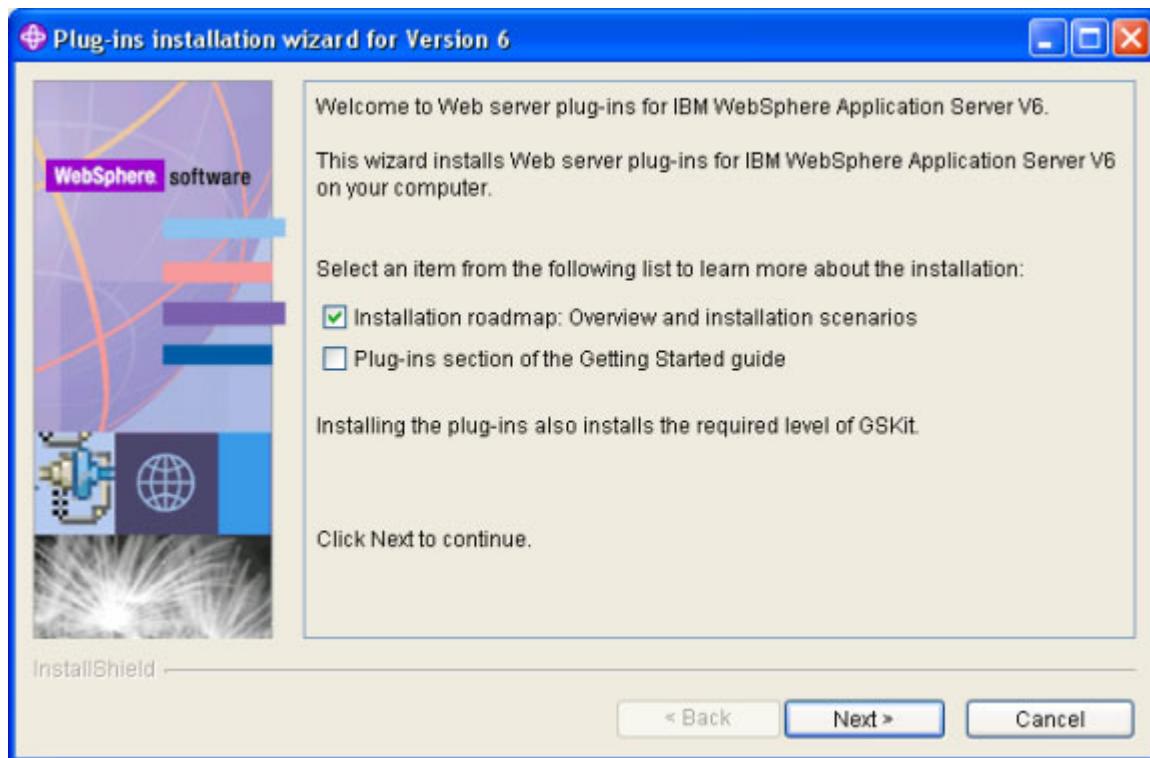
- __ a. Ensure the **Launch the WebSphere Application Server - Plugin Install** option is selected.



Information: After installing the application server and a Web server, you can install a WebSphere Application Server plug-in for a supported Web server. The binary plug-in is unique for each type of Web server. The purpose of the plug-in is to provide the communication protocol between the Web server and the application server.

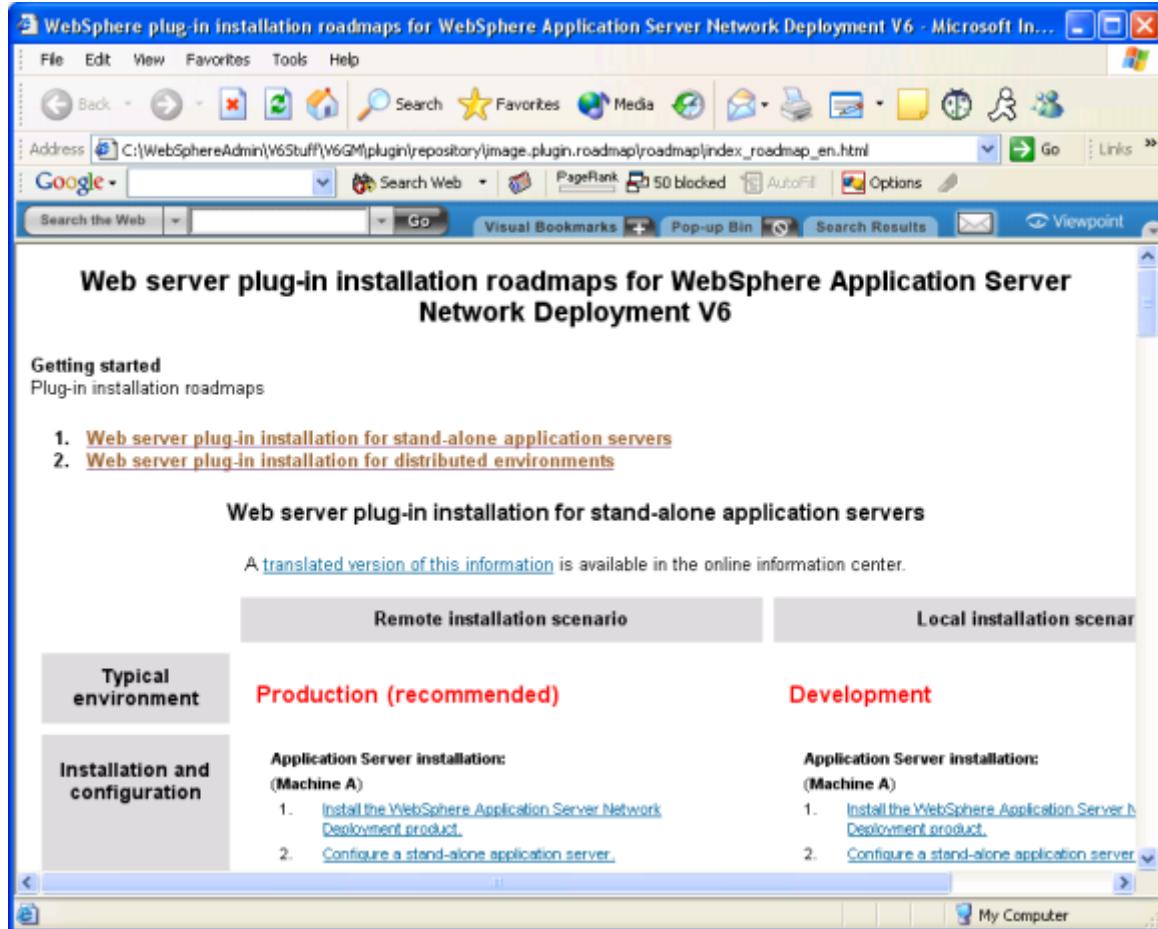
Click **Finish** to continue with the installation.

- __ b. On the welcome screen for the plug-in select the **Installation roadmap** option. Click **Next**.



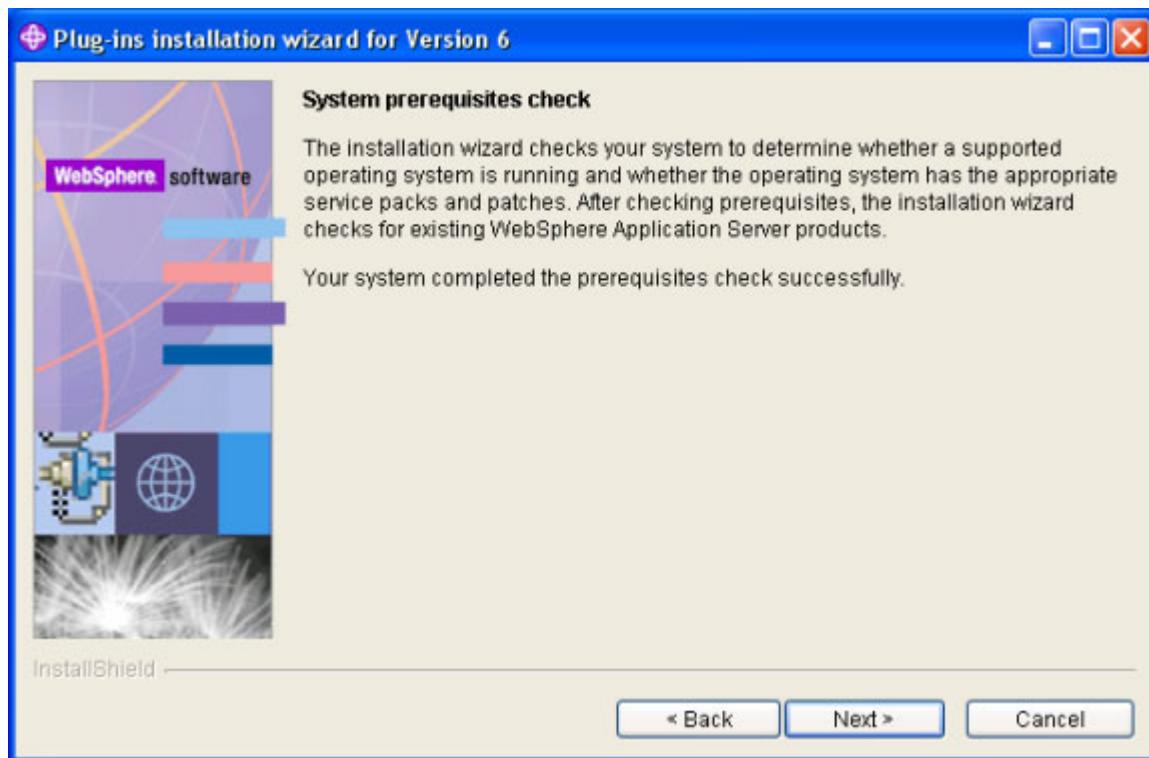
This launches another window for the Installation roadmap. This contains detailed information about the Web server plug-in installation for both a local and

remote installation. Feel free to look through this information. Close the browser window when you are done.



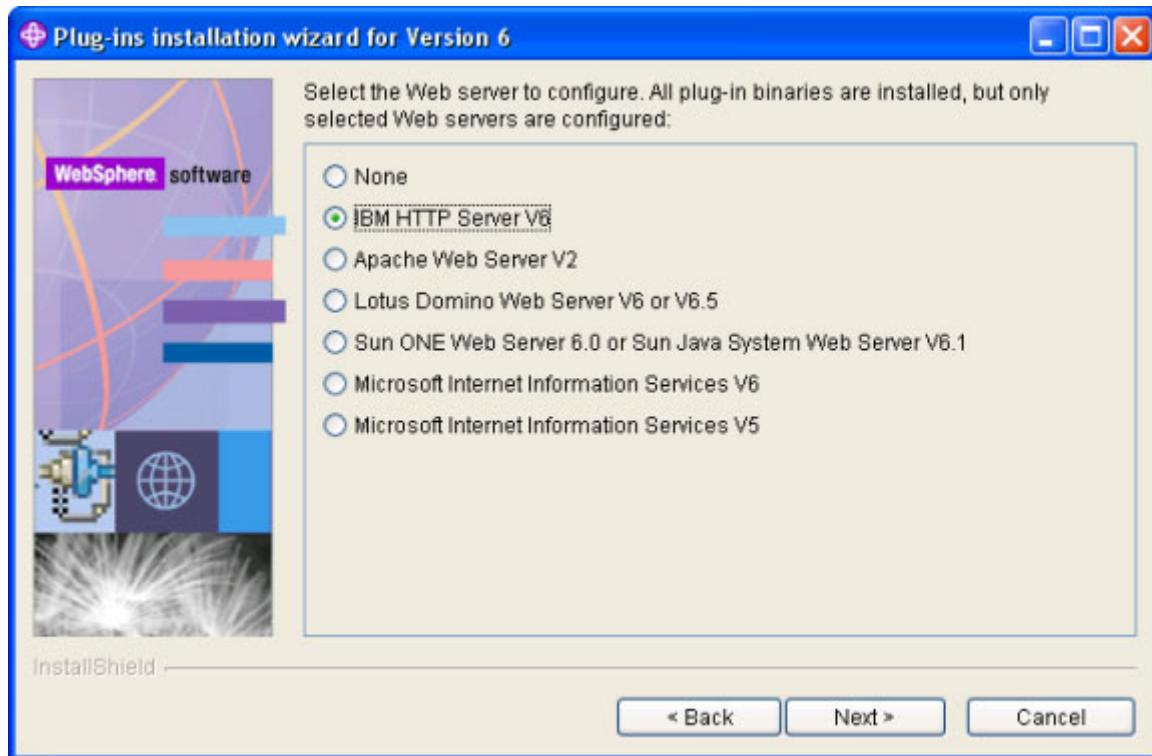
- ___ c. Accept the license agreement and click **Next**.
- ___ d. The installation wizard checks system prerequisites. It looks for a supported operating system with required fixes and patches and checks for existing

WebSphere Application Server products. Click **Next** when the prerequisite check is complete.



UNIX: If the lab system does not meet the prerequisites check with your instructor before continuing with the installation.

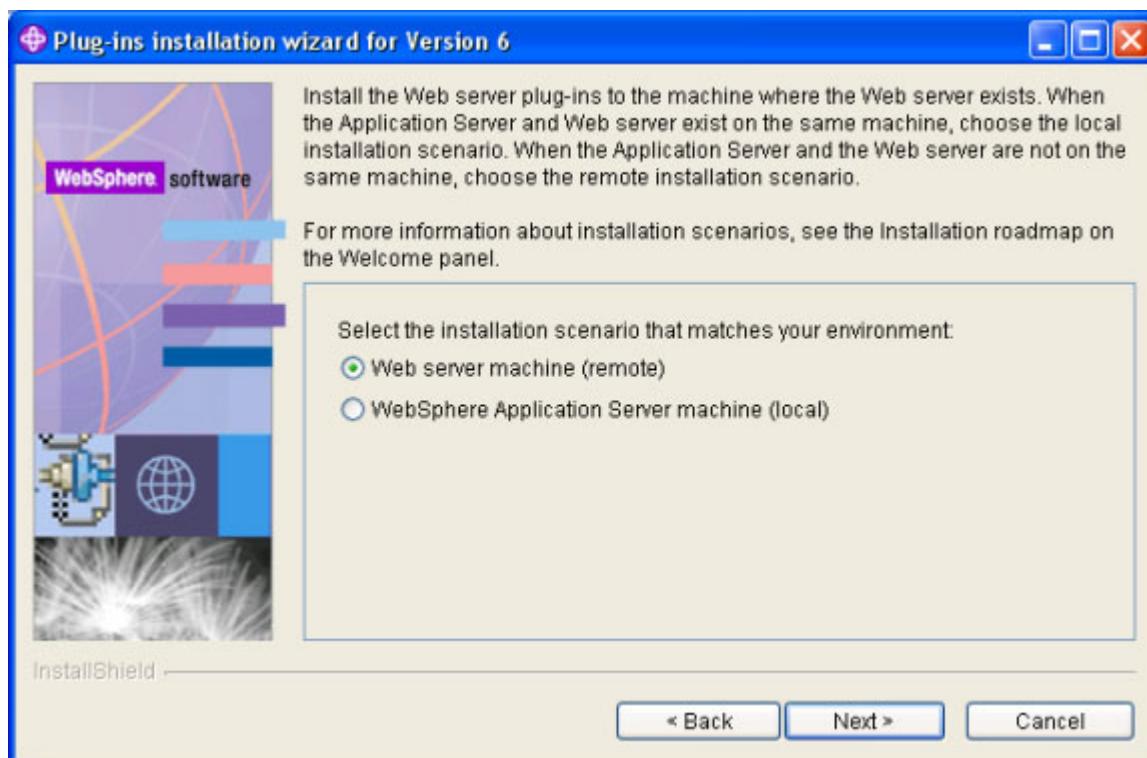
- ___ e. Select **IBM Http Server** and click **Next**.



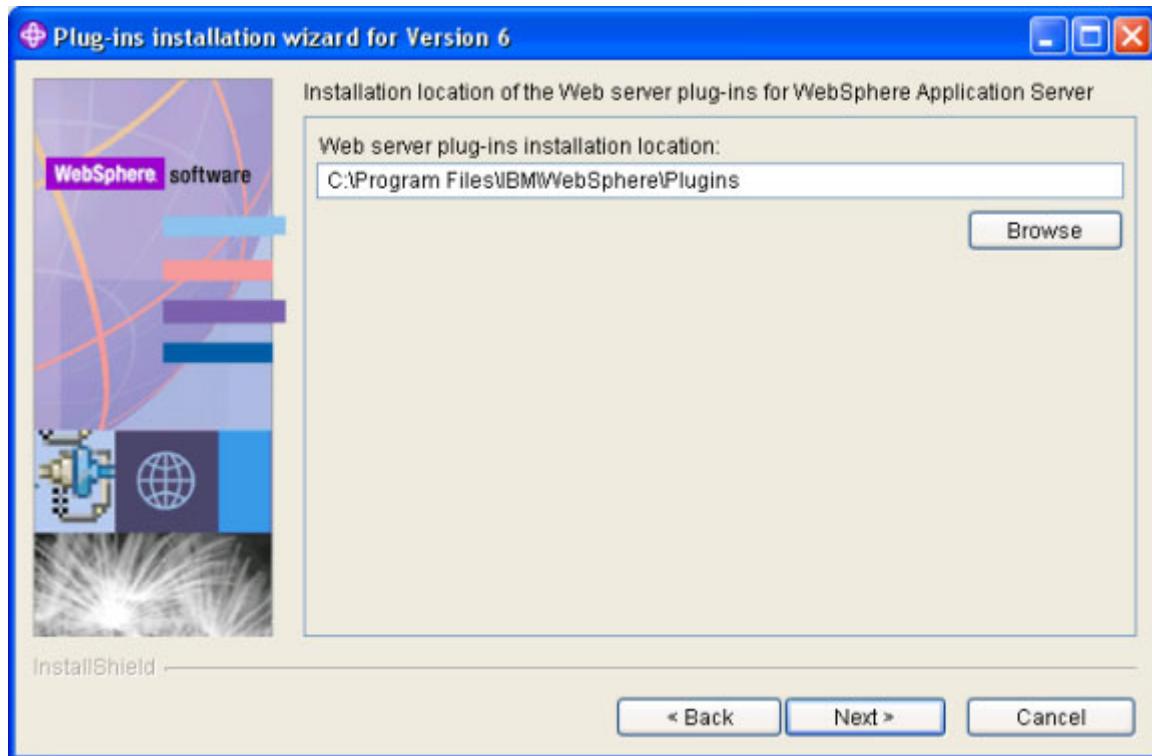
Information: In the figure above you can see all the other Web servers that are supported.

- ___ f. Select the installation scenario that matches your environment. When the application server and Web server exist on the same machine, choose the local installation scenario. If they are not on the same machine you choose remote installation.

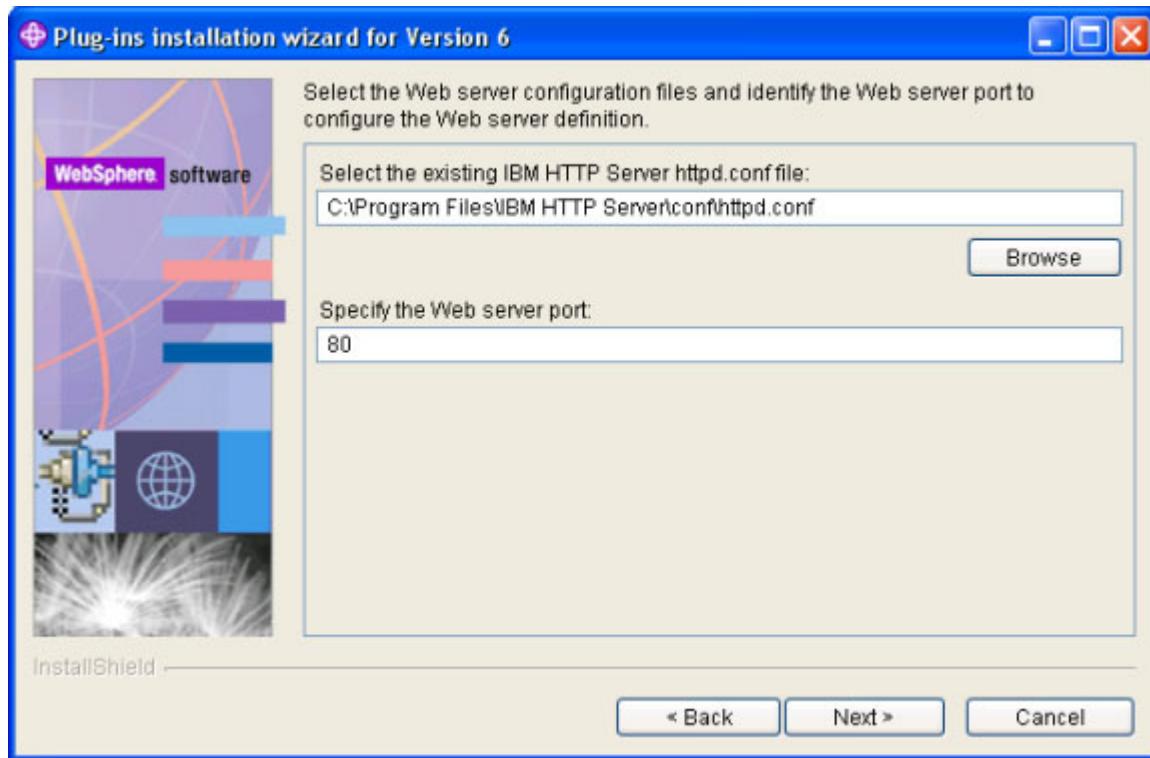
However, for the labs, specify the remote installation scenario. The remote configuration resembles more of a real world scenario. Select **Web server machine (remote)**. Click **Next**.



- g. Specify the Web server plug-ins installation location. Specify the fully qualified path to where the plug-in modules are installed. By default this value is filled in. Keep the default. Click **Next**.



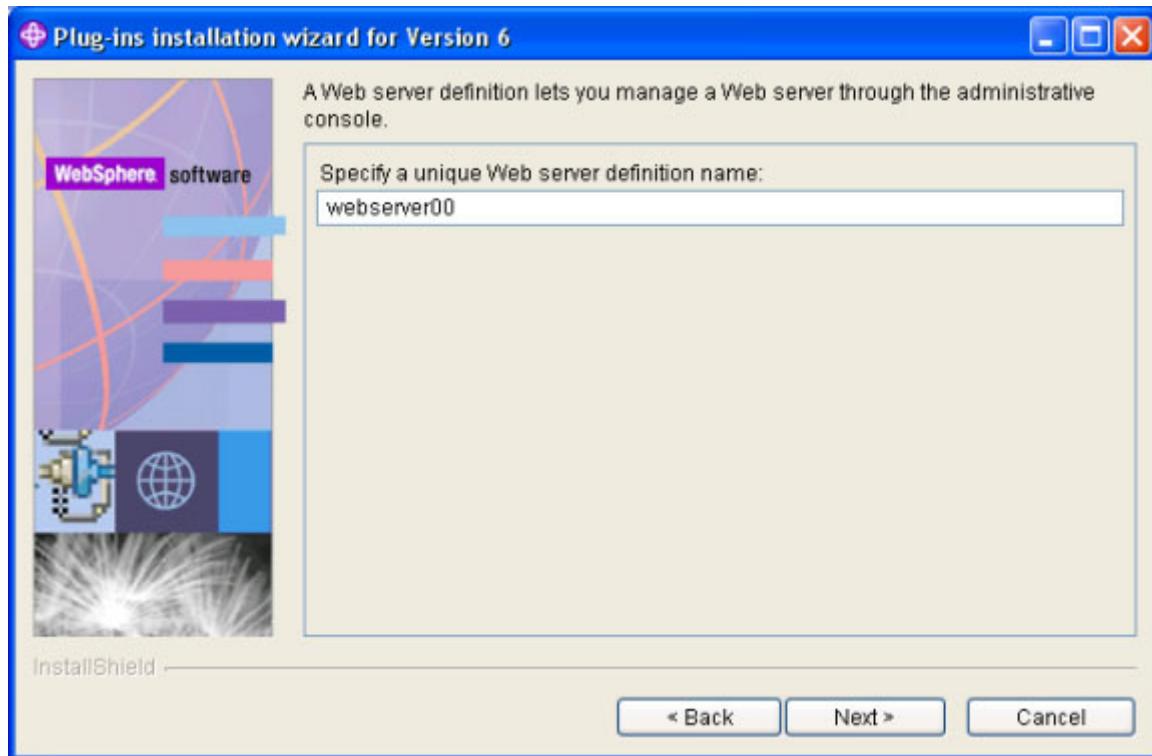
- ___ h. Select the existing IBM Http Server **httpd.conf** configuration file. Browse to the **C:\Program Files\IBM HTTP Server\conf** folder and select **httpd.conf**. Keep the default for the Web server port, port 80. Click **Next**.



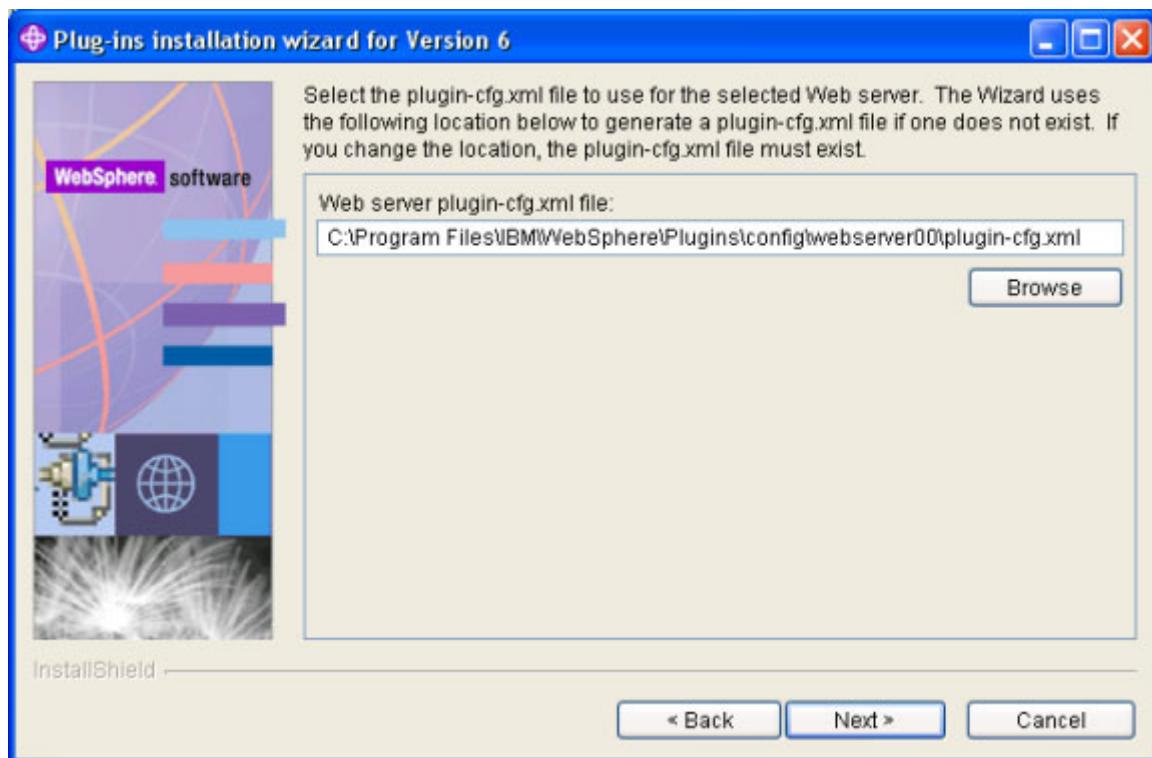
UNIX: In UNIX based systems, the directory structures will be different. Refer to the appropriate **[Appendix]** for the appropriate directory structures. The install wizard should give the appropriate default values for these directories.

- ___ a. Browse to **<ihs_root>/conf** and select **httpd.conf**. Keep the default for the Web server port, port 80.

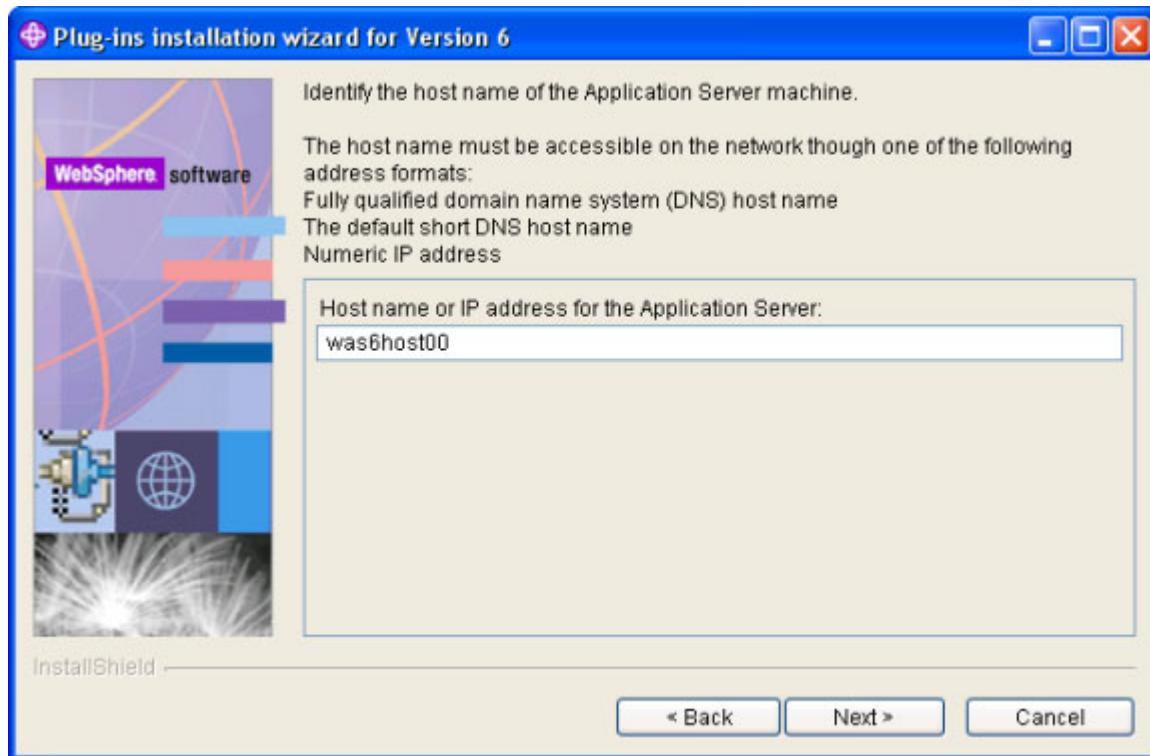
- ___ b. Configure a unique name for the Web server definition on the next window. Specify **webserverXX** for the definition name. Remember to use your machine number in place of **XX**. Click **Next**.



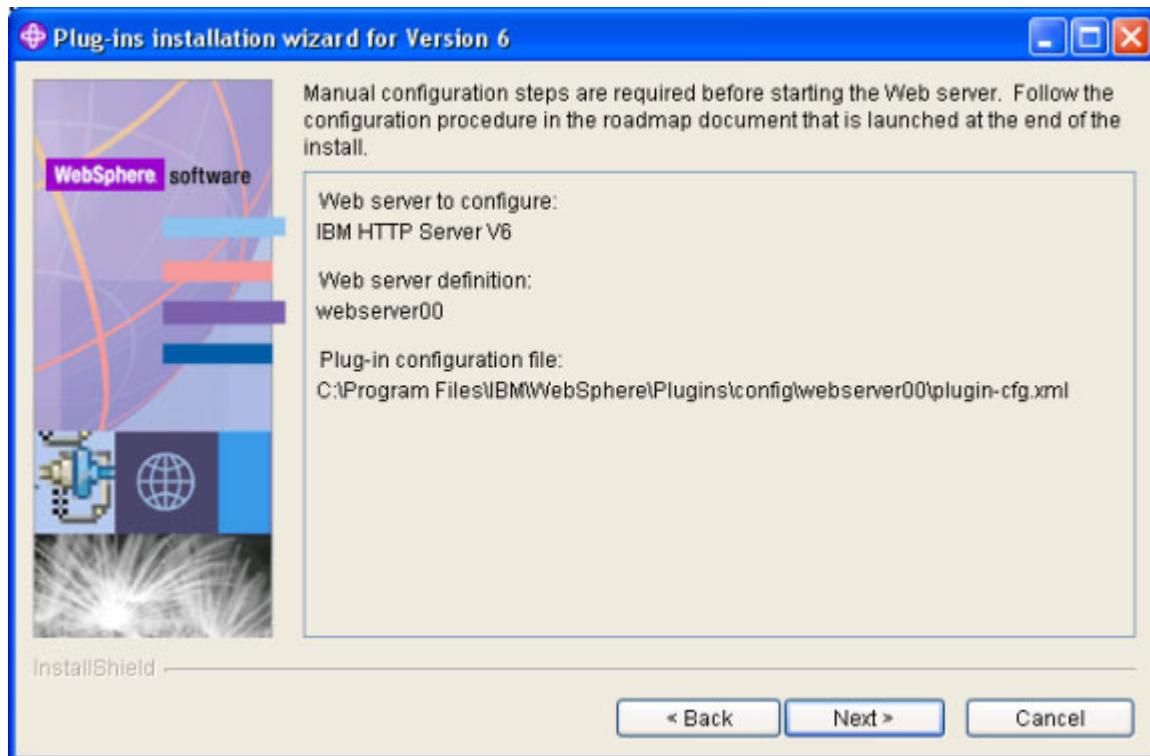
- ___ c. Select the plugin-cfg.xml file to use for the selected Web server. Keep the default location. Click **Next**.



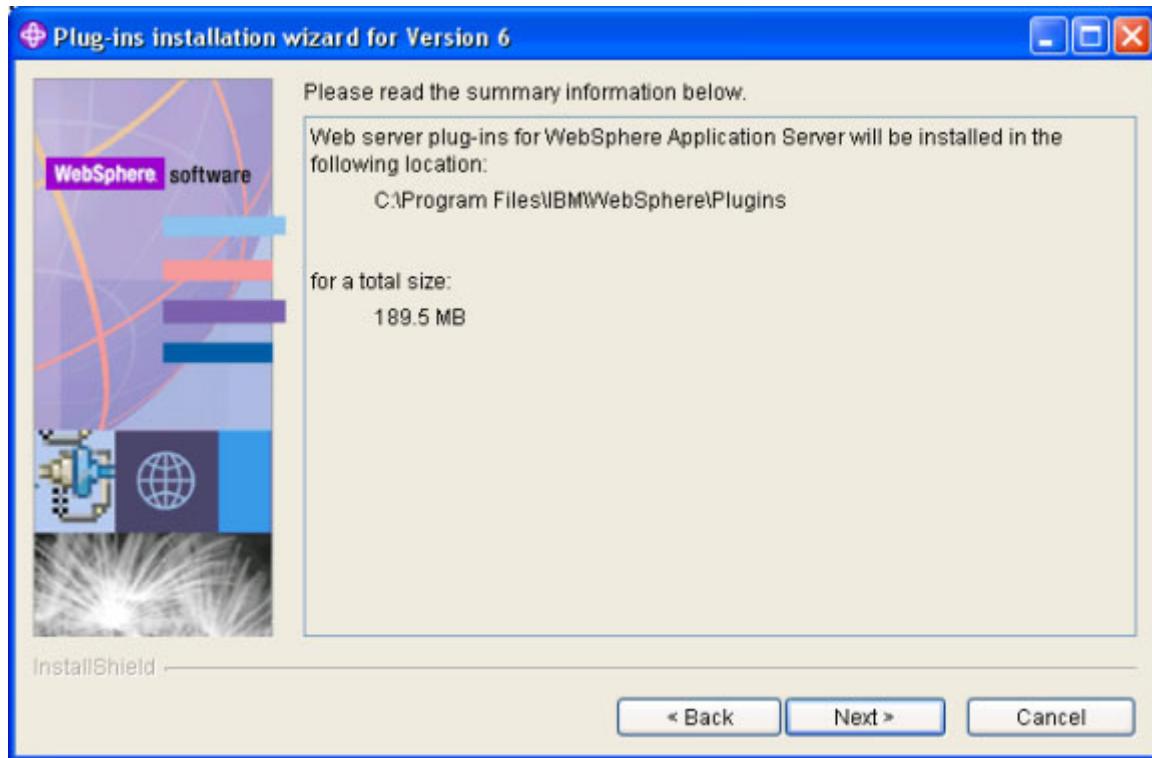
- ___ d. Specify the host name of the application server machine. Specify the machine name **was6hostXX**. Click **Next**.



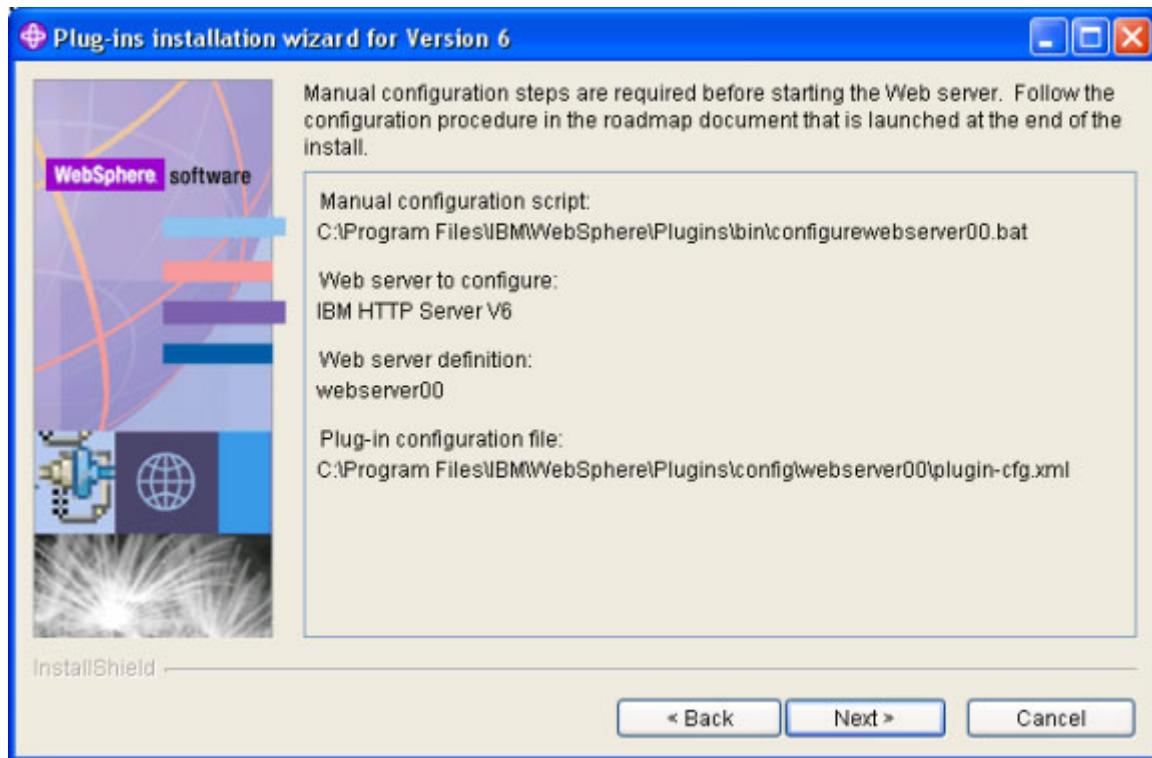
- ___ e. An information window regarding the manual configuration steps that are required before starting the Web server appears. Click **Next** to continue to the summary.



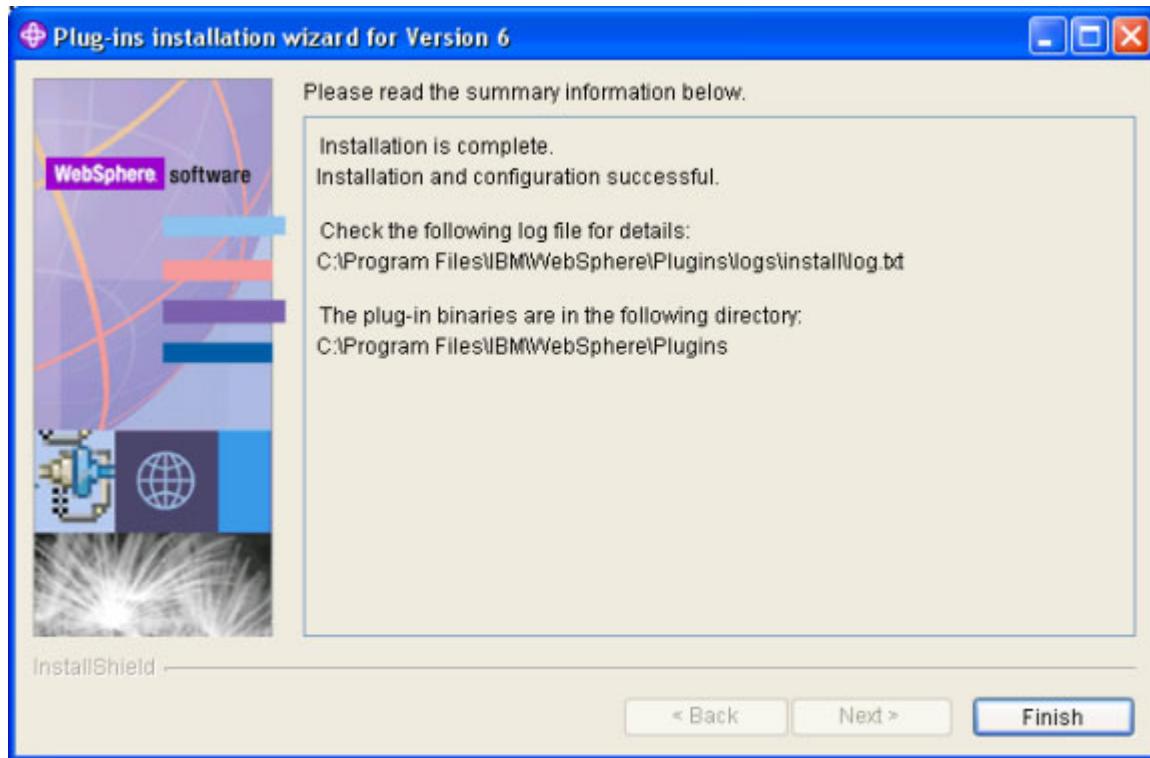
- ___ f. On the Plug-ins installation wizard for Version 6 window click **Next**.



- ___ g. When the installation of the Web server plug-ins is complete another manual configuration information window appears. Click **Next**.



- ___ h. The final summary of the installation appears. The Installation roadmap also launches. Close the browser window. Click **Finish** to complete the installation wizard.



- ___ i. Close the launchpad.

Web server administration server

Now that you have installed the IBM Http Server create a user ID for use for the IBM Http Server administration server.

- ___ 1. Start the IBM Http Server and the IBM HTTP administration server.
 - ___ a. Click **Start** → **Settings** → **Control Panel**.
 - ___ b. Double-click **Administrative Tools**.
 - ___ c. Double-click **Services**.
 - ___ d. Look for the **IBM HTTP Server 6.0** and **IBM HTTP Administration 6.0** services.
 - ___ e. Right-click **IBM HTTP Administration 6.0** and click **Start**.
 - ___ f. Right-click **IBM HTTP Server 6.0** and click **Start**.

- ___ g. Verify that both are running. The status for these should be **Started**.

UNIX: In a UNIX environment, processes will have to be checked and started using Terminal commands.

- ___ a. From a command line enter **ps -ef | grep httpd**
- ___ b. If no httpd processes are running from the <ihs_root>, navigate to <ihs_root>/bin and execute **./apachectl start**.
- ___ a. Start the administrative server using the **./adminctl start** command.

- ___ 2. Verify that the IBM HTTP Server administrative server has write access to the plugin-cfg.xml file. On Windows this is not an issue.

UNIX: On UNIX systems, if WebSphere is configured to remotely propagate the plugin-cfg.xml file through the IHS administrative process, the IHS administrative process needs to have write access to the plugin-cfg.xml file. By default, when the plug-in is first installed, a generic plugin-cfg.xml is created in the plug-in's config directory. The default protection bits for the plugin-cfg.xml are set to **-rw-r--r--** (644) and owned by whichever uid installed WebSphere. Since, by default, the IHS administrative process runs as the uid nobody, it does not have the required access to write new plugin-cfg.xml files when WebSphere attempts to propagate.

There are a number of possible solutions. One would be to change the uid used to run the IHS administrative process to match the owner of the plugin-cfg.xml file. Another solution, which will be used in this lab, would be to add world write access to the plugin-cfg.xml file (**-rw-rw-rw-** or 666).

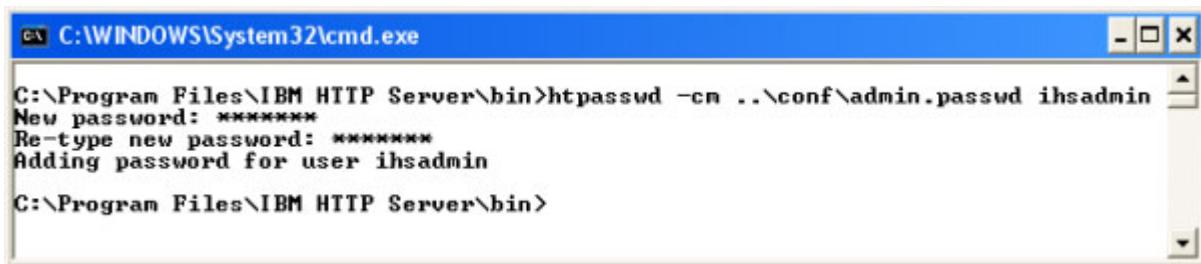
- ___ a. In a shell window, navigate to <plugin_root>/config/webserverXX
- ___ b. Execute the command **chmod a+w plugin-cfg.xml**
- ___ c. Enter an **ls -l** to verify that the plugin-cfg.xml now has the appropriate settings.

- ___ 3. Before working with the IBM Http Server administrative server a user ID and password must be set. This is a step usually performed by the Web server administrator but has not been performed on the lab machine yet. Set the user ID **ihsadmin** and password **was1edu**.

- ___ a. In a command prompt navigate to <ihs_root>\bin.
- ___ b. Enter the following command:

```
htpasswd -cm ..\conf\admin.passwd ihsadmin
```

- ___ c. When prompted for a password enter **was1edu**. Confirm the password and enter again.



A screenshot of a Windows Command Prompt window titled 'C:\WINDOWS\System32\cmd.exe'. The window shows the following command being run and its output:

```
C:\Program Files\IBM HTTP Server\bin>htpasswd -cm ..\conf\admin.passwd ihsadmin
New password: *****
Re-type new password: *****
Adding password for user ihsadmin
C:\Program Files\IBM HTTP Server\bin>
```

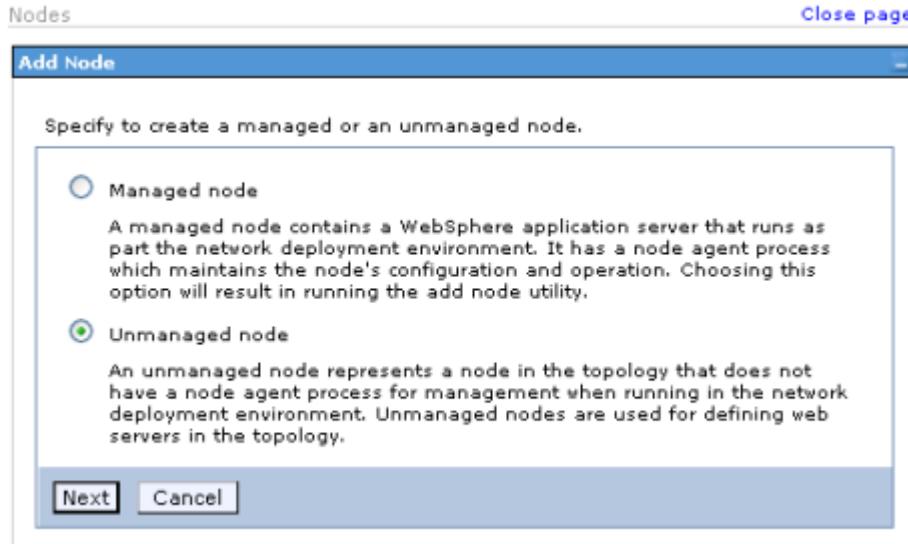
UNIX:

- ___ a. In a command prompt navigate to <ihs_root>/bin.
- ___ b. Enter the following command:
- ```
htpasswd -cm ../conf/admin.passwd ihsadmin
```
- \_\_\_ a. When prompted for a password enter **was1edu**. Confirm the password and enter again.
- \_\_\_ 4. Verify the IBM Http Server is running.
- \_\_\_ a. Open a browser and enter the address **http://localhost**
- \_\_\_ b. You should see the welcome page for the IBM Http Server.

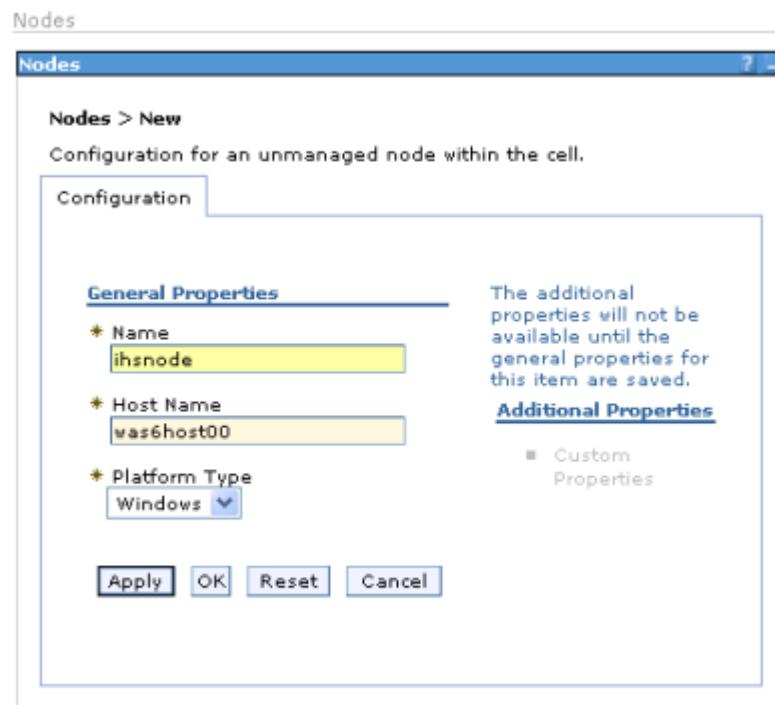
## Create a new node

Create a new node and add the Web server to the node. When adding a node you may create either a managed or 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. Verify the deployment manager is running.
  - \_\_\_ a. Navigate to <profile\_root>\DmgrProfile\bin and execute the **serverStatus -all** command.
  - \_\_\_ b. If the deployment manager is not running start it using the **startManager** command.
  - \_\_\_ c. Launch the administrative console for the deployment manager.
  - \_\_\_ d. Log in as the user ID **admin**.
- \_\_\_ 2. Create a new unmanaged node for the Web server. An unmanaged node does not have a node agent and is used for defining Web servers.
  - \_\_\_ a. In the navigation tree, expand **System administration** and click **Nodes**.
  - \_\_\_ b. Click the **Add Node** button.
  - \_\_\_ c. In the Add Node window select **Unmanaged node** and click **Next**.



- \_\_\_ d. In the Nodes window specify configuration information for the node. Specify the name **ihsnode** and **was6hostXX** for Host Name. The Platform Type is **Windows**. Click **OK**.



- \_\_\_ e. The node **ihsnode** should be added to the list of nodes.

| Nodes                                                                                                                                                                                                                                                                                                                                       |                                    |         |                    |        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|---------|--------------------|--------|
| A node corresponds to a physical computer system with a distinct IP host address. The node name is usually the same as the host name for the computer. The following table lists the nodes in this cell. You can add or remove nodes from the cell by clicking on "Add Node" and specifying a remote, running WebSphere Application Server. |                                    |         |                    |        |
| <input type="checkbox"/> Preferences<br><input type="button" value="Add Node"/> <input type="button" value="Remove Node"/> <input type="button" value="Force Delete"/> <input type="button" value="Synchronize"/> <input type="button" value="Full Resynchronize"/>                                                                         |                                    |         |                    |        |
| Select                                                                                                                                                                                                                                                                                                                                      | Name                               | Version | Discovery Protocol | Status |
| <input type="checkbox"/>                                                                                                                                                                                                                                                                                                                    | <u>ihsnode</u>                     | Unknown | TCP                |        |
| <input type="checkbox"/>                                                                                                                                                                                                                                                                                                                    | <u>was6host00Node02</u>            | 6.0.0.1 | TCP                |        |
| <input type="checkbox"/>                                                                                                                                                                                                                                                                                                                    | <u>was6host00Node03</u>            | 6.0.0.1 | TCP                |        |
| <input type="checkbox"/>                                                                                                                                                                                                                                                                                                                    | <u>was6host00hostCellManager01</u> | 6.0.0.1 | TCP                |        |

- \_\_\_ 3. Save the changes to master configuration.

## Add the Web server to the configuration

In this step is to add the Web server to the ihsnode.

Add the Web server to the configuration using the unmanaged node ihsnode.

- \_\_\_ 1. Add the Web server to the ihsnode configuration.
  - \_\_\_ a. In the administrative console select **Servers** —> **Web servers**.
  - \_\_\_ b. Click **New** to add a Web server.
  - \_\_\_ c. On step 1 of creating a new Web server entry select the node **ihsnode** from the drop-down menu. Specify the server name of **webserverXX**.

**Information:** The Web server name must match the name that was assigned during the IBM Http Server installation.

Click **Next**.

- \_\_\_ d. On step 2, specify the properties for the new Web server. Enter the following information in the fields as provided:

| Field Name                    | Value               |
|-------------------------------|---------------------|
| Type                          | IHS                 |
| Port                          | 80                  |
| Installation Path             | <ihs_root>          |
| Service name                  | IBM HTTP Server 6.0 |
| Plug-in installation location | <plugin_root>       |

**UNIX:** The field name **Service name** will not be an option.

Click **Next** to continue.

Web servers Close page

**Create new Web server entry**

Create a new Web server entry.

Step 1: Select a node

→ **Step 2: Enter the properties for the new Web server**

Step 3: Select a Web server template

Step 4: Confirm new Web server

**Enter the properties for the new Web server**

Enter the Web server properties.

\* Type: IHS

\* Port: 80

Installation path: C:\Program Files\IBM HTTP Server

\* Service name: IBM HTTP Server 6.0

Use secure protocol

Enter Plugin property

\* Plug-in installation location: C:\Program Files\IBM\WebSphere\Plugin

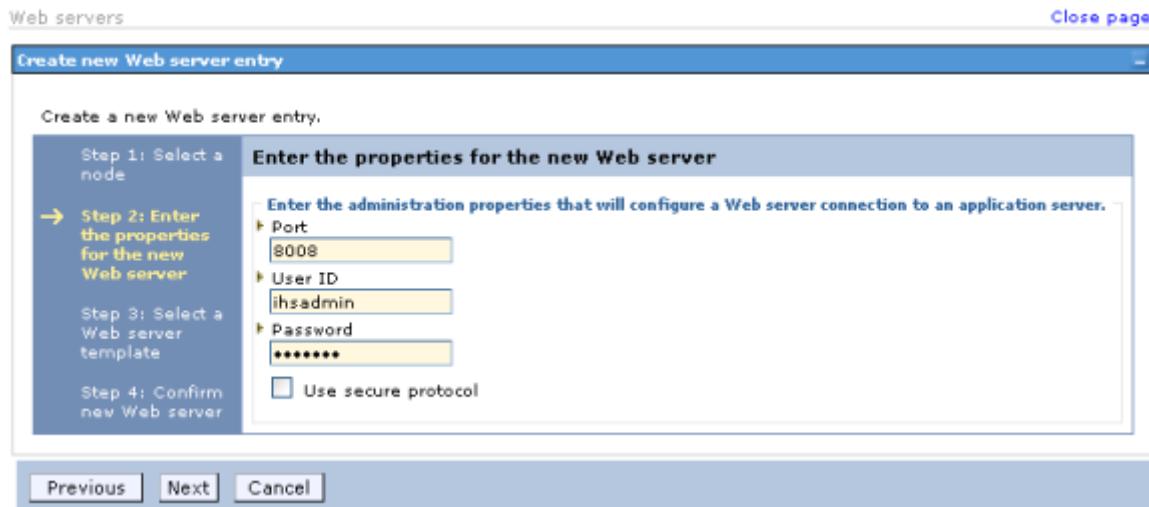
**Previous** **Next** **Cancel**

**UNIX:** The Create new Web Server entry page will have a different look.

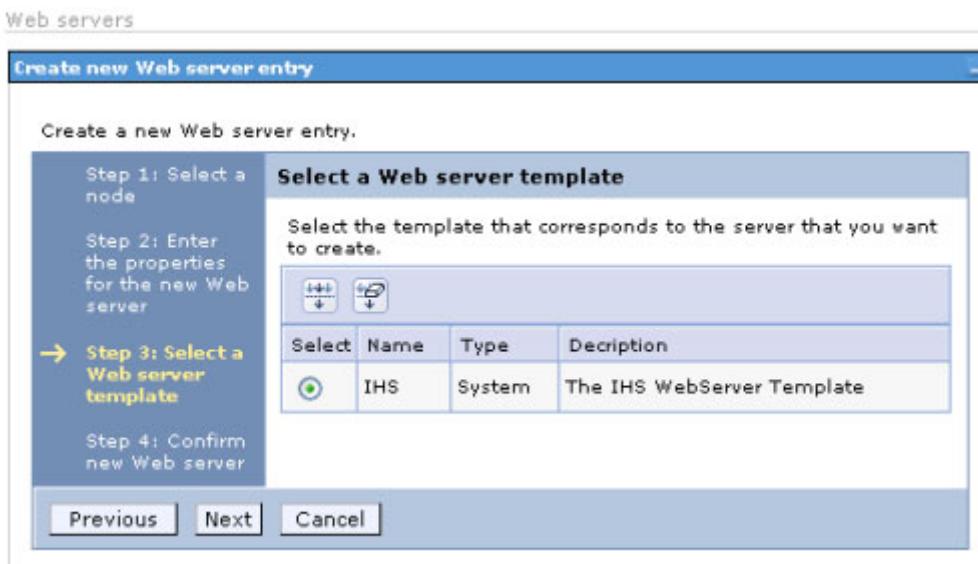
- \_\_\_ e. In order for the administrative console to access the IBM HTTP administration server, you must define a valid user ID and password. Enter the following information in the fields as provided:

| Field Name | Value    |
|------------|----------|
| Port       | 8008     |
| User ID    | ihsadmin |
| Password   | was1edu  |

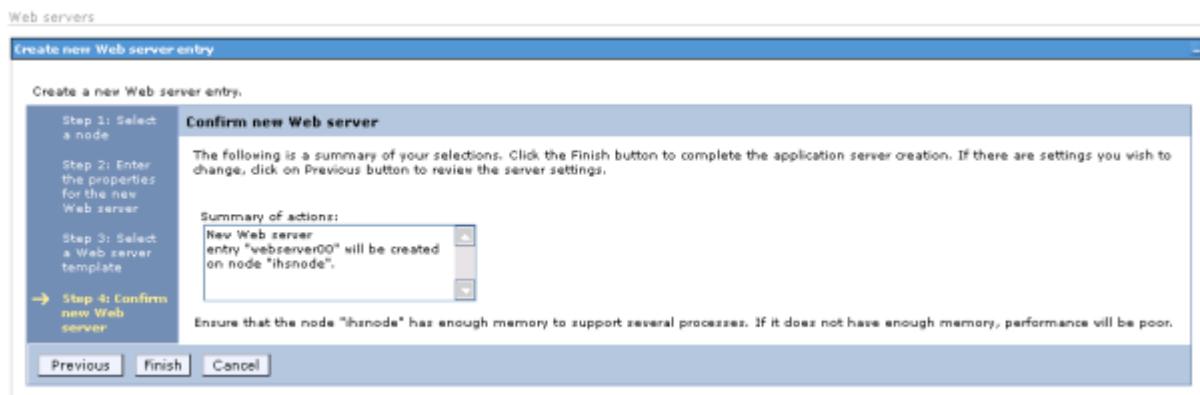
Click **Next** to continue.



- \_\_ f. On step 3, select **IHS** for the Web server template. Click **Next**.



- \_\_\_ g. On step 4, the summary, click **Finish**.



- \_\_\_ 2. Save the changes to master configuration.
- \_\_\_ 3. Stop and start the Web server using the administrative console.
- \_\_\_ a. Select **Servers** —> **Web servers**. The Web server is currently running.
- \_\_\_ b. Select the Web server **webserverXX** and click **Stop**. The Web server status should now be Stopped.
- \_\_\_ c. Verify in Windows services that the Web server is not running.
- UNIX:** Check for running http processes using **ps -ef | grep http**
- \_\_\_ d. Start the Web server before continuing. Select the Web Server and click **Start**.
- \_\_\_ e. To verify the server has started, open a Web browser and connect to the IBM Http Server welcome page. Specify the following address: **http://localhost**.

## **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, cluster of application servers or a Web server. Web servers specified as targets will have routing information for the application generated in the plug-in configuration file for the Web server.

This mapping takes place during application deployment. Since the DefaultApplication was already deployed when you installed the Web server, you need to map the modules to the Web server.

- \_\_\_ 1. Using the Deployment Manager's administrative console map the DefaultApplication's modules to the Web server.
- \_\_\_ a. Select **Applications** —> **Enterprise Applications** —> **DefaultApplication**.

\_\_ b. Under Additional Properties select Map modules to servers.

Enterprise Applications

**Enterprise Applications**

[Enterprise Applications > DefaultApplication](#)

Enterprise Applications

Configuration Local Topology

**General Properties**

- \* Name: DefaultApplication

**Binary Management**

- \* Application binaries: \${APP\_INSTALL\_ROOT}/\${CE}
- Use metadata from binaries
- Enable distribution

Validation: warn

**Class Loading and File Update Detection**

- \* Class loader mode: Parent First
- \* WAR class loader policy: Module
- Enable class reloading

Reloading interval: 3

**Additional Properties**

- [Stateful session bean failover settings](#)
- [Session management](#)
- [Application profiles](#)
- [Libraries](#)
- [Target mappings](#)
- [Last participant support extension](#)
- [View Deployment Descriptor](#)
- [Provide JMS and EJB endpoint URL information](#)
- [Publish WSDL files](#)
- [Provide HTTP endpoint URL information](#)
- [Map security roles to users/groups](#)
- [Provide JNDI Names for Beans](#)
- [Map EJB references to beans](#)
- [Map data sources for all 2.x CMP beans](#)
- [Provide default data source mapping for modules containing 2.x entity beans](#)
- [Map virtual hosts for Web modules](#)
- [Map modules to servers](#)

- \_\_\_ c. In the selecting server window select **webserverXX** and **server1**. Click the box to the left of **Default Web Application** to select the module. Click **Apply**.

[Enterprise Applications > DefaultApplication > Selecting servers](#)

Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules 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 will 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 which are routed through it.

Clusters and Servers:

WebSphere:cell=was6host00Cell01,node=ihsnode,server=webserver00  
WebSphere:cell=was6host00Cell01,node=was6host00Node2,server=server1

**Apply**

| Select                              | Module                         | URI                                       | Server                                                                                                                                 |
|-------------------------------------|--------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <input type="checkbox"/>            | Increment Enterprise Java Bean | Increment.jar,META-INF/ejb-jar.xml        | WebSphere:cell=was6host00Cell01,node=was6host00Node2,server=server1                                                                    |
| <input checked="" type="checkbox"/> | Default Web Application        | DefaultWebApplication.war,WEB-INF/web.xml | WebSphere:cell=was6host00Cell01,node=was6host00Node2,server=server1<br>WebSphere:cell=was6host00Cell01,node=ihsnode,server=webserver00 |

**OK** **Cancel**

- \_\_\_ d. Click **OK**.
- \_\_\_ 2. Save the changes to master configuration.
- \_\_\_ 3. Verify the DefaultApplication is mapped to both the Web server and server1.
- \_\_\_ a. In the administrative console select **Applications** —> **Enterprise Applications** —> **DefaultApplication**.

\_\_ b. Under Additional Properties select Target mappings.

The screenshot shows the 'Enterprise Applications' configuration interface for the 'DefaultApplication'. The 'Additional Properties' section is expanded, and the 'Target mappings' link is highlighted with a red oval. Other links in this section include:

- [stateful session bean failover settings](#)
- [Session management](#)
- [Application profiles](#)
- [Libraries](#)
- **[Target mappings](#)** (highlighted)
- [Last participant support extension](#)
- [View Deployment Descriptor](#)
- [Provide JMS and EJB endpoint URL information](#)
- [Publish WSDL files](#)
- [Provide HTTP endpoint URL information](#)
- [Map security roles to users/groups](#)
- [Provide JNDI Names for Beans](#)
- [Map EJB references to beans](#)
- [Map data sources for all 2.x CMP beans](#)
- [Provide default data source mapping for modules containing 2.x entity beans](#)
- [Map virtual hosts for Web modules](#)
- [Map modules to servers](#)

The 'General Properties' section contains fields for Name (DefaultApplication), Application binaries (\$APP\_INSTALL\_ROOT)/\$(CEI), Use metadata from binaries (unchecked), Enable distribution (checked), Validation (warn dropdown), Class loader mode (Parent First dropdown), WAR class loader policy (Module dropdown), Enable class reloading (unchecked), and Reloading interval (3). The 'Startup Options' section contains Starting weight (1), Enable background application (unchecked), and Create MBeans for resources (unchecked).

- \_\_\_ c. You should see the DefaultApplication mapped to **webserverXX** and **server1**.

**Enterprise Applications**

**Enterprise Applications > DefaultApplication > Target mappings**

This list displays the mapping of a deployed object, such as an application or module, into a server or cluster environment. The status of the enterprise application or module on each server or cluster is displayed.

Preferences

| Select                   | Target                      | Node             | Version | Status |
|--------------------------|-----------------------------|------------------|---------|--------|
| <input type="checkbox"/> | <a href="#">server1</a>     | was6host00Node02 | 6.0.0.1 |        |
| <input type="checkbox"/> | <a href="#">webserver00</a> | ihsnode          | 6.0.0.1 |        |

Total 2

## Working with the plug-in configuration file

The plug-in configuration file contains routing information for all applications mapped to the Web Server. The plug-in configuration file needs to be regenerated and propagated to the Web server when there are changes to the WebSphere configuration that affect how requests are routed from the Web server to the application server.

- \_\_\_ 1. Regenerate the plug-in configuration file.
- \_\_\_ a. In the administrative console select **Servers** → **Web servers**.
- \_\_\_ b. Click the box to the left of the Web server. Click **Generate Plug-in**.

**Web servers**

**Web servers**  
A list of installed Web servers.

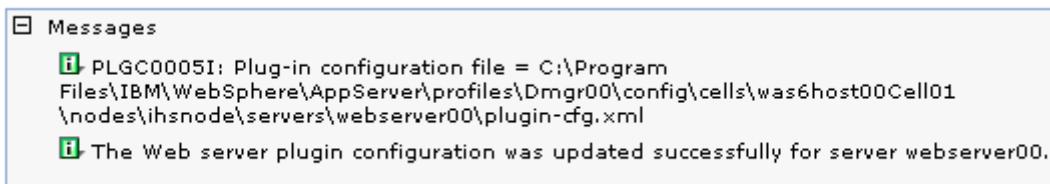
Preferences

**Generate Plug-in** (highlighted with a red circle)

| Select                              | Name                        | Node    | Version | Status |
|-------------------------------------|-----------------------------|---------|---------|--------|
| <input checked="" type="checkbox"/> | <a href="#">webserver00</a> | ihsnode | 6.0.0.1 |        |

Total 1

- \_\_\_ c. Verify the generation was successful by viewing the messages.



- \_\_\_ 2. View the plug-in configuration file, **plugin-cfg.xml**, from the administrative console.
- \_\_\_ a. Select the Web server **webserverXX**.
- \_\_\_ b. Under **Additional Properties** select **Plug-in properties**.
- \_\_\_ c. Under **Plug-in properties** select the **View** button to see the plugin-cfg.xml file.

- \_\_\_ d. The next window shows the plug-in configuration file. Verify that the element:  
<UriGroup Name="default\_host\_server1\_was6host00Node02\_Cluter\_URIs">  
includes the element

```
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
Name="/snoop/*/>
```

around the middle of the file. This ensures URLs containing **/snoop** will be recognized by the plug-in and forwarded to the application server.

Plug-in configuration file

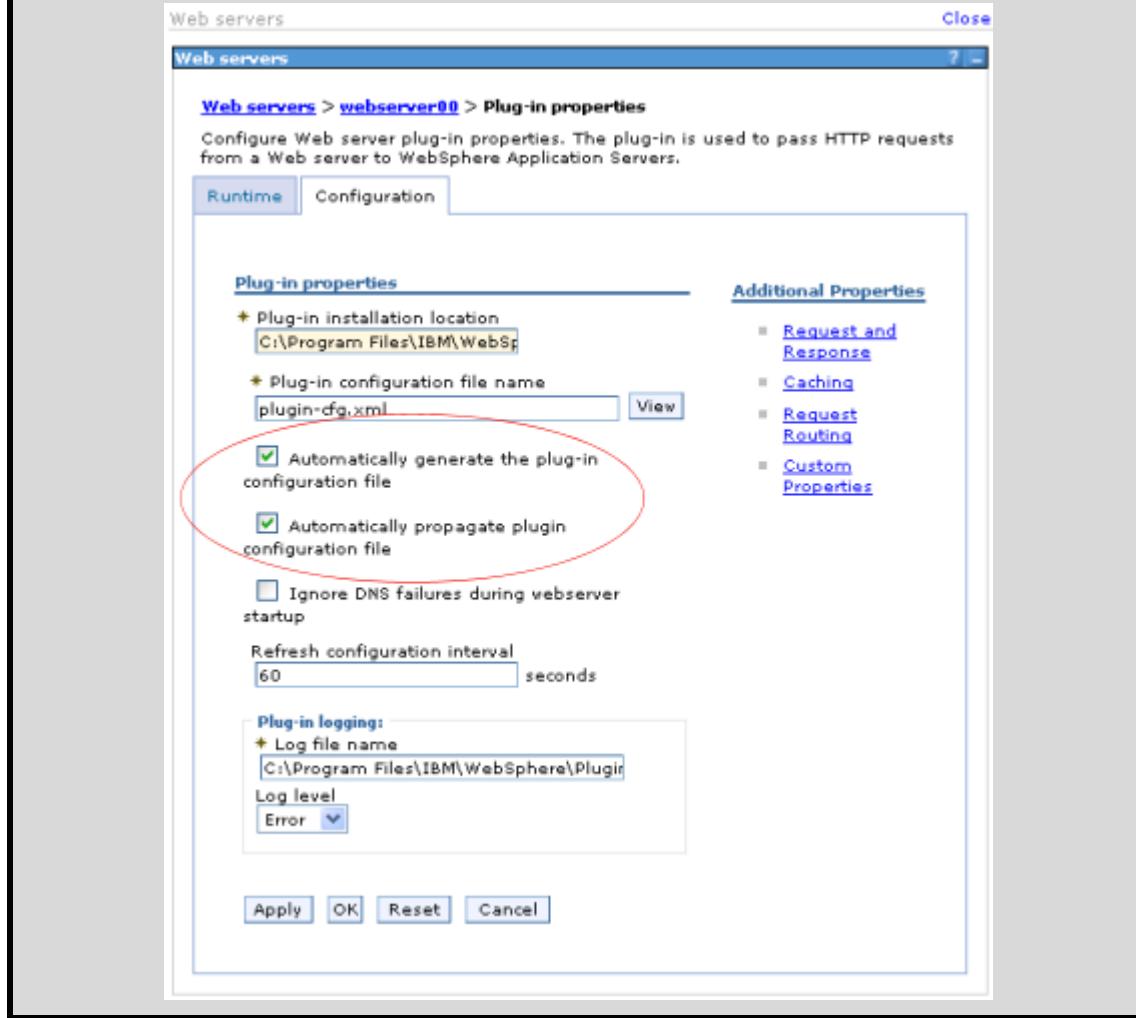
```
<?xml version="1.0" encoding="ISO-8859-1"?><!--HTTP server plugin config file for the webserver was6host00Ce
<Config ASDisableNagle="false" AcceptAllContent="false" AppServerPortPreference="HostHeader" ChunkedResponse="true"
<Log LogLevel="Error" Name="C:\Program Files\IBM\WebSphere\Plugins\logs\webserver00\http_plugin.log"/>
<Property Name="ESIEnable" Value="true"/>
<Property Name="ESIMaxCacheSize" Value="1024"/>
<Property Name="ESIValidationMonitor" Value="false"/>
<VirtualHostGroup Name="default_host">
<VirtualHost Name="*:9080"/>
<VirtualHost Name="*:80"/>
<VirtualHost Name="*:9443"/>
</VirtualHostGroup>
<ServerCluster CloneSeparatorChange="false" LoadBalance="Round Robin" Name="server1_was6host00Node02_Cluster"
<Server ConnectTimeout="0" ExtendedHandshake="false" MaxConnections="-1" Name="was6host00Node02_server"
<Transport Hostname="was6host00Node02" Port="9080" Protocol="http"/>
<Transport Hostname="was6host00Node02" Port="9443" Protocol="https">
<Property Name="keyring" Value="C:\Program Files\IBM\WebSphere\Plugins\etc\plugin-key.kdb"/>
<Property Name="stashfile" Value="C:\Program Files\IBM\WebSphere\Plugins\etc\plugin-key.sth"/>
</Transport>
</Server>
</ServerCluster>
<UriGroup Name="default_host_server1_was6host00Node02_Cluster_URIs">
<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="*.jsu"/>
<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/*"/>
</UriGroup>
<Route ServerCluster="server1_was6host00Node02_Cluster" UriGroup="default_host_server1_was6host00Node02_Cluster_URIs" arnEnabled="false" loggingEnabled="false" rmEnabled="false" traceLevel="HOPS">
<filters enable="false" type="URI">
<filterValues enable="false" value="/snoop"/>

```

3. After a plug-in configuration file is regenerated it needs to be propagated to the Web server. You can propagate manually by copying the file from the application server

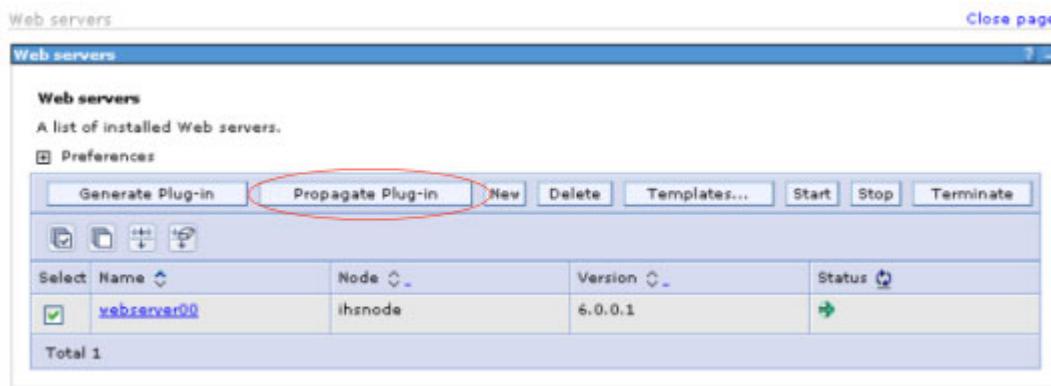
machine to the Web server machine or you can do it from the administrative console.

**Information:** The default is to automatically generate the plug-in configuration file and to propagate the plug-in to the Web server. You can view this information in the administrative console by selecting **Servers —> Web servers —> webserverXX**. Under Additional Properties select **Plug-in properties**.

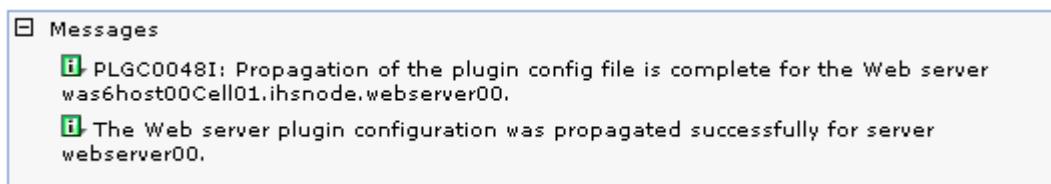


- \_\_\_ a. In the administrative console select **Servers —> Web servers**.

- \_\_\_ b. Click the box to the left of your Web server. Click **Propagate Plug-in**.



- \_\_\_ c. Verify the generation was successful by viewing the messages.



- \_\_\_ 4. Log out of the administrative console.

### ***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 the application server **server1** up and running.  
\_\_\_ 2. Access the snoop servlet by accessing the address:

`http://localhost:9080/snoop.`

by using the port 9080 you are bypassing the external IBM Http Server.

- \_\_\_ a. Open a Web browser.  
\_\_\_ b. Specify the address `http://localhost:9080/snoop.`  
\_\_\_ c. You should see details on the snoop servlet.
- \_\_\_ 3. Verify the Web server is forwarding requests to the application server by using the address `http://localhost/snoop`. This request will first go to the Web server.
- \_\_\_ a. Specify the address `http://localhost/snoop.`  
\_\_\_ b. You should see details on the snoop servlet.  
\_\_\_ c. Close the Web browser.

## End of lab

## Exercise review and wrap-up

The first part of the exercise looked at the installation for IBM Http Server and the WebSphere plug-ins. After configuring the IBM Http Server you added the Web server to the deployment managers configuration. You also mapped the DefaultApplication to the Web server and regenerated and propagated the plug-in configuration file.



# Exercise 11. Clustering

## What this exercise is about

This lab covers the creation of a cluster. While creating the cluster two nodes are added. After the cluster is created the Trade application is configured to run in the cluster. The application is tested to ensure workload management and failover work as expected.

A replication domain is setup to use memory-to-memory replication and again 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 new cluster
- Add cluster members to the cluster
- Configure resources at the cluster scope
- Create and configure a session replication domain
- Test the application in a clustered environment

## Introduction

Up to this point you have worked with WebSphere Application Server V6 in a single server environment. In this lab you continue after federating two nodes to the deployment manager cell. You create a cluster so that workload can be managed between two servers, one on each node.

You also setup a memory-to-memory replication domain so that Http sessions can be shared in case of failure of one of the servers.

## Required materials

To do this lab you require a properly set up computer with WebSphere Application Server V6 installed as well as the necessary startup and program files.

## Instructor exercise overview

This exercise creates a new cluster with two servers. The first server comes from the original profile created earlier in this class. The second server is part of the node that was created using a custom profile. The actual server is created at the time the cluster is created.

Testing is done both before and after the Distributed Session Management is configured, pointing out the differences.

The type of scaling you do in this exercise, is logical horizontal scaling. It is termed “logical” because all nodes are on the same machine. In this situation, where all servers are on the same machine, it doesn’t really make much difference if you do vertical or horizontal scaling. In fact vertical scaling would be more efficient because you would have less nodeagents. However, this scenario (using two nodes) is more realistic, even though both nodes are on the same machine.

If you have eager or advanced students in your class you might want to pair them up and have them do true horizontal scaling where they cluster one node from each machine. You might even suggest they use a separate machine for the deployment manager and for the Web server.

If your students get UnsatisfiedLink exceptions, make sure you check the definition of the DB2\_JDBC\_DRIVER\_PATH environment variable on both nodes. Make sure the definitions are correctly pointing to the path where the db2java.zip file exists on the machines. Next do a Full Resynchronize on both nodes. From a command window, not the console, stop and restart both node agents and also server1 and server2. That should fix it. It appears that the restart node agent button, does not restart the node agents in the same way as doing it from the command line.

# Exercise instructions

## Preface

To do this lab you must have completed the Federate Cell exercise as it sets up the environment of the nodes, nodeagents and servers which will be clustered in this exercise.

## Step 1 Check nodes and node agents

Before you can begin creating the cluster make sure both nodes agents are running and the nodes are synchronized.

- \_\_\_ 1. Log in to the Deployment Manager's console.
- \_\_\_ 2. Make sure that both federated nodes, **was6host00Node01** and **was6host00Node02** are up, running and synchronized.
- \_\_\_ a. Select **System Administration —> Node agents.**

### **Node agents**

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**

| <input type="button" value="Stop"/> <input type="button" value="Restart"/> <input type="button" value="Restart all Servers on Node"/> |                           |                  |           |        |
|---------------------------------------------------------------------------------------------------------------------------------------|---------------------------|------------------|-----------|--------|
| Select                                                                                                                                | Name ▲                    | Node ▲           | Version ▲ | Status |
| <input type="checkbox"/>                                                                                                              | <a href="#">nodeagent</a> | was6host00Node02 | 6.0.0.1   |        |
| <input type="checkbox"/>                                                                                                              | <a href="#">nodeagent</a> | was6host00Node01 | 6.0.0.1   |        |
| Total 2                                                                                                                               |                           |                  |           |        |

- \_\_ b. Select **System Administration** → **Nodes**.

#### **Nodes**

A node corresponds to a physical computer system with a distinct IP host address. The node name is usually the same as the host name for the computer. The following table lists the nodes in this cell. You can add new nodes into the cell by clicking on "Add Node" and specifying a remote, running WebSphere Application Server instance.

#### Preferences

| Add Node                            | Remove Node                             | Force Delete             | Synchronize              | Full Resynchronize       | Stop                     |
|-------------------------------------|-----------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| <input checked="" type="checkbox"/> | <input type="checkbox"/>                | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Select                              | Name ▲                                  | Version ▲                | Discovery Protocol ▲     | Status ▲                 |                          |
| <input type="checkbox"/>            | <a href="#">ihsnode</a>                 | 6.0.0.1                  | TCP                      |                          |                          |
| <input type="checkbox"/>            | <a href="#">was6host00CellManager01</a> | 6.0.0.1                  | TCP                      |                          |                          |
| <input type="checkbox"/>            | <a href="#">was6host00Node01</a>        | 6.0.0.1                  | TCP                      |                          |                          |
| <input type="checkbox"/>            | <a href="#">was6host00Node02</a>        | 6.0.0.1                  | TCP                      |                          |                          |
| Total 4                             |                                         |                          |                          |                          |                          |

**Information:** If the node agent needs to be started, use the **startNode** script from a command window to start them. Make sure you are in the **bin** folder for the proper profile you are trying to start.

## Step 2 Creating the TradeCluster cluster

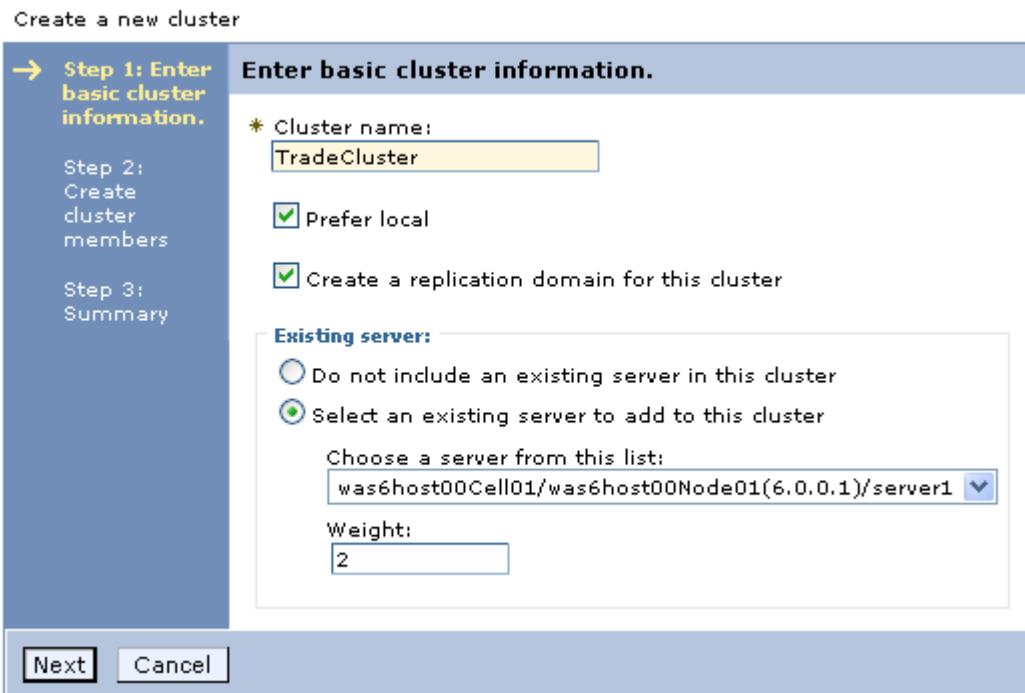
In this step you create the cluster that will contain the cluster members that will participate in workload management of the Trade application.

A cluster is composed of two or more servers in a cell which are assigned to run the same application.

Clusters are logical constructs which are equivalent to servers.

- \_\_ 1. Create a new cluster called **TradeCluster**.
  - \_\_ a. Select **Servers** → **Clusters**.
  - \_\_ b. Click **New**.
  - \_\_ c. Enter **TradeCluster** for the Cluster name,
  - \_\_ d. Select **Prefer local** and **Create a replication domain for this cluster**,

- \_\_\_ e. Use an existing server. Select **Select an existing server to add to this cluster** and choose server1 on node01..



- \_\_\_ f. Click **Next**.
- \_\_\_ 2. Add a new server called **server2** to the cluster, this server should be created in node **was6host00Node02**.
- \_\_\_ a. Enter **server2** for the **Member name**. This is the name of a new server which is created in the selected node, in this case **was6host00Node02**. This node was previously created by a custom profile during the previous exercise.
- \_\_\_ b. Select **was6host00Node02** for the node name.

\_\_ c. Select **Generate unique Http Ports.**

Create a new cluster

**Create cluster members**

Enter information about this new cluster member, and click Apply to add this cluster member to the member list. Use the Edit function to edit the properties of a cluster member that are already included in this list. Use the Delete function to remove a cluster member from this list.

\* Member name

Select node

Weight

Generate Unique Http Ports

**Apply**

| Select                   | Application servers | Nodes            | Version | Weight |
|--------------------------|---------------------|------------------|---------|--------|
| <input type="checkbox"/> | server1             | was6host00Node01 | 6.0.0.1 | 2      |

**Previous** **Next** **Cancel**

**Information:** Notice 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 displayed here.

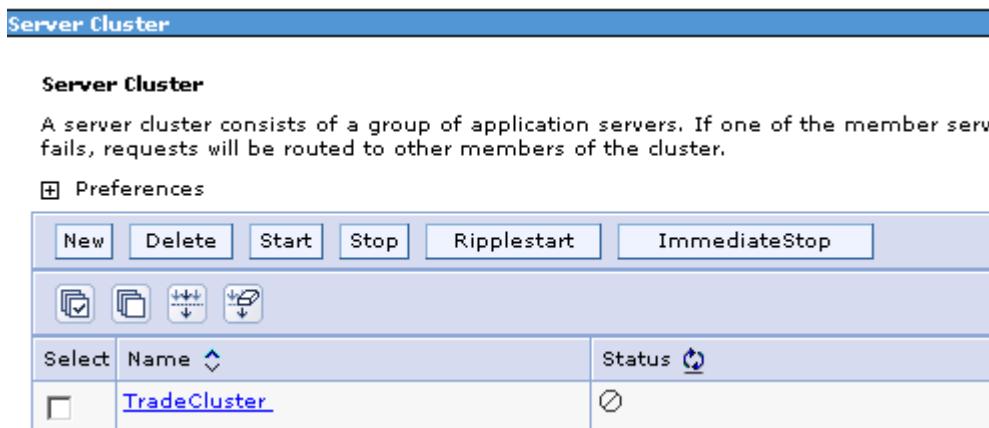
\_\_ d. Click **Apply**. At the bottom of the page, make sure **server2** has been added.

| <b>Edit</b> <b>Delete</b>           |                          |                  |         |        |
|-------------------------------------|--------------------------|------------------|---------|--------|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> |                  |         |        |
| Select                              | Application servers      | Nodes            | Version | Weight |
| <input type="checkbox"/>            | server1                  | was6host00Node01 | 6.0.0.1 | 2      |
| <input type="checkbox"/>            | server2                  | was6host00Node02 | 6.0.0.1 | 2      |

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

\_\_ f. Click **Finish**.

- \_\_\_ g. Save your changes and synchronize with the nodes.
- \_\_\_ 3. Verify the new cluster has been added to the server configuration. Select **Servers** → **Clusters**. The **TradeCluster** cluster should be displayed on the page.



### Step 3 Set the applications to run on the cluster

Now that the cluster has been 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 needs to be mapped to the applications.

- \_\_\_ 1. For both the existing applications, **TradeApplication** and **QuoteWS**, perform the next series of steps to map their modules to the **TradeApp** cluster and the Web server.
- \_\_\_ a. Select **Applications** → **Enterprise Applications**.

#### Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

##### Preferences

| Start                    | Stop                             | Install | Uninstall | Update | Rollout Update | Remove File | Export | Export DDL |
|--------------------------|----------------------------------|---------|-----------|--------|----------------|-------------|--------|------------|
| Select                   | Name                             |         |           | Status |                |             |        |            |
| <input type="checkbox"/> | <a href="#">QuoteWS</a>          |         |           |        |                |             |        |            |
| <input type="checkbox"/> | <a href="#">TradeApplication</a> |         |           |        |                |             |        |            |
| Total 3                  |                                  |         |           |        |                |             |        |            |

- \_\_\_ b. Click the name of one of the applications.
- \_\_\_ c. Under **Additional Properties** click **Map modules to servers**.
- \_\_\_ d. Select all the modules of the application.

- \_\_\_ e. In the **Clusters and Servers** list select both the **TradeCluster** cluster and the **webserver00** Web servers (use the Ctrl key to select multiple servers).

[Enterprise Applications > QuoteWS > Selecting servers](#)

Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules 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 will 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 which are routed through it.

Clusters and Servers:

WebSphere:cell=was6host00Cell01,cluster=TradeCluster  
WebSphere:cell=was6host00Cell01,node=ihsnode,server=webserver00

| Select                              | Module        | URI                               | Server |
|-------------------------------------|---------------|-----------------------------------|--------|
| <input checked="" type="checkbox"/> | QuoteEJB      | QuoteEJB.jar,META-INF/ejb-jar.xml |        |
| <input checked="" type="checkbox"/> | QuoteWSRouter | QuoteWSRouter.war,WEB-INF/web.xml |        |

- \_\_\_ f. Click **Apply**.  
 \_\_\_ g. Click **OK**.  
 \_\_\_ h. Make sure that the modules were mapped to both the **TradeCluster** and the Web server.

Clusters and Servers:

WebSphere:cell=was6host00Cell01,cluster=TradeCluster  
WebSphere:cell=was6host00Cell01,node=ihsnode,server=webserver00

| Select                   | Module        | URI                               | Server                                                                                                                  |
|--------------------------|---------------|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| <input type="checkbox"/> | QuoteEJB      | QuoteEJB.jar,META-INF/ejb-jar.xml | WebSphere:cell=was6host00Cell01,cluster=TradeCluster<br>WebSphere:cell=was6host00Cell01,node=ihsnode,server=webserver00 |
| <input type="checkbox"/> | QuoteWSRouter | QuoteWSRouter.war,WEB-INF/web.xml | WebSphere:cell=was6host00Cell01,cluster=TradeCluster<br>WebSphere:cell=was6host00Cell01,node=ihsnode,server=webserver00 |

- \_\_\_ i. Save the configuration changes  
 \_\_\_ j. Repeat the steps above for the next application.  
 \_\_\_ 2. Regenerate and propagate the plug-in configuration file. (This should happen automatically, do it manually if it didn't)

## Step 4 Test the application

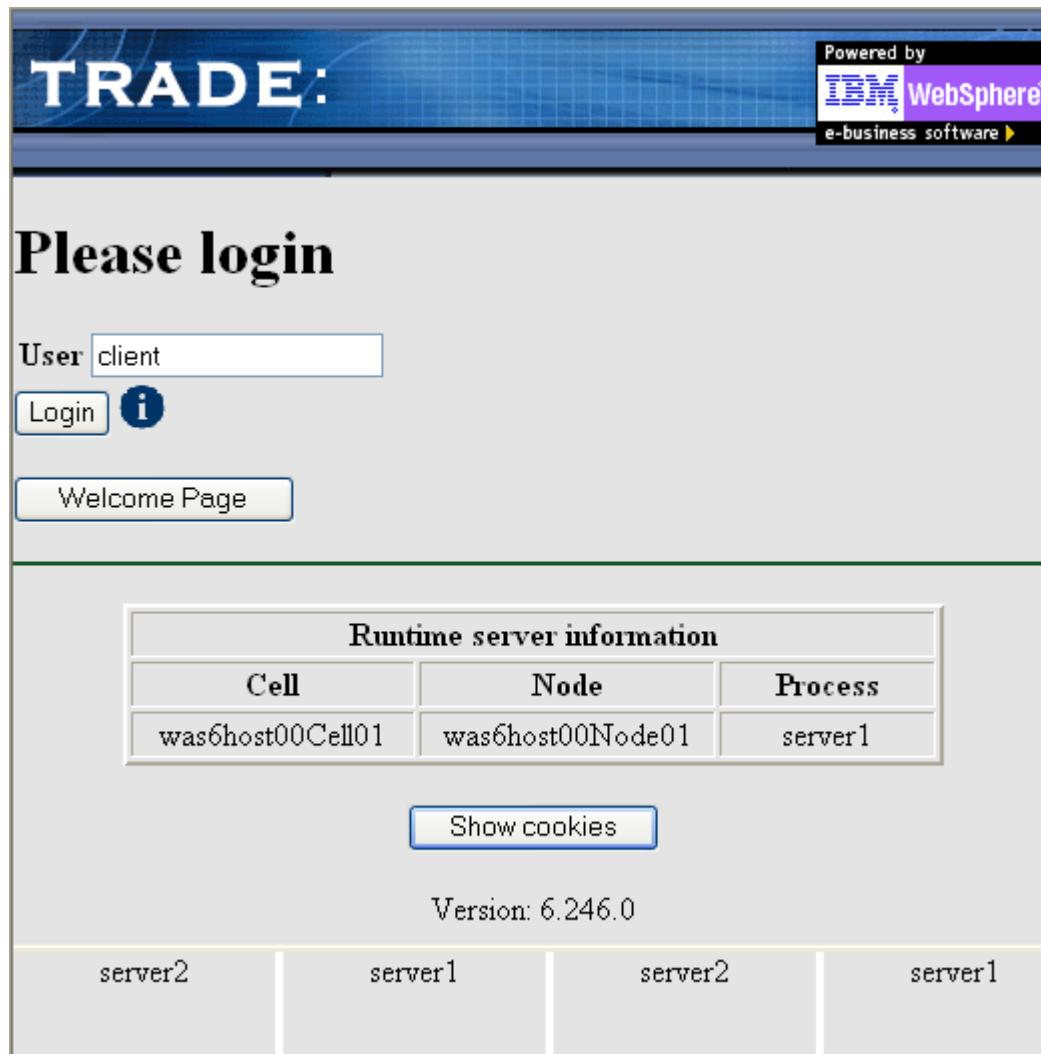
In this section of the exercise the application is tested in a clustered environment. The application is served from both application servers in the cluster until affinity is established

by the creation of an Http session object. From then on, all requests are directed to the same application server which created the session.

If the server that created the session is not available the Web server plug-in will re-route the request to another server on the cluster, creating a new affinity to that server.

Going to the new server will work to serve content that does not use information in the Http session. At this stage of the exercise, attempting to use session information will fail. This “problem” is fixed later in this exercise, where session replication is configured.

- \_\_\_ 1. Make sure all node agents, servers and applications are up and running.
- \_\_\_ 2. Make sure the Web server is running.
- \_\_\_ 3. Close all existing Web browser windows.
- \_\_\_ 4. Open a new Web browser window.
- \_\_\_ 5. Access the Trade application through address:  
`http://localhost/Trade/web/welcome.html`
- \_\_\_ 6. Click **Login**.



Note that the Runtime server information indicates the page was served from **server1**, which is the server in was6host00Node01. Also note, on the bottom of the screen, that subsequent requests were served from both **server1** and **server2**. The bottom of the screen is a frameset which calls a servlet to provide runtime information, requests to this servlet were workload managed between the two available servers in the cluster.

- \_\_\_ 7. Click **Refresh** on the Web browser a few times and see that workload management continues to occur.
- \_\_\_ 8. Click **Show cookies**. Notice that at this time there are no cookies, as login to the application has not yet happened. A session is established once a user logs in.



- \_\_\_ 9. Use **client** for the **User** name and log in to the application.
- \_\_\_ 10. Notice the Runtime server information, which server served the request? \_\_\_\_\_
- \_\_\_ 11. Click **Show cookies** again. This time you should see the ClonelD of the server that served the request.



**Information:** This ClonelD can be found in the plug-in-config.xml file used by the Web browser plug-in.

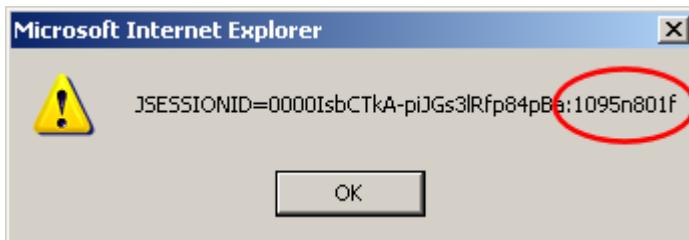
- \_\_\_ 12. Continue testing the application, all the requests should now be served by the same server.
- \_\_\_ 13. After you are satisfied that affinity has been established, use the WebSphere administrative console to stop the server which has been serving your requests. This will force failover to the other server in the cluster.

- \_\_\_ 14. Without logging out of the application, navigate to another page of the application. This should result in an error as distributed session management has not yet been set.



Note that the server, which served this request, is the “other” server in the cluster, **server2**. Since session management has not yet been configured, server2 cannot obtain server1’s session information and fails to find the account of the logged in user, which is stored in the session.

- \_\_\_ 15. Click **Show cookies** again and notice that the ClonelID belongs to server2.



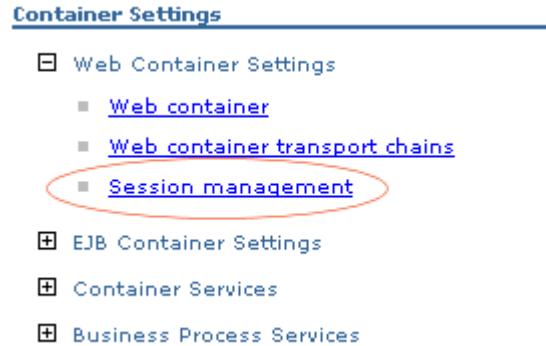
## Step 5 Configure Distributed Session Management

In order for session information to be shared between members of a cluster, a strategy to share session data must be put in place. WebSphere Application Server provides a variety of mechanisms to achieve this goal. The main strategies are Database and

memory-to-memory replication. Setting up either of these is very straightforward. In this exercise memory-to-memory replication is setup to handle session data replication.

Session management must be set on each of the servers in the cluster. Perform the following steps first on **server1** and then on **server2**.

- \_\_\_ 1. Select **Servers** —> **Application servers**.
- \_\_\_ 2. Click the hyperlink for either one of the servers.
- \_\_\_ 3. Under **Container Settings** expand **Web Container Settings**. Click **Session management**.



- \_\_\_ 4. Under **Additional properties**, click **Distributed environment settings**.
- \_\_\_ 5. Click **Memory-to-memory replication**. This will switch you to the next page.
- \_\_\_ 6. Select the **TradeCluster** Replication domain and set the Replication mode to **Both client and server**.

**Application servers > server1 > Memory-to-memory replication**

Configure memory-to-memory replication for failure recovery.

Configuration

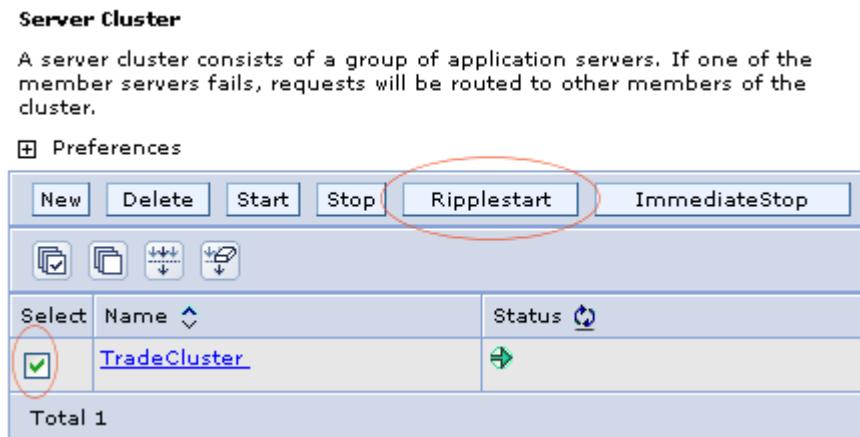
**General Properties**

- \* Replication domain
- \* Replication mode

Apply   OK   Reset   Cancel

- \_\_\_ 7. Click **OK**.
- \_\_\_ 8. Save the changes. Repeat these steps for the next server.
- \_\_\_ 9. Restart the cluster.

- \_\_\_ a. Select **Servers —> Clusters**.
- \_\_\_ b. Select the check box for the **TradeCluster**.
- \_\_\_ c. Click **Ripplestart**. Wait till all servers are cycled.



**Information:** Ripplestart will stop and start one server of the cluster at a time. Once the first server has been cycled, the next server is stopped and started. This way you are assured only one server is down at any one time. It also does not overload the system by cycling multiple servers at the same time.

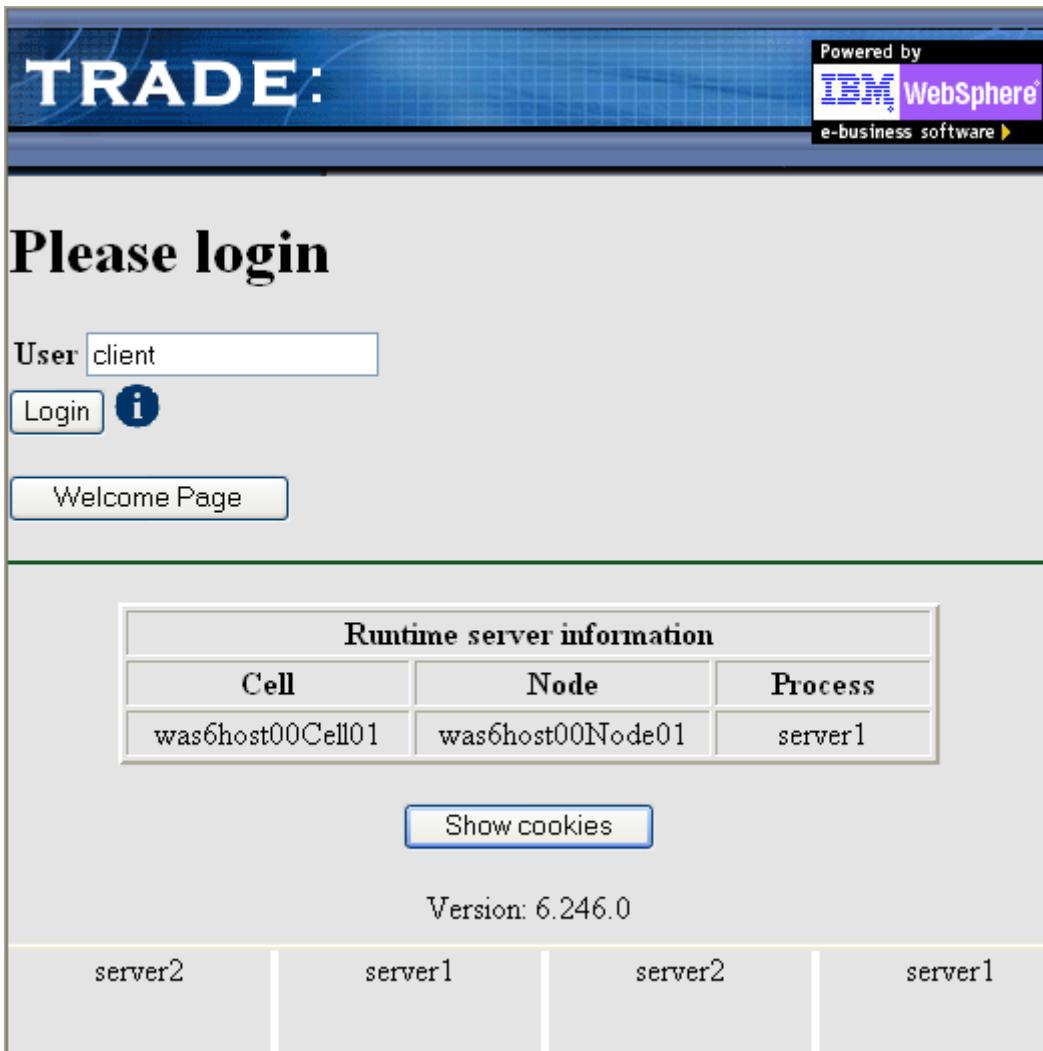
## Step 6 Test the application (again)

In this section of the exercise the application is tested again. However, now that session management has been configured, stopping one server should correctly fail over to the other server in the cluster without losing the session information.

- \_\_\_ 1. Make sure all node agents, servers and applications are up and running.
- \_\_\_ 2. Make sure the Web server is running.
- \_\_\_ 3. Close all existing Web browser windows.
- \_\_\_ 4. **Open a new Web browser window.** This will make sure that the old cookies have been flushed.
- \_\_\_ 5. Access the Trade application through address:

`http://localhost/Trade/web/welcome.html`

- \_\_\_ 6. Click **Login**.



As before, note that the Runtime server information indicates the page was served from **server1**, which is the server in was6host00Node01. Also note, on the bottom of the screen, that subsequent requests were served from both **server1** and **server2**.

**Information:** The bottom of the screen is a frameset which calls a servlet to provide runtime information, requests to this servlet were workload managed between the two available servers in the cluster.

- \_\_\_ 7. Click **Refresh** on the Web browser a few times and see that workload management continues to occur.

- \_\_\_ 8. Click **Show cookies**. Notice that at this time there are no cookies, as login to the application has not yet happened. A session is established once a user logs in.



- \_\_\_ 9. Use **client** for the **User** name and log in to the application.  
 \_\_\_ 10. Notice the Runtime server information, which server served the request? \_\_\_\_\_  
 \_\_\_ 11. Click **Show cookies** again. This time you should see the ClonelID of the server that served the request.



**Information:** The ClonelID is different now that memory-to-memory replication is configured. This ClonelID cannot be related back to the ClonelID on the plugin-config.xml file. It has been “encoded” in such away that the WebSphere plug-in module can determine which server it will send the request to and also which server is the backup server containing in its memory the session data.

- \_\_\_ 12. Continue testing the application, all the requests should now be served by the same server.  
 \_\_\_ 13. After you are satisfied that affinity has been established, use the WebSphere administrative console to stop the server which has been serving your requests. This will force failover to the other server in the cluster.

14. Without logging out of the application, navigate to another page of the application.

The screenshot shows the TRADE: application interface. At the top right, it says "Powered by IBM WebSphere e-business software Version: 6.246.0". Below the header is a navigation bar with links: Welcome, Login, Account Information, Holdings, and Transaction History. The main content area displays a table of transaction history:

| Trade Date      | Holding | Transaction Price | Shares | Transaction Type | Status  |
|-----------------|---------|-------------------|--------|------------------|---------|
| 2/13/03 5:18 PM | CSCO    | \$13.43           | 5.0    | SELL             | SUCCESS |
| 12/2/04 5:13 PM | PG      | \$55.82           | 1.0    | BUY              | SUCCESS |
| 12/2/04 5:15 PM | YHOO    | \$39.14           | 2.0    | BUY              | SUCCESS |
| 12/8/04 2:44 PM | IBM     | \$75.55           | 4.0    | BUY              | SUCCESS |

Below the table is a section titled "Runtime server information" containing a table:

| Cell             | Node             | Process |
|------------------|------------------|---------|
| was6host00Cell01 | was6host00Node02 | server2 |

A "Show cookies" button is located at the bottom of this section.

Note that the server which served this request is the “other” server in the cluster, **server2**. Also, failover was successful and the session was not lost. The request was routed to **server2** which had the session data replicated from **server1** by the memory-to-memory session data replication.

**Information:** At this time trying to get a quote from the local database, or buying a stock with a symbol that is not present in the local database table (QUOTE), may or may not work depending on which server is running. For more details and the instructions to fix this, see the Optional exercise at the end.

## End of lab

## Optional exercise: Update Trade Application using Fine Grained Application Update

As mentioned above, trying to get a quote from the database (Internet quotes disabled), or trying to buy stock using a symbol not currently in the local database, might fail at this point. Both of these operations involve dealing with the QUOTE database table and use the QuoteWS application that contains the Web service provider.

The Web service proxy is found using a URL (or Web address). This URL is hardcoded in the Web services client. To get around this problem with hardcoded URLs, the string containing the URL has been externalized in a properties file.

The hardcoded URL uses port 9080. Depending on which application server is running that port may or may not be available. Since the application is now running in a clustered environment, removing the port number from the URL and letting the Web server plug-in workload manager request will fix this problem, and at the same time add failover to the QuoteWS application.

In this part of the exercise you use Fine Grained Application Update to replace the properties file in the Trade application that contains the URL to the Web services proxy.

- \_\_\_ 1. Update the Trade Application to remove the port number from the URL pointing to the Web services provider. The URL is the **QuoteWSProps.properties** file.
  - \_\_\_ a. Edit the **QuoteWSProps.properties** file in **<software\_dir>\WLM**. Change the address in the URL to the IP address of the Web server. Do not enter a port number. Make sure to **save** the changes.



- \_\_\_ 2. Update the Trade Application with the new properties file.
  - \_\_\_ a. Using the WebSphere administrative console, go to **Applications** —> **Enterprise Applications**, select the **TradeApplication** and click **Update**.
  - \_\_\_ b. Select **Single file**.
 

**Information:** At this point there are four options to choose from. Perhaps the most common is Full application. In this case only a Single file will be updated. It is also possible to update a single module or a partial application.
  - \_\_\_ c. Use **TradeEJB.jar\com\ibm\trade\QuoteWSProps.properties** for the **Relative path to file**. This specifies which file inside of the application is changed. Choose **Local file system** for the location of the file that is used to update the

application. Browse to the **QuoteWSProps.properties** file in the **<software\_dir>\WLM** folder.

Single file

Select this option to update an existing file or to add a new file to the application. If the relative path to the file matches an existing path to a file in the installed application, the uploaded file replaces the existing file. If the relative path to the file does not exist in the installed application, the uploaded file is added to the application.

Relative path to file.

`bm\ibm\trade\QuoteWSProps.properties`

Path to the existing file, or to the desired path for the new file.

**Upload the new or replacement files.**

Local file system

Specify path

`QuoteWSProps.properties`

Remote file system

Specify path

\_\_\_ d. Click **Next**.

\_\_\_ e. A confirmation box appears. Click **OK**.

**Updating Application**

The following components will be updated. During application update, if the specified file does not exist, it otherwise the current file will be replaced.

`TradeEJB.jar\com\ibm\trade\QuoteWSProps.properties`

\_\_\_ f. At this point the Trade Application is updated. **Save** the changes to the master configuration and make sure the **Synchronize changes with Nodes** check box is selected.

\_\_\_ 3. Test the application

- \_\_\_ a. To make sure the application works, disable Internet quotes and try to obtain a quote from the database. Note that only symbols in the database will return a value, other values will cause the application to fail. Looking at the logs will reveal a `SymbolNotFound` exception.
- \_\_\_ b. Enable Internet quotes and buy stock using a symbol not in the database, the transaction completes successfully and the symbol is added to the database with the current purchase price.

## 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 were configured to run on the cluster by assigning the modules of the applications to the Web server and the cluster.

Lastly the application was thoroughly tested in the clustered environment and failover scenarios were created by stopping one of the servers.

To make failover to work when session data is involved the Data Replication Service was configured to use memory-to-memory replication. Again the application was tested in a failover scenario.



# Exercise 12. Install Trade Listener Application

## What this exercise is about

This lab covers the installation of the third module of the Trade Application. The Listener module contains a message-driven bean (MDB) which listens on a JMS queue for transactions made by users of the Trade Application. For this module to work the WebSphere Application Server needs to be configured to handle asynchronous messaging.

## What you should be able to do

At the end of the lab, you should be able to configure a:

- Systems Integration Bus (SIBus)
- Messaging Engine
- JMS queue

## Introduction

This module, called the Listener, can process the buy and sell transactions made by users of the Trade Application (if they have chosen to broadcast their transactions). The MDB in the module listens to a queue where transaction messages from the Trade Application arrive. As these messages are received, they are displayed on a table. As messages are processed, they are removed from the table.

Aside from installing the Trade Listener module, which is very straightforward, this module deals with the complexities of setting up the SIBus, Messaging Engine (ME), and JMS queues.

## Required materials

To do this lab you require a properly set up computer with WebSphere Application Server V6 installed as well as the necessary startup and program files

## Instructor exercise overview

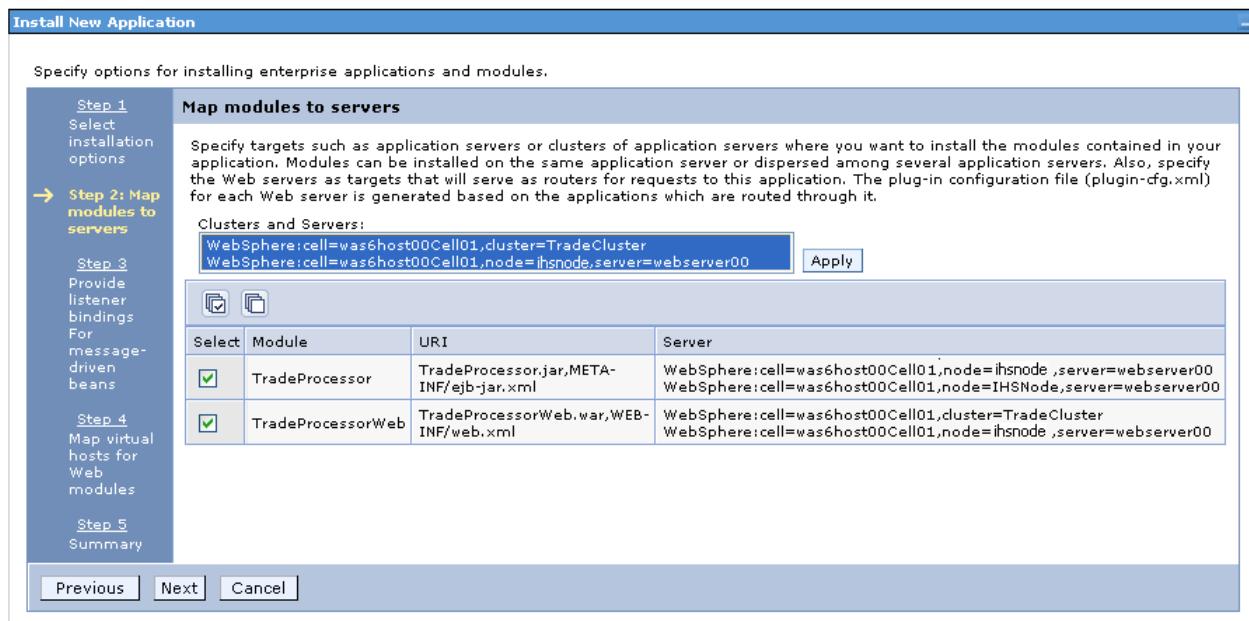
This lab deals mainly with the setup of the systems integration bus. There are lots of places for students to make mistakes because there is a lot of typing in dialog boxes. In addition to typographical errors, they may also forget to select the SIBus name to associate JMS resources with SIBus artifacts.

If you see the following error when testing the application, restart the Deployment Manager, node agents and cluster: com.ibm.websphere.sib.exception.SIResource Exception: CWSIT0019E: No suitable messaging engine is available in bus msgBus.

## Exercise instructions

### Step 1 Install the Trade Listener Application using the console

- 1. Using the WebSphere administrative console, install the **TradeListener.ear** enterprise application module.
  - a. Open a Web browser and connect to the WebSphere administrative console.
  - b. Log in to the console.
  - c. Select **Application** —> **Install New Application**
  - d. Click **Browse** and select  
 <software\_dir>\EARS and WARs\TradeListener.ear
  - e. Click **Next**.
  - f. Click **Next** again.
  - g. Click **Step 2: Map modules to servers**. Select both application modules. Select Both **TradeCluster** and **webserver00** from the **Cluster and Servers** list box. Click **Apply**.



- \_\_ h. Click the **Step 3: Provide listener bindings For message-driven beans** link.

The screenshot shows the 'Provide listener bindings For message-driven beans' step in the 'Install New Application' wizard. The left sidebar lists steps: Step 1, Step 2, Step 3 (highlighted in yellow), Step 4, and Step 5. The main panel has a table with the following data:

| Select                   | EJB module     | EJB               | URI                                     | Messaging Type | Bindings                                                                                                                                                                                                                                                                                                |
|--------------------------|----------------|-------------------|-----------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <input type="checkbox"/> | TradeProcessor | TradeProcessorEJB | TradeProcessor.jar,META-INF/ejb-jar.xml |                | <input type="radio"/> Listener port<br>Name: <input type="text"/><br><input checked="" type="radio"/> Activation Specification<br>JNDI name: <input type="text" value="ais/TradeLstnrsAS"/><br>Destination JNDI Name: <input type="text"/><br>ActivationSpec Authentication Alias: <input type="text"/> |

At the bottom are 'Previous', 'Next', and 'Cancel' buttons.

On this page you configure how MDBs in the application connect to the destination they are interested in listening. Listener ports can be used in WebSphere Application Server V6. New development projects should use a J2C activation specification. The JNDI name of the activation specification you see on this page was entered by the developer of the EJB module. It can be changed here if necessary.

Messages handled by message-driven beans have no client credentials associated with them. The messages are anonymous.

**Information:** With security enabled, making calls to secure enterprise beans from a message-driven bean requires that the message-driven bean be configured with a RunAs Identity deployment descriptor. Security depends on the role specified by the RunAs Identity for the message-driven bean as an EJB component. This identity is provided here as the J2C ActivationSpec Authentication Alias.

- \_\_ i. Click the **Summary** link on the page.
- \_\_ j. Click **Finish**.
- \_\_ 2. Save the changes to the master configuration.

- \_\_\_ 3. Verify that the application has been installed.

The screenshot shows the 'Enterprise Applications' interface. At the top, there's a toolbar with buttons for Start, Stop, Install, Uninstall, Update, Rollout Update, and Remove File. Below the toolbar is a toolbar with icons for selecting, adding, removing, and filtering applications. A table lists three applications: 'QuoteWS', 'Trade Order Processor', and 'TradeApplication'. Each row has a checkbox in the 'Select' column and a status icon in the 'Status' column. The 'Trade Order Processor' row is circled in red.

| Select                              | Name                                  | Status |
|-------------------------------------|---------------------------------------|--------|
| <input type="checkbox"/>            | <a href="#">QuoteWS</a>               | ∅      |
| <input checked="" type="checkbox"/> | <a href="#">Trade Order Processor</a> | ∅      |
| <input type="checkbox"/>            | <a href="#">TradeApplication</a>      | ∅      |

Total 3

- \_\_\_ 4. Make sure the servers of the TradeCluster are running and try to start the **Trade Order Processor** application

Notice that although the application installed fine, it cannot be started at this time. The reason being that the required setup for the MDB to work has not been done yet.

- \_\_\_ 5. Looking at the logs (SystemOut.log) you can see that the activation spec's JNDI cannot be found. This is just one of many setup steps you perform in the next parts of this exercise.

[12/7/04 13:35:25:341 EST] 00000031 EJBContainerI E WSVR0062E:  
Unable to start EJB, Trade Order  
Processor#TradeProcessor.jar#TradeProcessorEJB:  
javax.resource.ResourceException: Failed to lookup  
ActivationSpec.eis/TradeLstnrAS  
at  
com.ibm.ejs.j2c.RALifeCycleManagerImpl.activateEndpoint(RALifeCycleManagerImpl.java:1253)  
at  
com.ibm.ejs.container.MessageEndpointFactoryImpl.activateEndpoint(MessageEndpointFactoryImpl.java:237)  
at  
com.ibm.ejs.container.EJSContainer.loadBeanMetaData(EJSContainer.java:1611)  
at  
com.ibm.ejs.container.HomeOfHomes.loadBeanMetaData(HomeOfHomes.java:659)  
at  
com.ibm.ejs.container.HomeRecord.getHomeAndInitialize(HomeRecord.java:100)

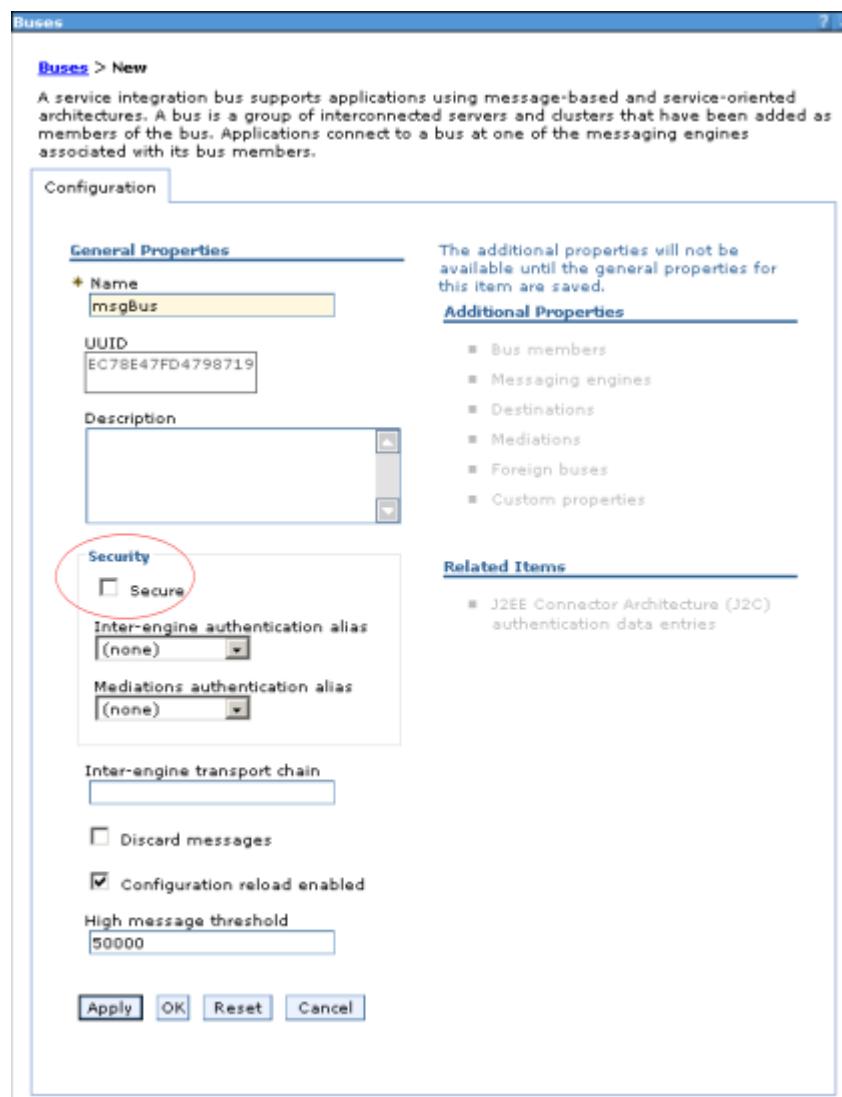
## ***Step 2 Setting up a Service Integration Bus***

Setting up a Service Integration Bus (SIBus) is the first step in creating the infrastructure required to participate in messaging. The SIBus is the main architectural component of WebSphere Platform Messaging.

A Service Integration Bus is made up of a group of interconnected servers and clusters that have been added as members of the bus. Applications connect to a SIBus at one of the messaging engines associated with its bus members.

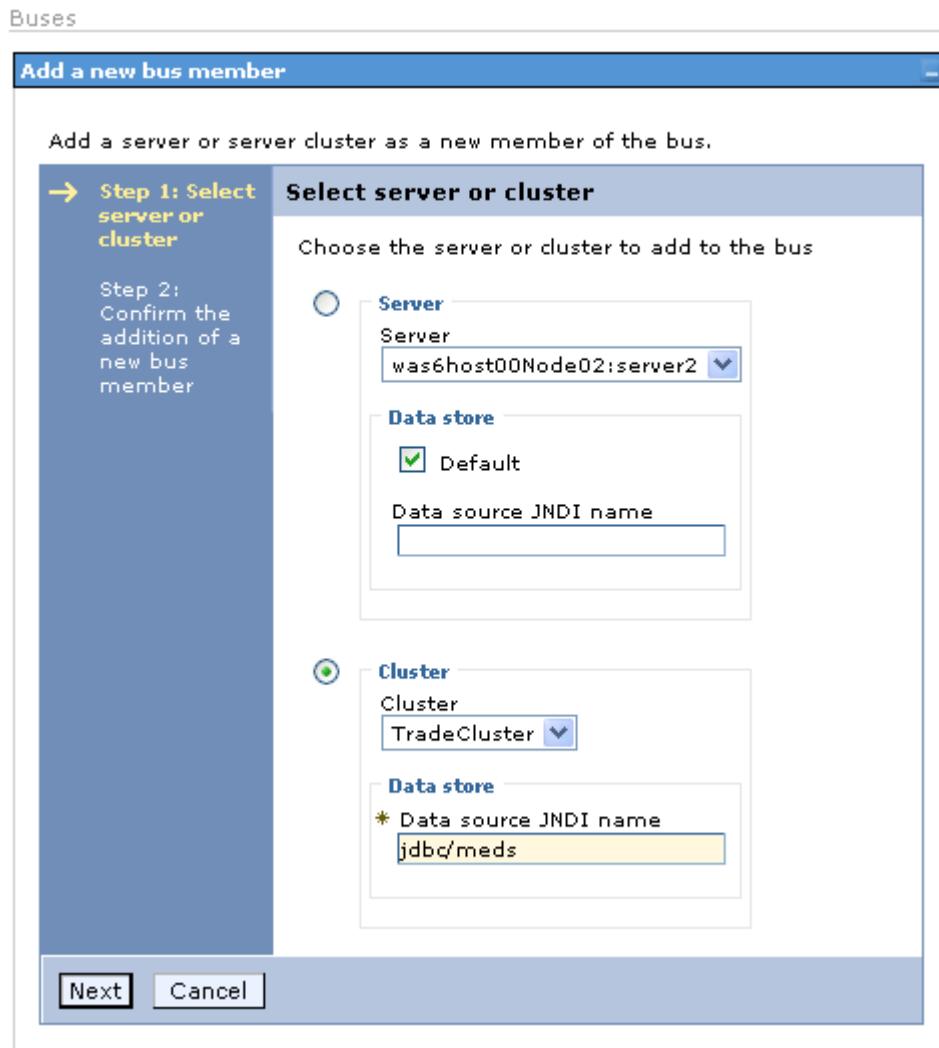
- \_\_\_ 1. Create a new bus named **msgBus**.
    - \_\_\_ a. Using the WebSphere administrative console select **Service Integration** —> **Buses**.
    - \_\_\_ b. Click **New**.
    - \_\_\_ c. Name the bus: **msgBus**.

- \_\_\_ d. Ensure the **Secure** check box is **not** selected.



- \_\_\_ e. Click **OK**.
- \_\_\_ f. Save the changes.
- \_\_\_ 2. Make TradeCluster a member of the bus.
- \_\_\_ a. Select **Service Integration** —> **Buses**.
- \_\_\_ b. Click **msgBus**.
- \_\_\_ c. Under **Additional Properties**, click **Bus Members**.
- \_\_\_ d. Click **Add**.

- \_\_\_ e. Since you are running the application in a cluster, select Cluster and select **TradeCluster** from the **Cluster** drop-down list. Provide the JNDI name of the data source for the messages as **jdbc/meds**.



**Information:** Creating a bus member automatically creates a messaging engine. Note that the JNDI name of the data source points to a non existing data source. It is created later in this exercise.

- \_\_\_ f. Click **Next**.
- \_\_\_ g. Click **Finish**.
- \_\_\_ h. Save your changes.
- \_\_\_ 3. Create a DB2 JDBC provider that will be used by data source for the messaging engine.
- \_\_\_ a. Select **Resources** —> **JDBC Providers**.
- \_\_\_ b. Make sure the **TradeCluster Cluster** scope is selected. If not, click **Cluster** and select **TradeCluster**.

- \_\_\_ c. Click **New**.
  - \_\_\_ d. Select the following values under **General properties**:  
**DB2** for the database type, **DB2 Legacy CLI-based Type 2 JDBC Driver** for the provider type and **XA data source** for the implementation type.
  - \_\_\_ e. Click **Next**.
  - \_\_\_ f. Take all defaults on the next page and click **OK**.
  - \_\_\_ g. Save the changes, make sure you synchronize with nodes.
4. Add a datasource for the messaging engine.
- \_\_\_ a. Click the just created DB2 JDBC provider.
  - \_\_\_ b. Click **Data sources** under Additional Properties.
  - \_\_\_ c. Click **New**.

- \_\_ d. Enter **Messaging Engine Datasource** for the Name, **jdbcm/meds** for the JNDI name.

The screenshot shows the 'JDBC providers' configuration interface. A message box at the top right says: 'Additional Properties for this object will not be available to edit until its general properties are applied by clicking on either Apply or OK.'

The navigation path is: JDBC providers > Trade > Data sources > New.

The 'General Properties' section contains:

- \* Scope: cells:was6host00Cell01:clusters:TradeCluster
- \* Name: DB2 Legacy CLI-based JDBC Driver XA DataSource (highlighted in yellow)
- JNDI name: jdbcm/meds
- Use this Data Source in container managed persistence (CMP)

To the right of the General Properties section, a note says: 'The additional properties will not be available until the general properties for this item are saved.' Below it is a 'Additional Properties' section with three items:

- Connection pool properties
- WebSphere Application Server data source properties
- Custom properties

The 'Description' section contains: New JDBC Datasource.

The 'Related Items' section includes a link to 'J2EE Connector Architecture (J2C) authentication data entries'.

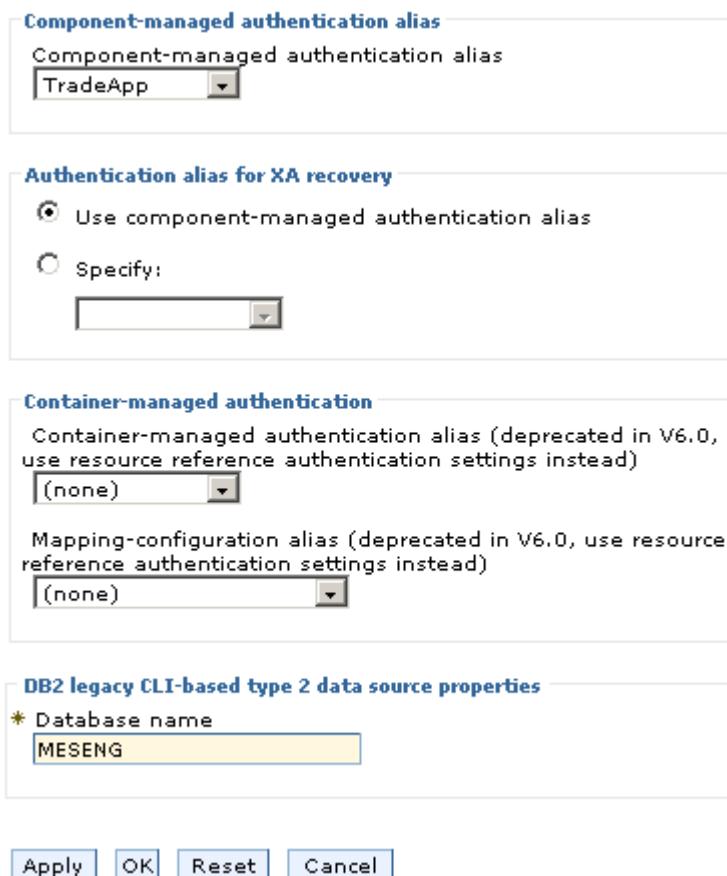
The 'Category' section has a single input field.

The 'Data store helper class name' section contains two options:
 

- Select a data store helper class: DB2 data store helper (com.ibm.websphere.rssadapter.DB2DataStoreHelper) (highlighted in blue)
- Specify a user-defined data store helper: Enter a package-qualified data store helper class name (with an empty input field)

- \_\_ e. Scroll the page and select the **TradeApp** for the **Component-managed authentication alias**.

- \_\_\_ f. Enter **MESENG** for the Database name.



**Information:** Note that the MESENG database must already exist. The Database Administrator should have been asked to create it. The tables for the database are created by WebSphere Application Server.

The database used by the messaging engine cannot be reused by other messaging engines. If the messaging engine using the database is deleted, the database must be deleted as well as it cannot be reused by a new messaging engine.

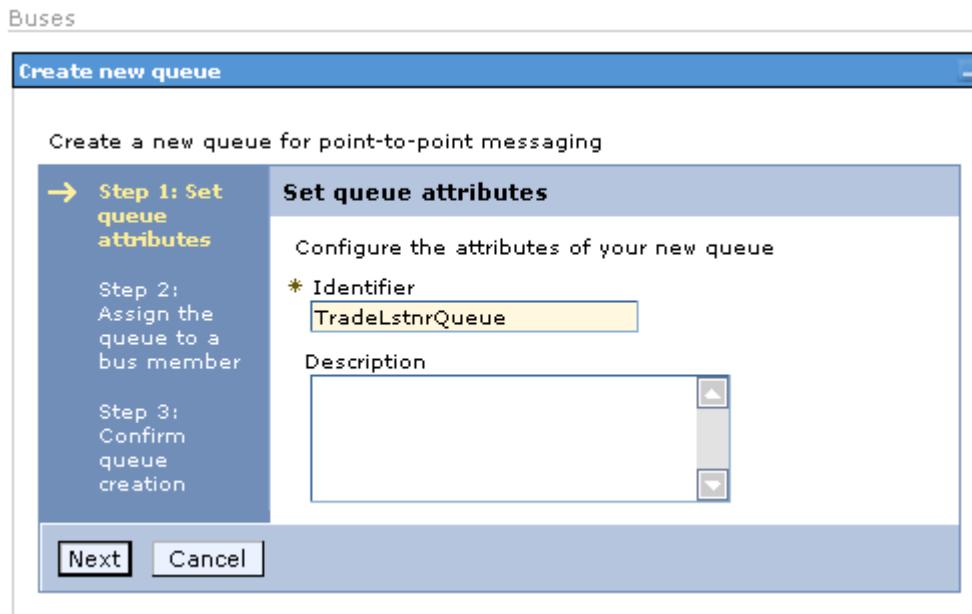
- \_\_\_ g. Click **OK**.  
\_\_\_ h. Save your changes.

**Information:** In production the connection pooling of this data source must be adjusted to have at least 50 Maximum connections (default is 10). Also the Purge Policy must be set to: EntirePool.

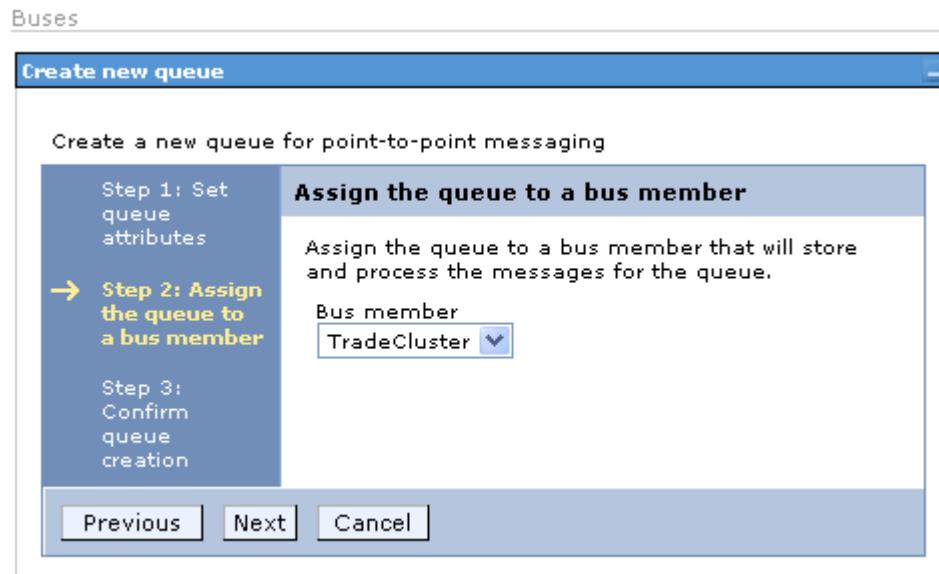
**Information:**

- \_\_\_ i. Test the connection.  
\_\_\_ 5. The Trade Listener application listens on a point-to-point destination, a queue. In this step you create a service integration bus destination named **TradeLstntrQueue**.  
\_\_\_ a. Select **Service Integration** —> **Buses**.  
\_\_\_ b. Click **msgBus**.

- \_\_\_ c. Click **Destinations**.
- \_\_\_ d. Click **New**
- \_\_\_ e. Select the **Queue** radio button.
- \_\_\_ f. Click **Next**.
- \_\_\_ g. Enter **TradeLstnrQueue** for the **Identifier**.



- \_\_\_ h. Click **Next**.
- \_\_\_ i. Assign the queue to the **TradeCluster** bus member.



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

- 
- \_\_\_ I. Save the configuration changes.

### **Step 3 Setting up a JMS destination**

So far you have set up both the Service Integration Bus and the Bus destination (in this case a queue). Both of these are infrastructure setup steps that must be set up before a JMS destination can be defined for the MDB to use.

Setting up the JMS resources requires two setup steps to create the:

- JMS Connection Factory
  - JMS Queue
- 
- \_\_\_ 1. Create a new JMS Connection Factory called **TradeLstnrJMSCF** and associate it with the SIBus you created earlier, **msgBus**.
    - \_\_\_ a. Select and expand **Resources** —> **JMS Providers**.
    - \_\_\_ b. Click **Default messaging**. Make sure the scope is **TradeCluster**.
    - \_\_\_ c. Under **Connection Factories**, click **JMS connection factory**. This is a generic connection factory that can later be programmatically used to create point-to-point queues, or publish/subscribe topic destinations.
    - \_\_\_ d. Click **New**.
    - \_\_\_ e. Enter **TradeLstnrJMSCF** for the Name and **jms/TradeLstnrJMSCF** for the JNDI name.

- \_\_\_ f. Select **msgBus** for the Bus name.

The screenshot shows the 'Configuration' interface with the 'General Properties' tab selected. In the 'Administration' section, the 'Scope' field contains 'cells:was6host00Cell01:clusters:TradeCluster'. The 'Name' field is set to 'TradeLstnrJMSCF' and the 'JNDI name' field is set to 'jms/TradeLstnrJMSCF'. The 'Description' and 'Category' fields are empty. In the 'Connection' section, the 'Bus name' dropdown is set to 'msgBus' and the 'Target' field is empty. To the right, a 'Related Items' sidebar lists 'J2EE Connector Architecture (J2C) authentication data entries' and 'Buses'.

- \_\_\_ g. Click **OK**.
- \_\_\_ h. Save the configuration changes.
2. Create a JMS Queue named **TradeLstnrJMSQueue**
- Select and expand **Resources** —> **JMS Providers**.
  - Click **Default Messaging**.
  - Under **Destinations**, click **JMS queue**.
  - Click **New**.
  - Enter **TradeLstnrJMSQueue** for the Name and **jms/TradeLstnrJMSQueue** for the JNDI name.
  - Select **msgBus** for the Bus name.
  - Select **TradeLstnrQueue** for the Queue name.

**Information:** This is the association between the JMS queue and the SIBus destination.

[Default messaging provider](#) > [JMS queue](#) > New

A JMS queue is used as a destination for point-to-point messaging. Use JMS queue administrative queues for the default messaging provider.

Configuration

**General Properties**

**Administration**

- \* Scope: cells:was6host00Cell01:clusters:TradeCluster
- \* Name: TradeLstnrJMSQueue
- \* JNDI name: jms/TradeLstnrJMSQueue

Description:

**Connection**

- Bus name: msgBus
- \* Queue name: TradeLstnrQueue
- Delivery mode: Application
- Time to live: 0 milliseconds
- Priority: 0

**Advanced**

- Read ahead: Enabled

- \_\_\_ h. Click **OK**.
- \_\_\_ i. Save the configuration changes.

#### **Step 4 Setting up the Activation Specification**

The activation specification is what a message-driven bean (MDB) uses to establish communication with the JMS destination. Activation Specs enable EJB 2.1 MDBs to use JCA 1.5 resource adapter to receive JMS (and non-JMS) messages. Although Listener

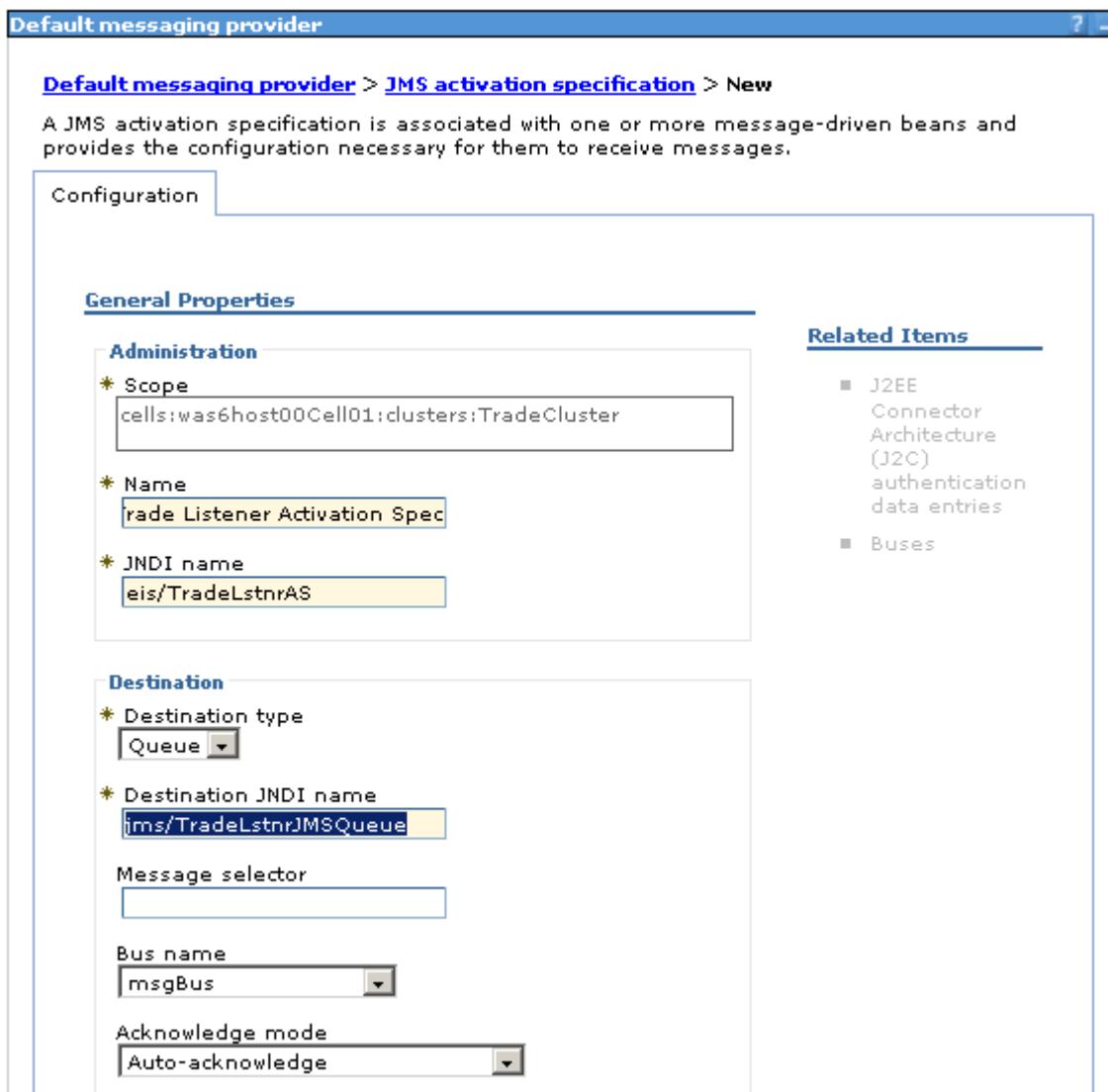
ports are still supported in WebSphere Application Server V6 it is recommended that new development use Activation Specs instead.

- \_\_\_ 1. Create the **TradeLstnrAS** Activation Spec for the MDB used in the application.
  - \_\_\_ a. Select and expand **Resources —> JMS Providers**.
  - \_\_\_ b. Click **Default messaging**.
  - \_\_\_ c. Under **Activation Specifications**, click **JMS activation specification**.
  - \_\_\_ d. Click **New**.
  - \_\_\_ e. Enter **Trade Listener Activation spec** for the Name.
  - \_\_\_ f. Enter **eis/TradeLstnrAS** for the JNDI name.

**Information:** This name matches the Activation spec defined on the MDB.

- \_\_\_ g. Select Destination type of **Queue**.
- \_\_\_ h. Enter **jms/TradeLstnrJMSQueue** for the Destination JNDI name.

- \_\_\_ i. Select **msgBus** for the Bus name.



- \_\_\_ 2. Click **OK**.  
\_\_\_ 3. Save the configuration changes.

### **Step 5 Test the Trade listener application**

Now that the SIBus and the JMS resources have been configured you are ready to test the Trade Listener application. As mentioned before this application works in concert with the other two modules, the Trade Application and the QuoteWS application.

- \_\_\_ 1. Start the Trade Listener application.  
\_\_\_ a. Select **Applications** → **Enterprise Applications**.  
\_\_\_ b. Select the check box by the **Trade Order Processor** application (the display name of the Trade Listener application).

- \_\_\_ c. Click **Start**. After a few moments the applications status icon should change to a green arrow indicating the application has started.

#### Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Preferences

| Select                   | Name                                  | Status |  |  |  |  |  |  |  |  |
|--------------------------|---------------------------------------|--------|--|--|--|--|--|--|--|--|
| <input type="checkbox"/> | <a href="#">QuoteWS</a>               |        |  |  |  |  |  |  |  |  |
| <input type="checkbox"/> | <a href="#">Trade Order Processor</a> |        |  |  |  |  |  |  |  |  |
| <input type="checkbox"/> | <a href="#">TradeApplication</a>      |        |  |  |  |  |  |  |  |  |

Total 3

**Information:** If the application does not start for you, make sure all the previous steps were completed correctly. You may wish to consult the log files to help you determine where the problem is. Call your instructor if you have problems.

- \_\_\_ 2. You need two browser windows opened at the same time to see the application working.
- \_\_\_ a. Open a Web browser to page:  
`http://localhost/Trade/web`
- \_\_\_ b. Log in to the Trade application as you did in previous labs.
- \_\_\_ c. On the **Account Information** page click **Update Account Page**.
- \_\_\_ d. Check **Broadcast transaction data**.
- \_\_\_ e. Click **Update Account**.
- \_\_\_ f. Proceed to the **Holdings** page.
- \_\_\_ g. Open another Web browser to page:

<http://localhost/TradeProcessorWeb/GetTradeJobs>

The screenshot shows a web page titled "TRADE" with a blue header. Below the header, the title "Trade Requests Received" is displayed in a large, bold, blue font. Underneath the title, there is a message "Refresh every: 20 seconds". Below this message is a horizontal table with five columns: "Account", "Buy/Sell", "Symbol", "Qty", and "Total cost". The table currently has no data rows.

This page refreshes every 10 seconds and lists any transactions which have not yet been processed.

- \_\_\_ h. Select the browser showing the **Holdings** page of the Trade application. Buy or sell some stock. Keep an eye on the Trade listener browser window.

The screenshot shows the same web page as before, but with a refresh interval of 5 seconds. The table now contains three data rows representing transactions:

| Account                 | Buy/Sell | Symbol | Qty | Total cost |
|-------------------------|----------|--------|-----|------------|
| 87652289 (Jessie James) | SELL     | IBM    | 3.0 | \$275.76   |
| 87652289 (Jessie James) | BUY      | GOOG   | 5.0 | \$887.75   |
| 87652289 (Jessie James) | SELL     | PG     | 1.0 | \$53.18    |

— 3.

**Information:** If you do not see any Trade Requests appear on the page, check that you have enabled the broadcast of transactions user preference. If it still does not work check and make sure that the Messaging Engine has started at: Service integration —> Buses —> msgBus —> Messaging engines. If the Messaging Engine won't start recheck your settings and then server logs.

- 4. Continue watching the Trade listener browser window and see how transactions are removed from the list as they are processed.

**Information:** Since this is a simulation, transactions are considered processed after 30 seconds of them being received.

### **End of lab**

## Exercise review and wrap-up

The first part of the exercise looked at creating a service integration bus, adding a bus member and creating a bus destination.

The next two parts created the JMS resources used by the application.

Finally you tested the Trade listener application by creating some transactions and verifying they appeared on the listener window. After the transactions were processed they are removed from the window.



# Exercise 13. Configuring SSL (optional)

## What this exercise is about

This lab secures the communications between a browser and the Web server by configuring the IBM Http Server to use https. It does this by creating a Keyring with a self signed key. You will then define a virtual host within the httpd.conf to use that keyring for https communications to the IBM Http Server.

The final step of the lab is to test connectivity to the IBM Http Server using https. This also tests connectivity through the Web server back to the WebSphere Application Server with https (using the existing DummyKeyring).

## What you should be able to do

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

- Create a Keyring
- Generate a self signed key
- Configure IBM Http Server to load and use https
- Define port 443 on WebSphere Application Server's virtual host

## Required materials

This lab requires a properly setup computer with WebSphere Application Server V6 installed as well as the necessary startup and program files.

## Instructor exercise overview

This lab starts off with creating a keyring and self-signed key using ikeyman. The keyring will be encrypted using a password, which will in turn be stored in an encoded stash file.

The second part of the lab configures IHS for a new virtual host which accepts https connections. This virtual host will be defined through editing the IHS's httpd.conf. An example of a correctly configured httpd.conf exists in the <software\_dir>\SSL directory (for both Windows and Linux).

The final step of the lab is to test the http connection to both the IHS as well as all the way back to the Application Server.

**NOTE:** this lab does not cover configuring the plug-in to the Application Server. Since WebSphere 6 includes a preconfigured https connection using the dummykeyring, the https connection should work out of the box. Stress to the students that in a real world environment that they would need to replace the dummykeyring.

# Exercise instructions

## Preface

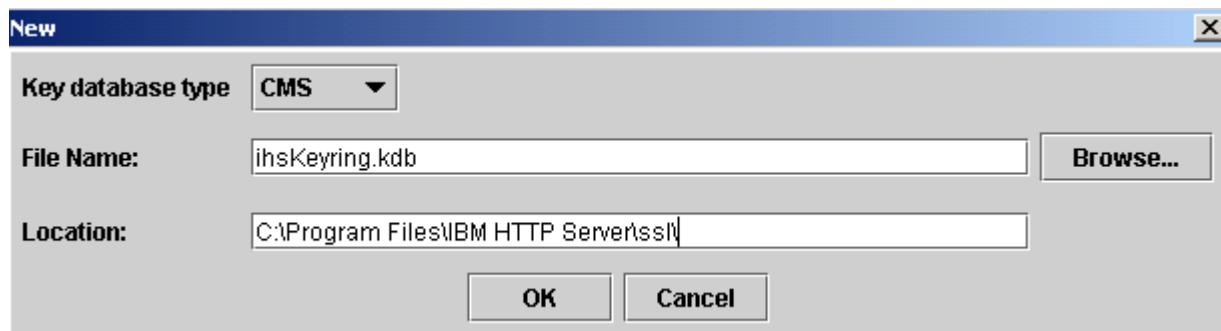
### Step 1 Creating a Keyring and a certificate

This section of the lab steps through the process of creating a certificate and a Keyring. These will then be used to configure SSL on the connection between the client browser and the Web server.

- \_\_\_ 1. Create a new directory to hold the keyring.
  - \_\_\_ a. Create the directory <ihs\_root>\ssl.
- \_\_\_ 2. Open Ikeyman in <ihs\_root>\bin
  - \_\_\_ a. Bring up IHS's Ikeyman. This can be done through either a command line window or using **Start —> Programs —> IBM HTTP Server 6.0 —> Start Key Management Utility.**
  - \_\_\_ b. The command line can be executed from the <ihs\_root>\bin directory:  
ikeyman.bat

**UNIX:** The UNIX command would be ./ikeyman.sh

- \_\_\_ 3. Create a Keyring
  - \_\_\_ a. Create a new Keyring by selecting **Key Database File —> New**
  - \_\_\_ b. Use type **CMS**, file name **ihsKeyring.kdb** and a location of <ihs\_root>\ssl. Click **OK**.



- \_\_\_ c. When prompted for a password for the Keyring, enter and confirm **was1edu** as the password. If desired, modify the **expiration time**. Check the “**Stash the password to a file**” check box.



**Information:** The stash file will be created containing an encoded form of the password. This encoding prevents casual viewing of the password, but is not highly secure. Therefore, you should protect this file by using operating system file permissions to prevent all access from unauthorized principals

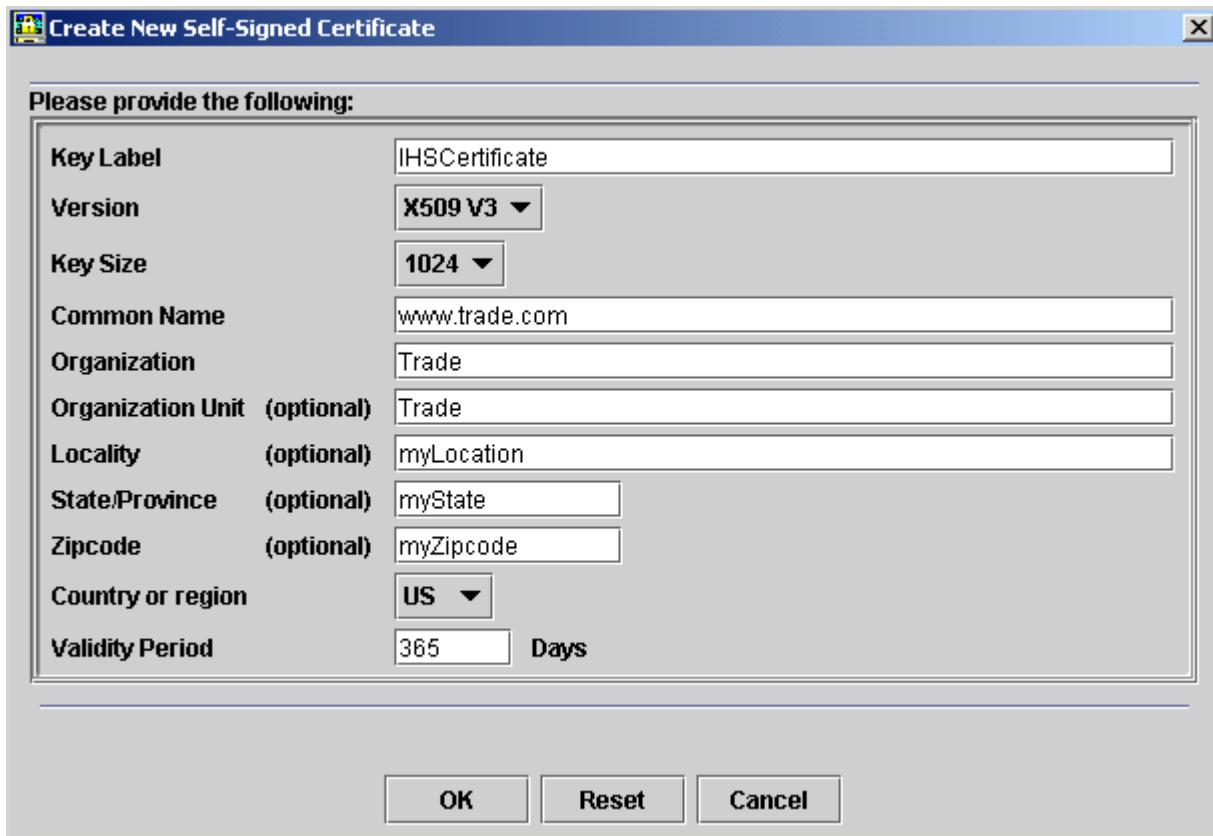
- \_\_\_ d. Click **OK**, and **OK** again for the informational box.

**Information:** The file name of the stash file will be the same as the name of the key file only it will have an .sth suffix. The stash file gets stored in the same directory as the key file.

- \_\_\_ 4. Create a new self-signed certificate.  
\_\_\_ a. In iKeyman, select **Create —> New Self-Signed Certificate** and enter::

| Example           | Description    |
|-------------------|----------------|
| Key Label         | IHSCertificate |
| Common Name       | www.trade.com  |
| Organization      | Trade          |
| Organization Unit | Trade          |
| Locality          | myLocation     |
| State/Province    | myState        |
| Zipcode           | myZipcode      |

- \_\_\_ b. Accept the defaults for the version, Key Size and Validity Period.



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

**Information:** This 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 could use it. To do this, click **Extract Certificate**, enter a **File Name** and **Location**. Click **OK**.

- \_\_\_ d. **Exit** ikeyman.  
\_\_\_ e. Get a directory listing of <ihs\_root>\ssl\ and verify that the following files were created:

```
ihsKeyring.kdb
ihsKeyring.sth
ihsKeyring.crl
ihsKeyring.rdb
```

## Step 2 Configure a Virtual Host on IHS for HTTPS

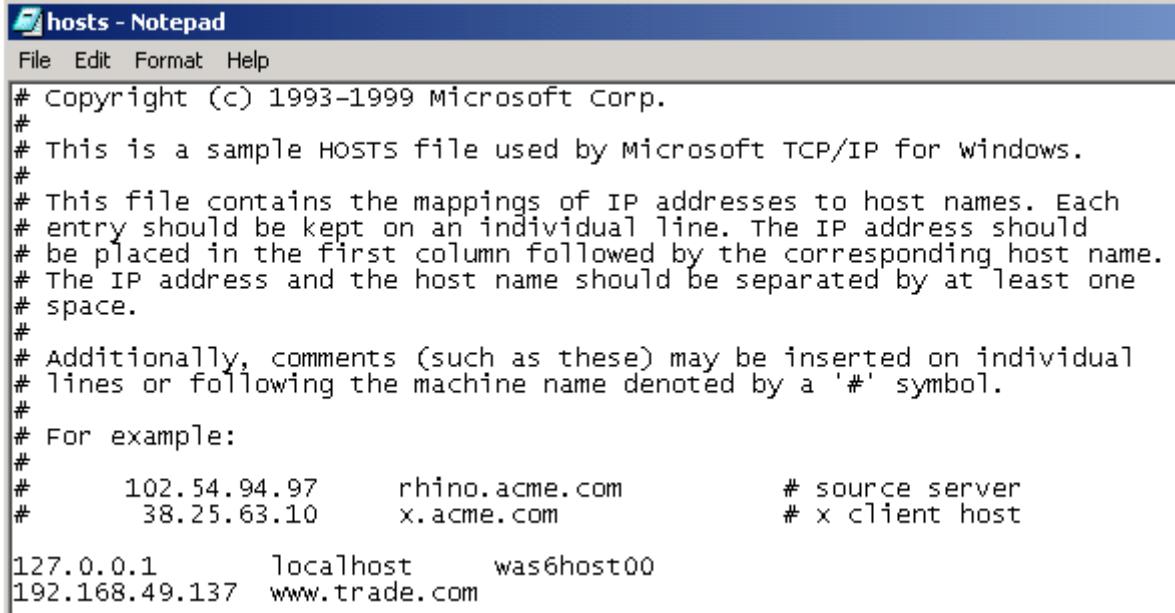
This section of the lab will go through modifying the httpd.conf in order to define the required setting to enable SSL for IBM Http Server. This will include loading the SSL module, defining a listener port, defining a virtual host and enabling SSL.

- \_\_\_ 1. Backup the httpd.conf

- \_\_\_ a. Since changes are about to be made to the httpd.conf, it would be a good idea to make a backup of it before starting. Copy the httpd.conf in <ihs\_root>\conf to httpd-backup.conf
- \_\_\_ 2. Add www.trade.com to hosts file.
  - \_\_\_ a. Edit C:\Winnt\System32\Drivers\etc\hosts

**UNIX:** The hosts file on UNIX systems can be found at /etc/hosts

  - \_\_\_ b. Add a line at the bottom of the hosts file for www.trade.com and use your system's ip address.



```

hosts - Notepad
File Edit Format Help
Copyright (c) 1993-1999 Microsoft Corp.
#
This is a sample HOSTS file used by Microsoft TCP/IP for windows.
#
This file contains the mappings of IP addresses to host names. Each
entry should be kept on an individual line. The IP address should
be placed in the first column followed by the corresponding host name.
The IP address and the host name should be separated by at least one
space.
#
Additionally, comments (such as these) may be inserted on individual
lines or following the machine name denoted by a '#' symbol.
#
For example:
#
102.54.94.97 rhino.acme.com # source server
38.25.63.10 x.acme.com # x client host
#
127.0.0.1 localhost was6host00
192.168.49.137 www.trade.com

```

- \_\_\_ 3. Add Virtual Host definition for HTTPS. This allows for the definition of https on a separate virtual host from http.
  - \_\_\_ a. Edit the **httpd.conf** in <ihs\_root>\conf
  - \_\_\_ b. Load the SSL module by adding the following line after where most of the other Load Modules are done:

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
```

```
#LoadModule mime_magic_module modules/mod_mime_magic.so
#LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
#LoadModule proxy_http_module modules/mod_proxy_http.so
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
#LoadModule rewrite_module modules/mod_rewrite.so
LoadModule setenvif_module modules/mod_setenvif.so
#LoadModule spelling_module modules/mod_speling.so

#LoadModule unique_id_module modules/mod_unique_id.so
LoadModule userdir_module modules/mod_userdir.so
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so
#LoadModule ibm_ssl_module modules/mod_ibm_ssl.so

#
Listen: Allows you to bind Apache to specific IP addresses and/or
ports, in addition to the default. See also the <virtualHost>
directive.
```

- c. Add the following lines to httpd.conf to configure the virtual host and SSL. Make sure to use the correct ServerName for your machine and the appropriate DocumentRoot and Keyfile. Place these lines near the very bottom of the httpd.conf, after the VirtualHost example and before the loading of the WebSphere plug-in module.

**Information:** There are sample configuration files in <software\_dir>\ssl that can be used to copy/paste. If used, make sure to correct the ServerName and paths to correspond with the local system,

```
Listen 443
<VirtualHost www.trade.com:443>
ServerName was6host00
DocumentRoot "C:/Program Files/IBM HTTP Server/htdocs/en_US"
SSLEnable
Keyfile "C:/Program Files/IBM HTTP Server/ssl/ihskKeyring.kdb"
SSLV2Timeout 100
SSLV3Timeout 1000
</VirtualHost>

SSLDisable
```

**UNIX:** There appear to be cases within Linux where it is necessary to use ip addresses instead of full or short names.

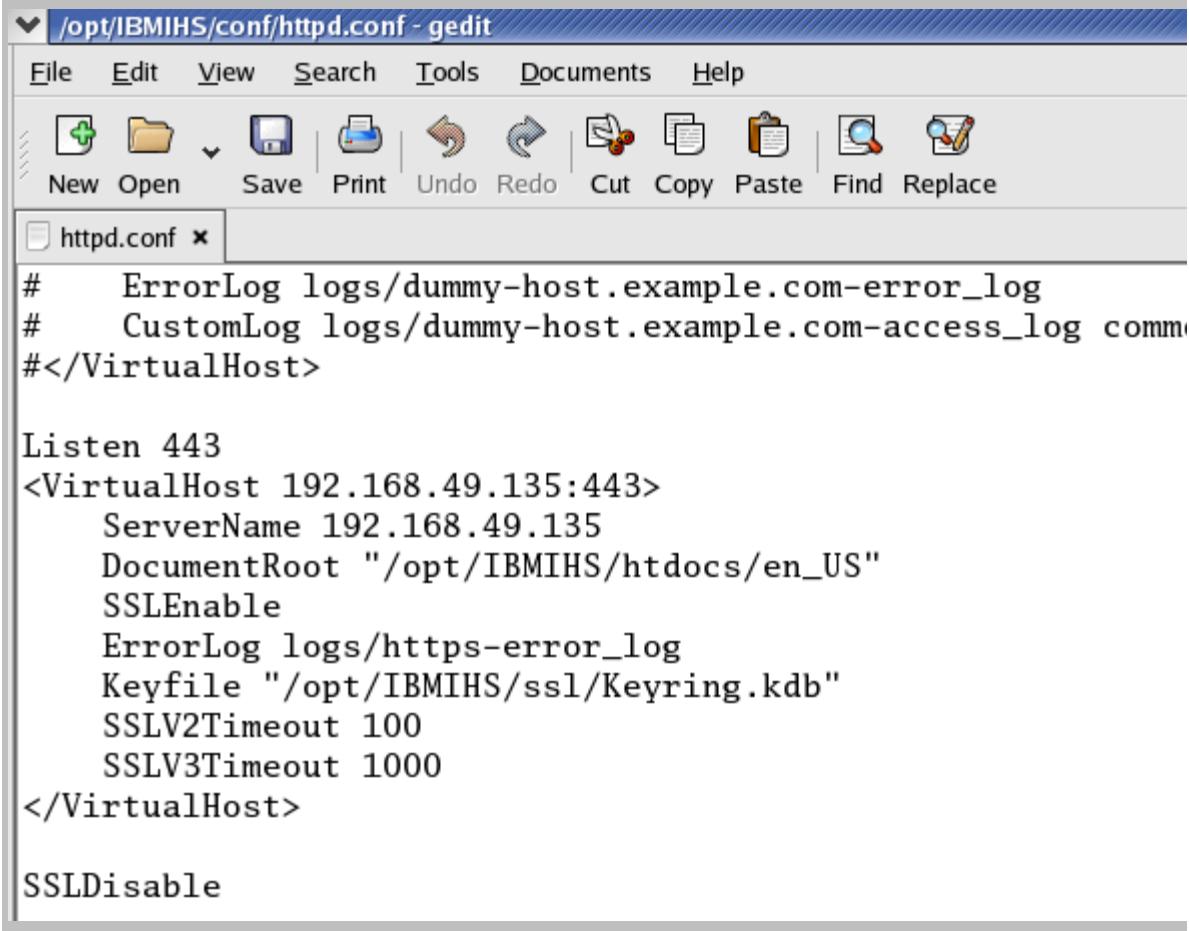
The screenshot shows a window titled "/opt/IBMIHS/conf/httpd.conf - gedit". The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. The toolbar contains icons for New, Open, Save, Print, Undo, Redo, Cut, Copy, Paste, Find, and Replace. A tab bar shows "httpd.conf x". The main text area contains the following configuration code:

```
ErrorLog logs/dummy-host.example.com-error_log
CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>

Listen 443
<VirtualHost 192.168.49.135:443>
 ServerName 192.168.49.135
 DocumentRoot "/opt/IBMIHS/htdocs/en_US"
 SSLEnable
 ErrorLog logs/https-error_log
 Keyfile "/opt/IBMIHS/ssl/Keyring.kdb"
 SSLV2Timeout 100
 SSLV3Timeout 1000
</VirtualHost>

SSLDisable
```

**UNIX:** Make sure the paths and slashes are correct if this is being done on a UNIX system. Also, make sure to specify the virtual host with the ip address instead of the full name.



```
ErrorLog logs/dummy-host.example.com-error_log
CustomLog logs/dummy-host.example.com-access_log comm-
#</VirtualHost>

Listen 443
<VirtualHost 192.168.49.135:443>
 ServerName 192.168.49.135
 DocumentRoot "/opt/IBMIHS/htdocs/en_US"
 SSLEnable
 ErrorLog logs/https-error_log
 Keyfile "/opt/IBMIHS/ssl/Keyring.kdb"
 SSLV2Timeout 100
 SSLV3Timeout 1000
</VirtualHost>

SSLDisable
```

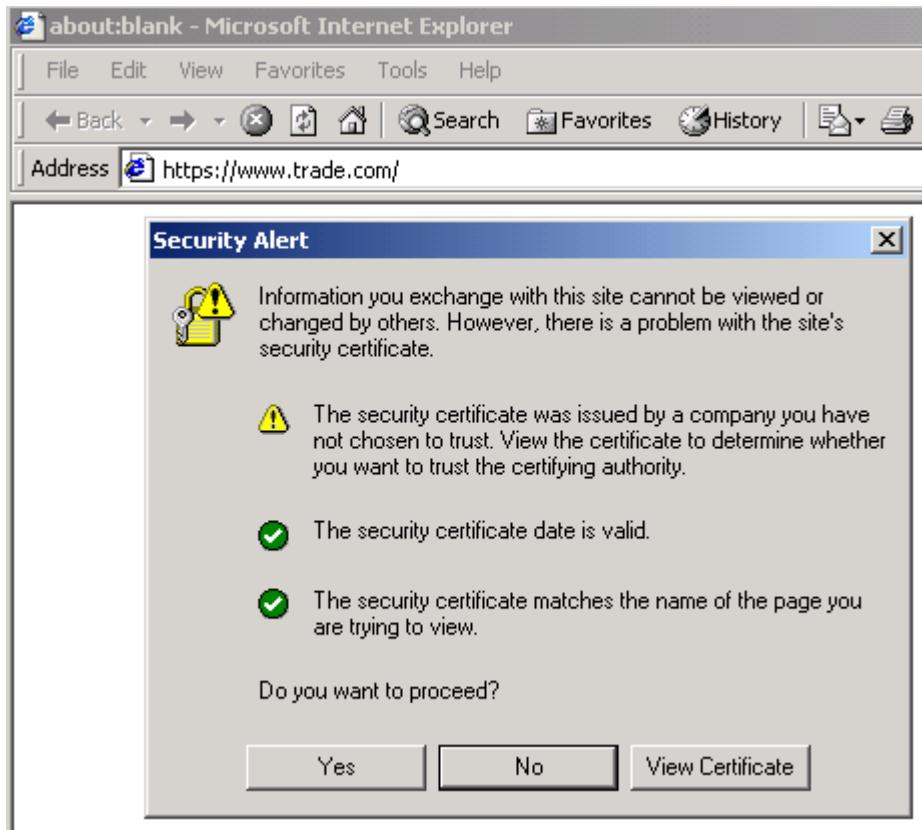
### Step 3 Testing the SSL Connection

- \_\_\_ 1. Restart the IBM Http Server process so that the new httpd.conf settings take effect.
- \_\_\_ a. Using the Window Services, select the IBM HTTP Server 6.0 service and click Restart (or right-click and select Restart).

**UNIX:** On UNIX, use the command <ihs\_root>/bin/apachectl restart to restart the server.

- \_\_\_ b. Verify that the IBM Http Server process is running by checking the system process list. If IBM Http Server failed to start, check the <ihs\_root>\logs\error.log and <profile\_root>\logs\http\_plugin.log for the cause.
- \_\_\_ 2. Connect to IBM Http Server using https
- \_\_\_ a. Open a browser and connect to:  
<https://www.trade.com/>

- \_\_\_ b. There should be a challenge regard the certificate since it is self signed.



- \_\_\_ c. Click **Yes** to accept the certificate.

- \_\_\_ d. The front page for IBM Http Server should now be displayed having used https.



- \_\_\_ 3. In order for https connections to be able to reach the application server, port 443 needs to be enabled on the virtual host.
- \_\_\_ a. Using the administrative console, select **Environment** —> **Virtual Hosts**.
  - \_\_\_ b. Click **default\_host**.
  - \_\_\_ c. On the right, under **Additional Properties**, click **Host Aliases**.
  - \_\_\_ d. Click **New** to create a new host alias.
  - \_\_\_ e. Leave the \* for the **Host Name** and enter **443** for the **Port**.
  - \_\_\_ f. Click **OK** and save the changes.
  - \_\_\_ g. Regenerate and repropagate the Web server plugin-cfg.xml file.
  - \_\_\_ h. Restart the IBM Http Server to pick up the new plugin-cfg.xml file immediately.
  - \_\_\_ i. Restart the WebSphere Application Servers. This will cause them to activate the new virtual host settings.
- \_\_\_ 4. Test the https connection from the browser, through IBM Http Server, and back to WebSphere Application Server using snoop.
- \_\_\_ a. Ensure that the WebSphere Application Server profile1 is running.

- \_\_\_ b. Using a browser, verify that snoop is reachable with the following URL:  
<http://www.trade.com/snoop>
- \_\_\_ c. Now try snoop using https, using the following URL (make sure to use https):  
<https://www.trade.com/snoop>



- \_\_\_ d. Notice that the snoop servlet is showing that the connection was made via https.

**Information:** WebSphere Application Server comes with SSL enabled from the plug-in to the application server. This is a security issue since the encryption is based on the DummyKeyring which is shared with all other WebSphere Application Server installations. This Keyring needs to be replaced in order to truly secure the communications between the plug-in and the application server.

### ***End of lab***

## Exercise review and wrap-up

This lab introduced basic https configuration concepts for both IBM Http Server and WebSphere Application Server.



# Exercise 14. Enabling WebSphere and J2EE security

## What this exercise is about

This exercise secures the WebSphere Application Server V6 Administrative Console and then verifies that access is limited to certain functions.

It also secures the administration module of the Trade application. The application is tested with Global security enabled and an explanation, through exploration of configuring J2EE security in Application Server Toolkit (AST) is presented.

## What you should be able to do

By completing this lab, you will learn to configure WebSphere Application Server V6 to limit access for the Administrative Console to particular users.

You should also be able to map Users and Groups to J2EE security roles and to understand how J2EE security is configured using AST.

## Introduction

This lab will enable global security within WebSphere Application Server. It will then configure access to the Administrative console by defining a number of roles and mapping those roles to existing users. In order to test the configuration, you will attempt to perform various functions from the various users and see if the security configurations correctly limits access to various functions.

The last part of this lab deals with configuring J2EE security for the Trade application using the WebSphere administrative console and the AST.

## Required materials

This exercise requires a workstation with WebSphere Application Server V6 installed and that the previous exercises for this course have been successfully completed.

## Instructor exercise overview

This exercise is simple and straight forward, as long as the previous labs have been completed successfully. Since this lab is run on an ND environment it is important that ALL servers be stopped and restarted after Global security is enabled. Note that SWAM authentication mechanism is NOT available in an ND environment, LTPA is used instead, and it needs a password be set before global security enablement is accepted by the console.

In the optional part of the exercise, the explanation of how security works and how to configure using the AST, it is assumed that the student can open, close and navigate AST. You might have to do a quick demo to refresh students' memories, but they will have used the AST by the time they get to this exercise.

## Exercise instructions

### Step 1 Defining WebSphere administrative console users

This section of the lab will create users and map them to different levels of console access.

- \_\_\_ 1. First create some local users for testing purposes. From a Command Window, create the following users/passwords by using the following commands:

```
net user wasadmin was1edu /add
net user wasconfig was1edu /add
net user wasmonitor was1edu /add
net user wasoperator was1edu /add
```

```
C:\>net user wasadmin was1edu /add
The command completed successfully.

C:\>net user wasconfig was1edu /add
The command completed successfully.

C:\>net user wasmonitor was1edu /add
The command completed successfully.

C:\>net user wasoperator was1edu /add
The command completed successfully.
```

**UNIX:** Please refer to the appropriate appendix for the instructions required to create these users.

- \_\_\_ 2. Verify that the Deployment Manager is running.
- \_\_\_ 3. Bring up a WebSphere administrative console and log in as admin.
- \_\_\_ 4. In the WebSphere administrative console, expand **System Administration** in the left navigation and select **Console settings -> Console Users**. Click **Add**.



- \_\_\_ 5. Enter **wasadmin** for the **User** and select the **Administrator** role. Click **OK**. Repeat these last two steps for **wasconfig**, **wasmonitor** and **wasoperator**.

**Console Users > Add**

Configuration for adding, updating and removing Console Users.

**General Properties**

\* User  
wasadmin

\* Role(s)  
Administrator  
Configurator  
Operator  
Monitor

Buttons: Apply, OK, Reset, Cancel

- \_\_\_ 6. Once all 4 users are mapped, Save the changes.

| Select                   | User                        | Role(s)       | Login Status |
|--------------------------|-----------------------------|---------------|--------------|
| <input type="checkbox"/> | <a href="#">wasadmin</a>    | Administrator | Not Active   |
| <input type="checkbox"/> | <a href="#">wasconfig</a>   | Configurator  | Not Active   |
| <input type="checkbox"/> | <a href="#">wasmonitor</a>  | Monitor       | Not Active   |
| <input type="checkbox"/> | <a href="#">wasoperator</a> | Operator      | Not Active   |

Total 4

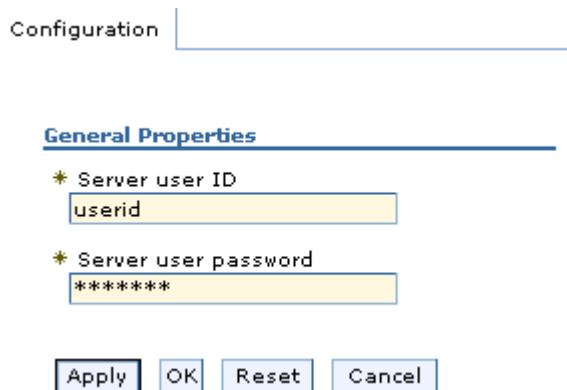
## Step 2 Enabling Global Security

In this section of the lab, WebSphere Global security will be enabled through the Console.

- \_\_\_ 1. Configure the **Local OS** user registry.
- \_\_\_ a. Select **Security -> Global security**, and click **Local OS** on the right side under User registries.



- \_\_\_ b. Enter **userid** for the **Server user ID** and **was1edu** for the **Server user password**. Click **OK**.



**UNIX:** Enter **root** for the Server user ID and **was1edu** for the Server user password.

- \_\_\_ c. **Save** the changes.  
\_\_\_ 2. Set the LTPA password  
\_\_\_ a. Select **Security** → **Global security**.  
\_\_\_ b. Under **Authentication**, expand **Authentication mechanisms**.  
\_\_\_ c. Select **LTPA**.

- \_\_ d. Enter **was1edu** for the Password and Confirm password.

[Global security > LTPA](#)

Specifies the Lightweight Third Party Authentication (LTPA) configuration settings. When security is enabled and any of these properties are changed, go to the Global Security panel, located under Security in the left navigation menu. Click Apply or OK to validate the changes. The LTPA keys are automatically generated the first time that you enable security. After you enable security, WebSphere Application Server can generate a new set of keys in two ways. If you need to change the password, make the change and click OK or Apply to generate the keys. You do not need to click Generate Keys. If you do not need to change the password, click Generate Keys. The new set of keys are not used until they are saved.

Configuration

Generate Keys Import keys Export Keys

|                                                                                                                                                   |                                       |                              |
|---------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|------------------------------|
| <b>General Properties</b>                                                                                                                         |                                       | <b>Additional Properties</b> |
| * Password<br>*****                                                                                                                               | ▪ <a href="#">Single signon (SSO)</a> |                              |
| * Confirm password<br>*****                                                                                                                       | ▪ <a href="#">Trust association</a>   |                              |
| * Timeout<br>120                                                                                                                                  |                                       |                              |
| Key file name<br><input type="text"/>                                                                                                             |                                       |                              |
| <input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/> |                                       |                              |

- \_\_ e. Click **OK**.  
 \_\_ f. **Save** the changes.  
 \_\_\_ 3. Enable Global Security.

- \_\_\_ a. Return to **Security -> Global security**, and check **Enable global security**. Uncheck **Enforce Java 2 security**.

The screenshot shows the 'Global security' configuration panel. On the left is a navigation tree with 'Global security' selected under 'Security'. The main panel has a title 'Global security' and a sub-section 'Global security' with the following description: 'Specifies the global security configuration for 1. Configure the desired user registry listed 2. Select the security option on this panel. 3. Select the configuration panel.' Below this is a 'Configuration' section containing a 'General Properties' table:

| <u>General Properties</u>           |                                   |
|-------------------------------------|-----------------------------------|
| <input checked="" type="checkbox"/> | Enable global security            |
| <input type="checkbox"/>            | Enforce Java 2 security           |
| <input type="checkbox"/>            | Enforce fine-grained JCA security |
| <input type="checkbox"/>            | Use domain-qualified user IDs     |

- \_\_\_ b. Select **LTPA** for the **Active authentication mechanism** and **Local OS** for the **Active user registry**.

The screenshot shows the 'Active authentication mechanism' and 'Active user registry' configuration panel. It includes dropdown menus for both settings and a checkbox for FIPS compliance. At the bottom are four buttons: 'Apply', 'OK', 'Reset', and 'Cancel'.

|                                                                                                                                                   |                                                                              |
|---------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| Active authentication mechanism                                                                                                                   | Lightweight Third Party Authentication (LTPA)                                |
| Active user registry                                                                                                                              | Local OS (single, stand-alone server or sysplex and root administrator only) |
| <input type="checkbox"/> Use the Federal Information Processing Standard (FIPS)                                                                   |                                                                              |
| <input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/> |                                                                              |

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

- d. Read the message, disregard the single signon paragraph, since you are not using it.

Messages

⚠ Changes have been made to your local configuration. Click [Save](#) to apply changes to the master configuration.

💡 The server may need to be restarted for these changes to take effect.

⚠ The domain name for single signon is not defined. The Web browser defaults the domain name to the host name that runs the Web application. Single signon is restricted to the application server host name and does not work with other application server host names in the domain.

⚠ The Local OS user registry is supported only when the cell is configured on a single machine or when you use a Windows domain controller.

⚠ The security configuration is enabled or modified in a Network Deployment environment. The following steps need to be followed so that all the processes in this environment have the same security run-time settings: 1) Verify that all nodes are synchronized with these security configuration changes before stopping these processes. 2) If any node agents are currently stopped, issue a manual **syncNode** command before starting that node agent. 3) Stop all of the processes in the entire cell, including the deployment manager, node agents, and Application Servers. 4) Restart all of the processes in the cell; restart the deployment manager and node agents first, then Application Servers.

⚠ If Java 2 security is not enabled, the Java virtual machine (JVM) system resources are not protected. For example, applications can read and write to files on file systems, listen to sockets, exit the Application Server process, and so on. However, by enabling Java 2 security, applications might fail to run if the required permissions are not granted to the applications.

💡 If any of the fields are changed, save the configuration and then stop and restart the server.

- e. **Save** the changes.
4. Resynchronize both nodes
5. Restart all WebSphere Application Server processes in this order:
- Stop the cluster, which stops the application servers (console).
  - Stop the node agents (console).
  - Stop the Deployment manager (command line).
6. Using the command line, start the deployment manager and node agents. Note: You will start the the cluster later once you have logged in to the console.

### Step 3 Authenticate to the WebSphere administrative console and testing

In this section of the lab, access to the console will be granted to only correctly mapped users. And depending on the role to which they are mapped, the console will allow those users to execute only certain functions.

**Information:** By default, the userid used to define the authentication mechanism has implicit access as an administrator role.

- \_\_\_ 1. Bring up a new **WebSphere administrative console** and attempt to log in. There should be a Security alert. Click **Yes** to accept the console's certificate.



- \_\_\_ 2. Log into the WebSphere administrative console using the userid **wasadmin**.

- \_\_\_ 3. Verify that full access to administrative functions are available.
- \_\_\_ a. Check the functions available by going to **Applications -> Enterprise Applications**.

- \_\_\_ b. Notice that all standard functions are available.

| Select                   | Name                                  | Status |
|--------------------------|---------------------------------------|--------|
| <input type="checkbox"/> | <a href="#">DefaultApplication</a>    |        |
| <input type="checkbox"/> | <a href="#">QuoteWS</a>               |        |
| <input type="checkbox"/> | <a href="#">Trade Order Processor</a> |        |
| <input type="checkbox"/> | <a href="#">TradeApplication</a>      |        |

- \_\_\_ 4. Start the **TradeCluster** from the **Servers -> Clusters** page.  
 \_\_\_ 5. Check the available functions for other users.  
 \_\_\_ a. **Logout** from the console. The Logout button is located near the top of the page.

WebSphere Administrative Console - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back → × Search Favorites History

Address https://localhost:9043/ibm/console/secure/securelogon.do?action=secure

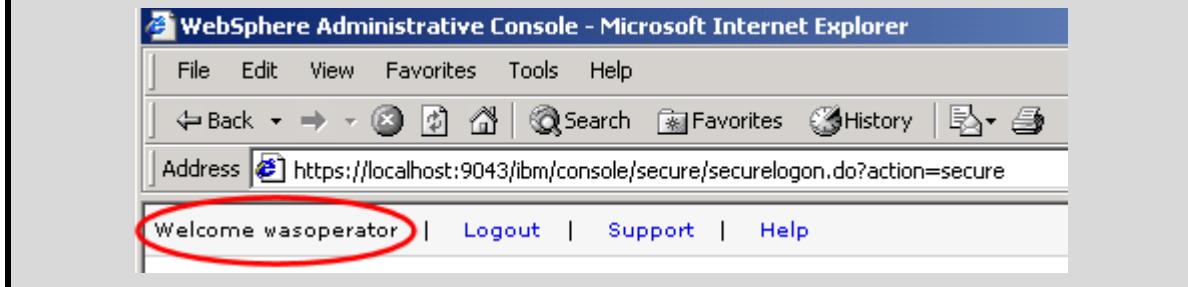
Welcome wasadmin | **Logout** | Support | Help

- \_\_\_ b. Log back into the console as **wasoperator**, and see what functions are available on the Enterprise applications.

| Select                   | Name                                  | Status |
|--------------------------|---------------------------------------|--------|
| <input type="checkbox"/> | <a href="#">DefaultApplication</a>    |        |
| <input type="checkbox"/> | <a href="#">QuoteWS</a>               |        |
| <input type="checkbox"/> | <a href="#">Trade Order Processor</a> |        |
| <input type="checkbox"/> | <a href="#">TradeApplication</a>      |        |

- \_\_\_ c. Notice that not all the same function are available. Now only **Start**, **Stop**, **Export** and **Export DDL** are available.

**Information:** Notice that it is possible to see what user ID was used to log in to the WebSphere administrative console by looking in the upper left corner of the page.



- \_\_\_ 6. Log in as **wasconfig** and **wasmonitor** and see what functions are available.
- \_\_\_ 7. Verify that **userid** has implicit access to the console, even though it has not been explicitly defined as a console user.
  - \_\_\_ a. **Logout** of the console.
  - \_\_\_ b. **Log in as userid**
  - \_\_\_ c. Notice what functions are available for the Enterprise applications.
- \_\_\_ 8. Close all Web browsers. This will ensure there are no existing cookies when you start the next part of the exercise.

This concludes the Console security part of this exercise. Next you will work with securing the Adminstrative functions of the Trade application.

## Step 4 Securing the Trade application

When running with security enabled, enterprise applications can take advantage of J2EE security to restrict access to servlet and EJB resources. The Trade application's administration module has already been configured to take advantage of J2EE security, all that needs to be done by the administrator is to map users to security roles. A security role of **TradeAdmin** has been defined in the application.

**Information:** Java 2 security can also be used to provide fine grained access to system resources, such as ports, sockets, file system, and so forth. Java 2 security is orthogonal to J2EE security and does not require Global security to be enforced. In this exercise you do not use Java 2 security.

- \_\_\_ 1. Test the application before mapping roles to users and groups.
  - \_\_\_ a. With all the servers up and running, access URL:  
`http://localhost/Trade/web/welcome.html`

- \_\_ b. Click Administration.



**Information:** Note that only the Administration part of the application has been enabled for security. The rest of the application can be accessed just like before when Global security was not enabled.

- \_\_ c. Try to login to the administration module of the Trade application. Enter **userid** and **was1edu..**

The screenshot shows the 'Welcome to Trade Application Administration' page. It features a login form with fields for 'User' (containing 'userid') and 'Password' (containing '\*\*\*\*\*'). A 'Login' button is below the password field. To the left of the form, there is a link 'Welcome page'.

- \_\_\_ d. Click **Login**. You get an error 403, **AuthorizationFailed** error page. This is because you have not defined **userid** as a user which can assume the role of **TradeAdmin**.
- \_\_\_ 2. Create a new group in the Local OS user registry that will become the authorized group to access the administration module of the Trade application.
  - \_\_\_ a. From a command window, create the **wasadmins** group by entering the command:  
`net localgroup wasadmins /add`
  - \_\_\_ b. Add the **userid** and **wasadmin** users to the group:  
`net localgroup wasadmins userid /add`  
`net localgroup wasadmins wasadmin /add`
  - \_\_\_ c. Close the command window.
- \_\_\_ 3. Map users and groups to roles.
  - \_\_\_ a. Make sure you are logged in to the WebSphere administrative console with an id that gives you administrator privileges, like **wasadmin**, or **userid**.
  - \_\_\_ b. Select **Applications** —> **Enterprise Applications**.
  - \_\_\_ c. Click **TradeApplication**.

- \_\_\_ d. Under **Additional Properties**, click **Map security roles to users/groups**. Note at this time there are no users or groups mapped to the TradeAdmin security role.

[Enterprise Applications](#) > [Trade Application](#) > [Mapping users to roles](#)

Map security roles to users/groups

Each role that is defined in the application or module must map to a user or group from the domain user registry.

|                          |            | Look up users            | Look up groups           |              |               |  |
|--------------------------|------------|--------------------------|--------------------------|--------------|---------------|--|
| Select                   | Role       | Everyone?                | All authenticated?       | Mapped users | Mapped groups |  |
| <input type="checkbox"/> | TradeAdmin | <input type="checkbox"/> | <input type="checkbox"/> |              |               |  |

**OK**    **Cancel**

**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 has been no previous opportunity to authenticate, that rules out All authenticated.

The mapped entries refer to users and groups in the current user registry. As you remember, at the current time the Local OS user registry is being used.

- \_\_\_ e. Select the **TradeAdmin** role.
- \_\_\_ f. Click **Look up groups**.
- \_\_\_ g. Make sure the Search String is \*. Click Search. Note that the Available list fills up with the groups defined in the Local OS user registry. If wasadmin does not appear in the Available list because there are many entries, change limit to 60 and search again.

- \_\_\_ h. Select WAS6HOST00\wasadmins and click >> to copy the entry to the Selected list.

[Enterprise Applications](#) > [Trade Application](#) > [Mapping users to roles](#) > Look up users or groups

Specifies whether to look up users and/or groups.

The following roles are mapped to the items in the selected list.

|                                                                                                        |                                                                                    |
|--------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| <input checked="" type="checkbox"/> TradeAdmin<br><input type="checkbox"/><br><input type="checkbox"/> | <input type="button" value="&lt;&lt;"/><br><input type="button" value="&gt;&gt;"/> |
|--------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|

To search for users or groups, type in a limit (number) and a search pattern (such as a\*) and click the "Search" button:

limit (number of items)  
20

Search String  
\*

Select users or groups below in the "Available" list. Move them to the "Selected" list by clicking on the >> button.

|                                                                                                                                                                                                                                                                                                                                                |                                                                                        |                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| <p>Available:</p> <ul style="list-style-type: none"> <li>WAS6HOST00\Administrators</li> <li>WAS6HOST00\Backup Operators</li> <li>WAS6HOST00\Guests</li> <li>WAS6HOST00\Power Users</li> <li>WAS6HOST00\Replicator</li> <li>WAS6HOST00\Users</li> <li>WAS6HOST00\DB2ADMNS</li> <li>WAS6HOST00\DB2USERS</li> <li>WAS6HOST00\wasadmins</li> </ul> | <input type="button" value="&gt;&gt;"/><br><br><input type="button" value="&lt;&lt;"/> | <p>Selected:</p> <ul style="list-style-type: none"> <li>WAS6HOST00\wasadmins</li> </ul> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|

- \_\_\_ i. Click OK.

[Enterprise Applications](#) > [Trade Application](#) > [Mapping users to roles](#)

Map security roles to users/groups

Each role that is defined in the application or module must map to a user or group from the domain user registry.

| Look up users      Look up groups                                       |            |                          |                          |              |                      |
|-------------------------------------------------------------------------|------------|--------------------------|--------------------------|--------------|----------------------|
| <input type="checkbox"/> <input type="checkbox"/>                       |            |                          |                          |              |                      |
| Select                                                                  | Role       | Everyone?                | All authenticated?       | Mapped users | Mapped groups        |
| <input type="checkbox"/>                                                | TradeAdmin | <input type="checkbox"/> | <input type="checkbox"/> |              | WAS6HOST00\wasadmins |
| <input type="button" value="OK"/> <input type="button" value="Cancel"/> |            |                          |                          |              |                      |

- \_\_\_ j. Click OK.

- \_\_\_ k. Save your changes. Make sure that the **Synchronize** check box is selected.
- \_\_\_ 4. Close all Web browser windows. Open a new Web browser.
- \_\_\_ 5. Access the welcome page of the Trade application. Click **Administration**.
- \_\_\_ 6. Log in using either of the users in the wasadmins group, userid or wasadmin.
- \_\_\_ 7. This time you are authenticated properly and allowed access.

## Trade Application Administration

The following Administrative Functions are available:

| Function                       |
|--------------------------------|
| <a href="#">Adjust Balance</a> |
| Manage Passwords               |

[Logout](#)

**Information:** The administration module of the trade application allows administrators to adjust the balance of any user registered to the system. At this time the Manage Passwords option is not implemented.

- \_\_\_ 8. From the menu select **Adjust balance**.

**All user balances**

Below are the current users of the Trade Application.

To adjust a user's balance select the user and click **Set balance**.

| Full Name    | Account Number | Current Balance | New Balance          | Select                |
|--------------|----------------|-----------------|----------------------|-----------------------|
| Cisco Kid    | 147789999      | \$123.00        | <input type="text"/> | <input type="radio"/> |
| Lone Ranger  | 148443822      | \$2,100.00      | <input type="text"/> | <input type="radio"/> |
| James Bond   | 343335454      | \$32,000.00     | <input type="text"/> | <input type="radio"/> |
| Mako Shark   | 123444321      | \$100,000.00    | <input type="text"/> | <input type="radio"/> |
| Big Bird     | 123456789      | \$50,000.00     | <input type="text"/> | <input type="radio"/> |
| Mr. Student  | 987654321      | \$123.00        | <input type="text"/> | <input type="radio"/> |
| Jessie James | 87652289       | \$10,000.00     | <input type="text"/> | <input type="radio"/> |

**Set balance**

[Administration menu](#) [Logout](#)

- \_\_\_ 9. Enter a new amount for any of the users, select its radio button and click **Set balance**.
- \_\_\_ 10. Confirm that the amount changed.
- \_\_\_ 11. Click **Logout**.
- \_\_\_ 12. Click **Logout** again to return to the welcome page.

**Information:** Once you logout, the security context is disassociated and you will need to authenticate again to access Administration.

- \_\_\_ 13. Disable Global security
- \_\_\_ 14. Stop all the servers as you did before. Make sure to use the **-username** and **-password** options from the command line when stopping the servers.
- \_\_\_ 15. Restart all the servers as you did before.

## Step 5 How does it work? (optional)

To do this part of the exercise you need to open AST to the workspace you created earlier in this class when assembling the Trade application. The AST is used to explore the EAR file, the TradeEJB and Tradeweb modules to see how security is configured.

Instructions on using the AST itself will be short and to the point, you should be familiar with AST from the previous exercises.

**Information:** The AST tool, with the application assembly plug-in, is currently only available for Windows and Linux.

- \_\_\_ 1. Start AST and point it to the workspace where you assembled the Trade application.
- \_\_\_ 2. Switch to the J2EE perspective.
- \_\_\_ 3. The first interesting question on your mind probably is: Why clicking the Administration button on the welcome page causes an authorization challenge to be presented, and how does the application know which page to display with the login form?
  - \_\_\_ a. On the Project Explorer expand **Dynamic Web Projects**.
  - \_\_\_ b. Expand **TradeWeb**. Double-click **Deployment Descriptor: TradeWeb**. The deployment descriptor editor for this module opens on the editor pane on the upper right corner of the window.

- \_\_\_ 4. From the tabs along the bottom of the editor pane, select **Pages**. If necessary scroll to the bottom of the page to reveal the **Login** portion of the page.

**Welcome Pages**  
The server will search for the following pages, in this order, when the web application is referenced:

- Welcome File List
  - welcome.html
  - login.jsp
  - index.jsp

Add... Remove

**Login**  
The following Login configuration values are used for this web application:

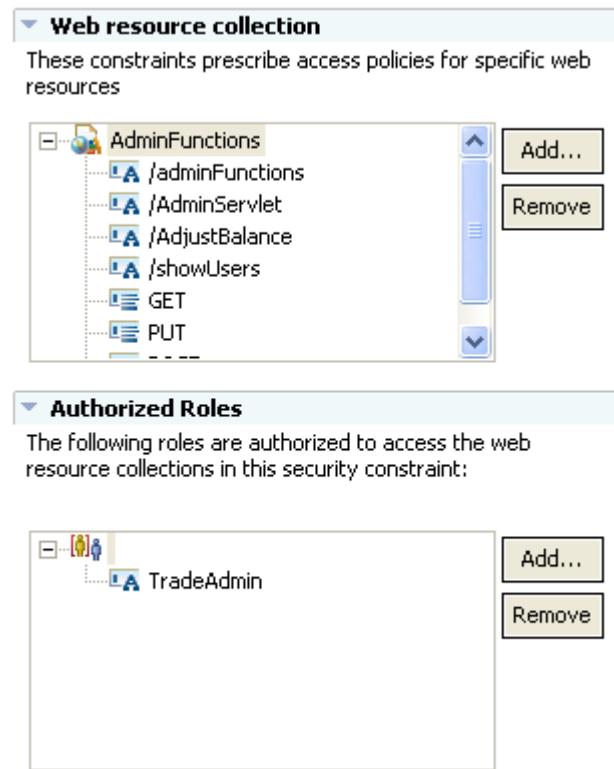
|                        |                   |
|------------------------|-------------------|
| Authentication method: | FORM              |
| Realm name:            |                   |
| Login page:            | /adminlogin.jsp   |
| Error page:            | /loginfailure.jsp |

Note that the Authentication method is set to **FORM** which means the application provides its own login, and error page, in this case adminlogin.jsp and loginfailure.jsp respectively. You may open these pages if you wish to see how they are coded.

When a protected resource is accessed by a non-authenticated user WebSphere Application Server presents the login page instead of the requested resource. After successful authentication the originally requested resource is served.

- \_\_\_ 5. The next question, then is: How are resources protected or secured?
- Click the **Security** tab of the editor.
  - Select the **TradeAdmin** security role.
  - Click the padlock icon under **Security Constraints**.

- \_\_\_ d. Expand **AdminFunctions** under **Web resource collection**.



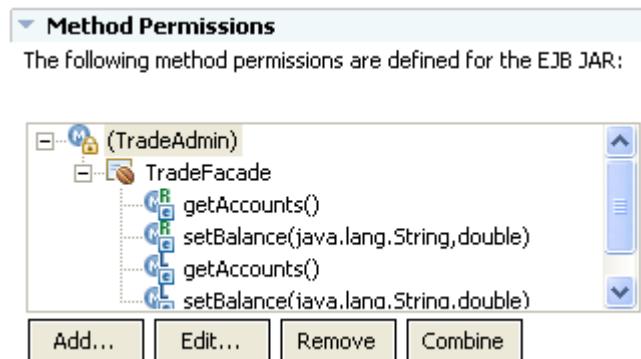
Web resource collections are a set of URIs and HTTP access methods which can be assigned to an Authorized role. In this case you see that one of the URIs is **/adminFunctions**, this is the page called when the **Administration** button is clicked on the welcome page. At the bottom of the figure above note that the AdminFunctions resource collection is associated with the **TradeAdmin** role.

Any time any of the above resources is accessed by a non-authenticated user, the login page is presented instead.

This is how the Web module is protected.

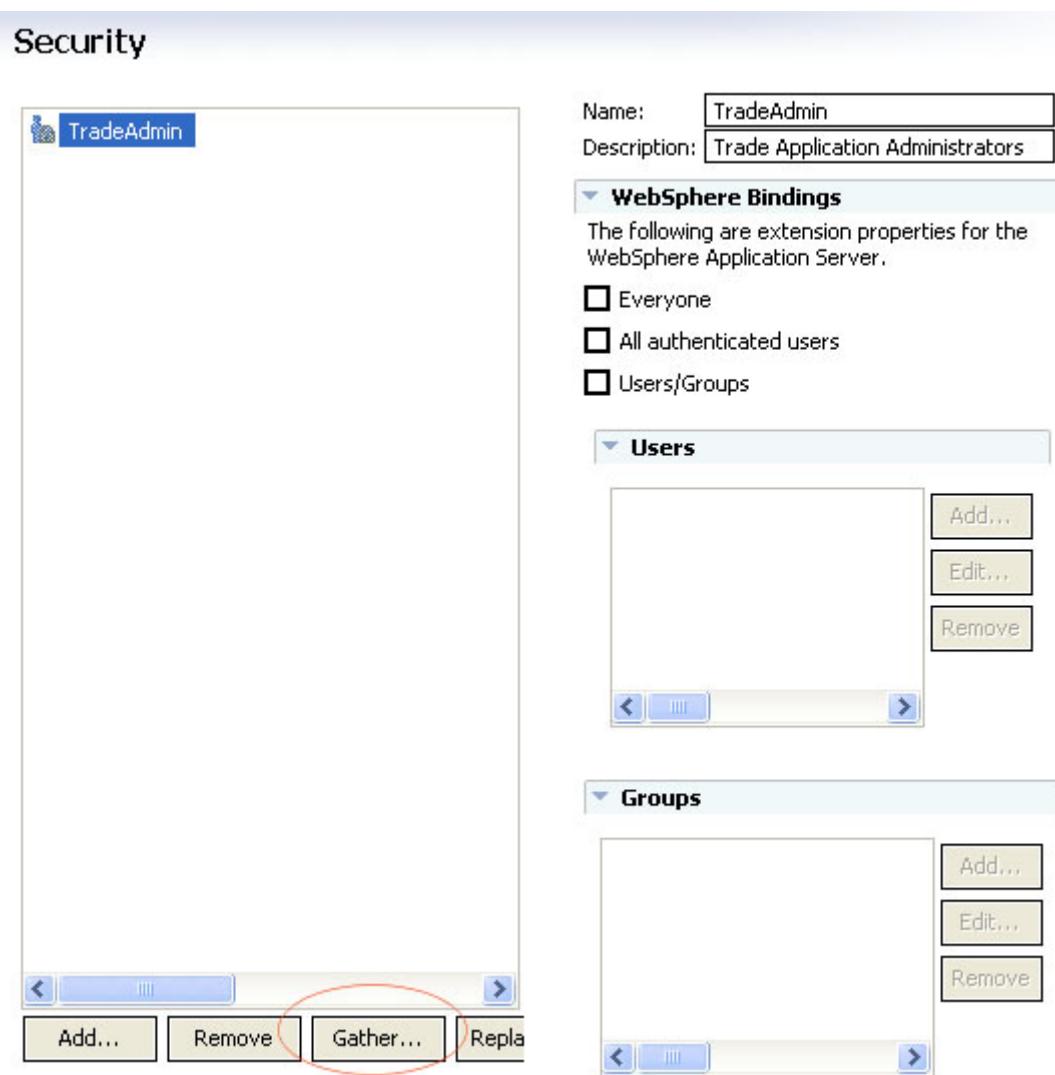
- \_\_\_ 6. EJBs are protected through method permissions, all EJBs in this application are accessed through a session bean called **TradeFacade**. Therefore protecting the EJBs only involves protecting key methods of this bean.
- \_\_\_ a. On the Project Explorer expand **EJB Projects**.
  - \_\_\_ b. Expand **TradeEJB**. Double-click **Deployment Descriptor: TradeEJB**. The deployment descriptor editor for this module opens on the editor pane on the upper right corner of the window.
  - \_\_\_ c. Click the **Assembly** tab of the editor.

- \_\_\_ d. Locate the **Method Permissions** area of the page.



- \_\_\_ e. Here you see that the **TradeAdmin** security role has been granted permission to access four methods of the **TradeFacade** EJB. These methods are the ones called to get the registered accounts and the set the balances for the accounts.
- \_\_\_ 7. Security roles can be added through the Application, EJB and Web module deployment descriptor editors. The application deployment descriptor needs to be aware of all security roles defined in every module of the application. For this purpose a **Gather** function has been added to the Application's deployment descriptor editor.
- \_\_\_ a. On the Project Explorer expand **Enterprise Applications**.
- \_\_\_ b. Expand **TradeApplication**. Double-click **Deployment Descriptor: TradeApplication**. The deployment descriptor editor for this module opens on the editor pane on the upper right corner of the window.
- \_\_\_ c. Click the **Security** tab of the editor.

- \_\_ d. Click **Gather...**.



Note that the TradeAdmin security role is visible here, as would all the security roles defined in all modules of the application. Also notice that you could set the mapping between Users and groups, and Security roles right here. Whether to do it here, or through the WebSphere administrative console at installation time will depend on the security policies of your organization.

- \_\_ 8. Close the AST when you finish exploring.

## End of lab

## Exercise review and wrap-up

The first part of the exercise looked at setting up security for accessing the WebSphere administrative console. After creating the appropriate users Global security was enabled and access to the console was tested.

Next J2EE security for the administration module of the Trade application was setup. the wasadmins group was assigned the security role of TradeAdmin. The application was tested.

Lastly an explanation of how J2EE security is configured in AST was presented.



# Exercise 15. Integrated Performance Viewer (optional)

## What this exercise is about

In this exercise you will use the integrated Tivoli Performance Viewer to monitor various application server resources in real time.

## What you should be able to do

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

- Start and stop Monitoring Process
- Monitor application server resources
- Use the features and functions supplied with the Tivoli Performance Viewer (Performance Advisor)
- Use Request Metrics Data
- Write/Read a data collection session to a log (optional)

## Introduction

The integrated Tivoli Performance Viewer is the user interface for monitoring the performance of application servers, servlets and other resources in the WebSphere Application Server. The Performance Viewer 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.

## Required materials

To do this lab you require a properly set up computer with WebSphere Application Server V6 installed as well as the necessary startup and application files.

## Instructor exercise overview

This lab deals mainly with the setup of the PMI settings and work with the integrated Tivoli Performance Viewer.

We use first the default Web application with snoop. If server1 is stand-alone (contains still

running Admin Console), the students need to scroll or filter in the TPV to see the Snoop Servlet, because all components of the Admin Application are shown (step 2).

In the next step you use /hitcount to monitor the Increment EJB, which is a CMP Entity Bean and have only one instance. Compare this with the ivt application and use the /ivtejb servlet to create many stateless Session Beans.

- first with **/hitcount** (using **CMP EJB**) --> only one instance

- Answer to question in lab: **only 300 calls to one instance, existing Bean is activated**

- second with **/ivtejb** (using **Session EJB**) --> each request - one Session Bean

- Answer to question in lab: **300 instances of EJB**

**Note:** /ivtejb returns a blank screen only.

For creating stress you will use **ApacheBench** with the -n option to create a number of tests. Be careful with the -c option (concurrent user) of ApacheBench when using hitcount working with CMP, because of the Cloudscape data source has some limitation for concurrent connections. (Highly skilled students can change this data source to DB2.)

The only files needed for this exercise are located in  
**<software\_dir>\Troubleshooting\SVGViewer** (adobe plug-in for the browser). Use the Trade application for the Request Metrics part of the lab.

The Integrated Performance Viewer is a new tool. Because of this, the recommendation is to do this lab on a single server. Highly skilled students can play around with dmgr and different server as an optional part of the lab.

# Exercise instructions

## Preface

In this exercise you will use Integrated Performance Viewer to monitor applications running at WebSphere Application Server V6 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. This architecture enables the monitoring work to be distributed among the nodes.

In this lab you monitor applications running on profile1's single server, server1.

## Step 1 Start Performance Monitoring

Before the Performance Viewer can begin monitoring performance data, the monitoring service must be started. The Monitoring service is enabled by default on the application server.

- \_\_\_ 1. Verify the deployment manager, node agent and WebSphere Application Server V6 server1 in profile1 are all running.
- \_\_\_ 2. Start the Admin Console and log in.
- \_\_\_ 3. Verify the Performance Monitoring Infrastructure (PMI) is enabled for server1.
  - \_\_\_ a. Click **Monitoring and Tuning**. Click **Performance Monitoring Infrastructure (PMI)**. On the workspace area on the right pane of the WebSphere administrative console, click **server1**.
  - \_\_\_ b. On the **Configuration** tab **Enable Performance Monitoring Infrastructure (PMI)** is already selected. This is the default setting. In the **Currently monitored**

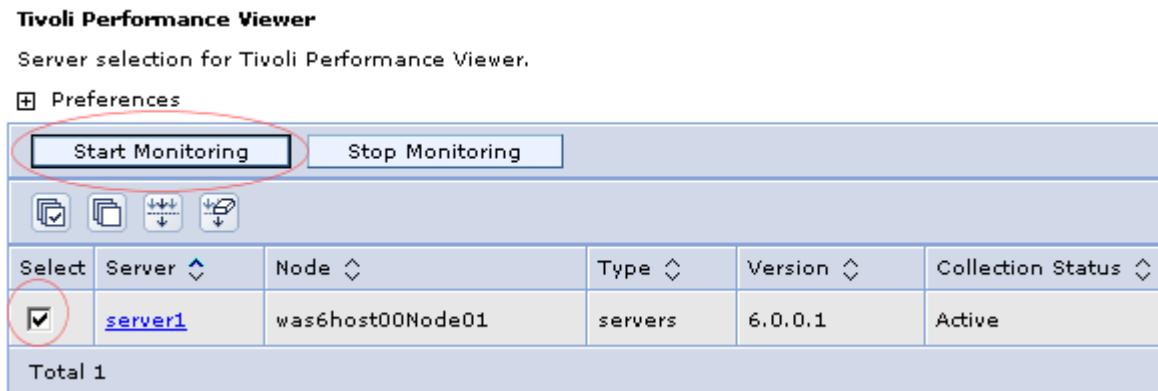
**statistic set** area select **Basic** level as shown below (this may already be selected as the default).

- \_\_\_ c. Click the **Runtime** tab and check the selection **Basic** (may already be selected as the default).
- \_\_\_ d. Click **OK**. It is not necessary to restart the server for the new settings to take effect.

**Information:** The settings on the **Runtime** tab are the ones used to monitor the runtime environment. These settings can be changed and the new settings are applied immediately after clicking **OK**. If the **Persist my changes** check box is selected the runtime settings are saved and will appear in the Configuration tab. This setup allows you to make temporary changes to the configuration, with the option of the runtime settings becoming part of the next configuration.

- \_\_\_ 4. In order to get more frequent data collection, change the **Monitoring Refresh Rate** to 20 seconds
  - \_\_\_ a. 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**:



- \_\_\_ c. Click **server1** to view its Currently Activity. If the tree for server1 is collapsed click **[+]** by 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 "User Settings" panel. On the left is a tree view of server1 modules: Advisor, Settings (which has User and Log), Summary Reports (which has Servlets, EJBs, EJB Methods, Connection Pool, and Thread Pool), and Performance Modules. The "User" node under "Settings" is expanded. On the right is a "Data Collection" panel with "Refresh Rate" set to 20 seconds and "Buffer Size" set to 40. Below it is a "View Data As" section with radio buttons for "Raw" (selected), "Change In Value", and "Rate Of Change". At the bottom are "Apply" and "Cancel" buttons.

## **Step 2 Introduction to viewing Web Modules and Data**

To view any performance data in a graphical mode, the browser must be enabled to show SVG Data. Scalable Vector Graphics (SVG) is a new graphics file format and Web development language, which can be displayed using an SVG Viewer, or SVG enabled browser.

- \_\_\_ 1. Install the SVG Viewer from Adobe before using the browser to view the data produced by the Performer viewer.

- \_\_ a. Run the svgviewer.exe program to install this viewer. This program can be found in the <software\_dir>\Troubleshooting\SVGViewer directory.

**UNIX:** Install the appropriate SVG plug-in into your browser!

- \_\_ b. Log out from the WebSphere administrative console and restart the browser.

**Information:** If you are using **Mozilla or Firefox** as your browser, there may be some graphical data visible without installing SVGViewer, but you should install this program and plug-in, because some details and the table view are only available when this product is installed.

To download the latest version of SVGViewer go to:

<http://www.adobe.com/svg/>

- \_\_ 2. Open a new browser and start the snoop servlet. The Web address (URL) for the snoop servlet is **http://localhost:9080/snoop**. This executes the snoop servlet, which is part of the Default Application and brings up 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 will come back here soon.
- \_\_ 3. If you do not have a running **WebSphere administrative console** open it for Dmgr and log in. Go to the Tivoli Performance Viewer and monitoring server1.
  - \_\_ a. Click **Monitoring and Tuning** → **Performance Viewer** → **Current Activity** → **server1**.
  - \_\_ b. Expand **Summary Reports**, select **Servlets**. On the Servlet Summary Report, locate the entry for the **Snoop Servlet**, which is part of **DefaultWebApplication.war**.

**Information:** You may have to go to the next page of the report to find the listing for Snoop Servlet. Use the arrows at the bottom of the page to navigate to the different pages of the report. As an alternative you may use filters to reduce the amount of information shown on the report (For example, **D\*** for **Application**, not for Name). The second alternative is to sort by Application (^).

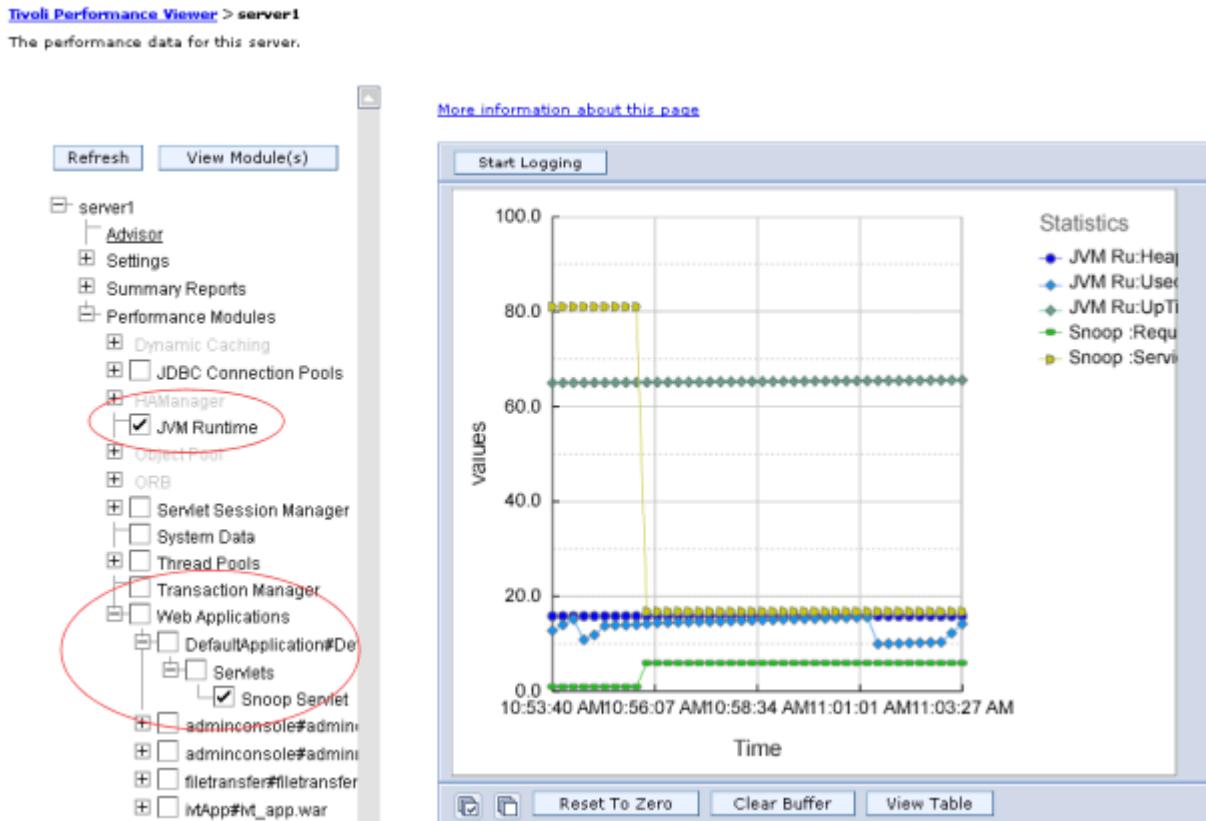
### Servlets Summary Report

[More information about this page](#)

| <a href="#">Start Logging</a>                                                                                                                                           |               |                           |                  |               |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|---------------------------|------------------|---------------|
|   |               |                           |                  |               |
| Select                                                                                                                                                                  | Name ▲        | Application ▲             | Total Requests ▲ | Avg Resp Time |
|                                                                                                                                                                         |               | Filter: D*                |                  |               |
|                                                                                                                                                                         | Snoop Servlet | DefaultWebApplication.war | 4                | 10            |
| Total 48                                                                                                                                                                |               |                           |                  |               |

- \_\_ c. Additional activities:

- Are there any EJBs being monitored? Click EJBs under Summary Reports.
  - Any Connection or Thread Pools? These display in graph form. If this is the first time displaying a graph in the browser after installing the SVG plug-in, you may be asked to activate the plug-in.
- 4. Inside Tivoli Performance Viewer (TPV) set the data monitoring level for the Snoop Servlet and the JVM Runtime using the default Basic level.
- a. In TPV expand **Performance Modules** and select the **JVM Runtime** check box. Expand **Web Applications** —> **DefaultApplication** —> **Servlets** and select **SnoopServlet**, If **SnoopServlet** is not available, try to call snoop again using the other browser then click **Refresh** on the TPV (Tivoli Performance Viewer).
- b. Click **View Modules** button. You will see a table or a graph with the monitoring data represented.



- c. Use the browser to reload the snoop servlet some times by clicking the browser refresh button several times. Review the changes in the Console.

- \_\_\_ d. In the Integrated Performance Viewer click the **View Graphic** or **View Table** to switch to see the other display of the data..

The screenshot shows the Tivoli Performance Viewer interface. On the left, a tree view of the server1 components is displayed, with several nodes circled in red. The circled nodes include 'JVM Runtime' under 'HAManager', 'Snoop Servlet' under 'DefaultApplication#De...', and multiple entries under 'adminconsole#admin'. On the right, a table titled 'Start Logging' displays JVM Runtime statistics over time. The table has columns for Time, JVM Runtime HeapSize, JVM Runtime UsedMemory, JVM Runtime UpTime, Snoop Servlet RequestCount, and ServiceTime. The data shows measurements from 10:55:56 AM to 10:53:10 AM.

| Time        | JVM Runtime HeapSize | JVM Runtime UsedMemory | JVM Runtime UpTime | Snoop Servlet RequestCount | ServiceTime |
|-------------|----------------------|------------------------|--------------------|----------------------------|-------------|
| 10:55:56 AM | 158590.00            | 141903.00              | 65098.00           | 6.00                       |             |
| 10:55:41 AM | 158590.00            | 140279.00              | 65083.00           | 1.00                       |             |
| 10:55:26 AM | 158590.00            | 139702.00              | 65068.00           | 1.00                       |             |
| 10:55:10 AM | 158590.00            | 139073.00              | 65053.00           | 1.00                       |             |
| 10:54:55 AM | 158590.00            | 138229.00              | 65038.00           | 1.00                       |             |
| 10:54:40 AM | 158590.00            | 118899.00              | 65023.00           | 1.00                       |             |
| 10:54:25 AM | 158590.00            | 108892.00              | 65008.00           | 1.00                       |             |
| 10:54:10 AM | 158590.00            | 151915.00              | 64993.00           | 1.00                       |             |
| 10:53:55 AM | 158590.00            | 140071.00              | 64978.00           | 1.00                       |             |
| 10:53:40 AM | 158590.00            | 128011.00              | 64963.00           | 1.00                       |             |
| 10:53:25 AM | 158590.00            | 116102.00              | 64948.00           | 1.00                       |             |
| 10:53:10 AM | 158590.00            | 103969.00              | 64933.00           | 1.00                       |             |

**Information:** You can toggle back and forth between the table view and the graphic view by selecting the **View Table** or **View Graph** button (below the data).

- \_\_\_ e. Reload the snoop servlet several times. You can also test the HelloWorldServlet using the URL <http://localhost:9080/hello> in the browser window.
- \_\_\_ f. To monitor other modules and components you may need to click the **Refresh** button and the **View Modules** again.

### Step 3 Viewing EJB Modules of different applications

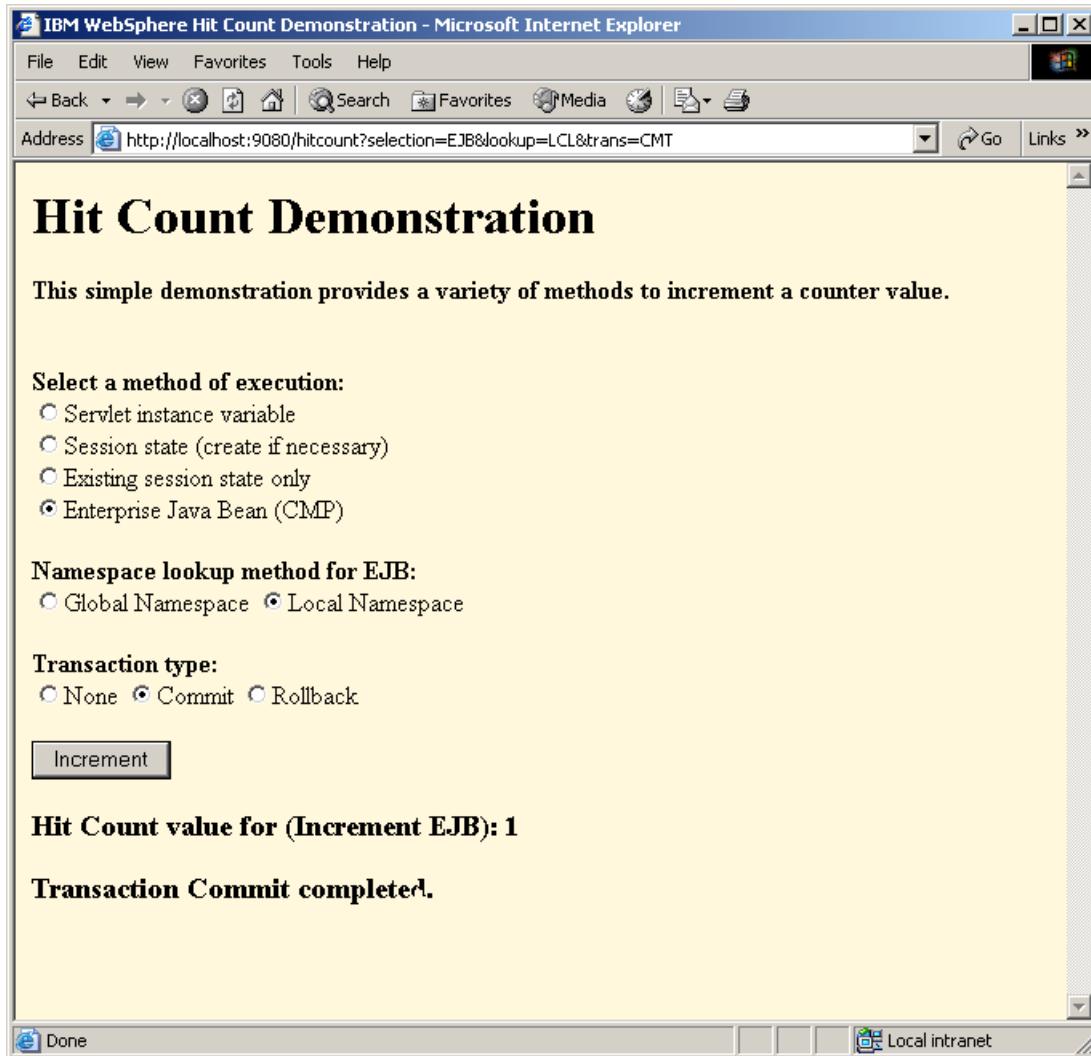
- \_\_\_ 1. Change the PMI Runtime Setting to a higher level: **All** for server1.
- \_\_\_ a. Using the WebSphere administrative console select **Monitoring and Tuning** → **Performance Monitoring Infrastructure (PMI)** click **server1**.
- \_\_\_ b. Click the **Runtime** tab and select **All** in the **Currently monitored statistic set** area. Do not select to **Persist my changes**. Click **OK**.
- \_\_\_ c. Click **Monitoring and Tuning** → **Performance Viewer** → **Current Activity**.
- \_\_\_ d. Click **server1**.

- \_\_\_ e. Click **Deselect all items**.
  - \_\_\_ f. Click **Clear Buffer** which is located under the table area.
2. Start monitoring two different EJBs from two different applications (/hitcount and /ivtejb). To monitor both EJBs you need to run each of the applications first so that EJBs are loaded and data collection can take place.

**Information:** You may remember from previous exercise “Problem Determination”, that you used **hitcount** and **ivtejb** to trace Connections. Now you want to monitor the EJB components of the applications: **Increment** and **ivtEJBObject**.

- \_\_\_ a. Start the request of /hitcount using a browser with the following URL:  
`http://localhost:9080/hitcount`
- \_\_\_ b. To load the EJB Increment of this application chose the following options on the page:
  - Select a method of execution: **Enterprise JavaBean (CMP)**
  - Namespace lookup method for EJB: **Local Namespace**
  - Transaction type: **Commit**

- Click Increment



- \_\_\_ c. You should now have the following URL in your browser:

`http://localhost:9080/hitcount?selection=EJB&lookup=LCL&trans=CMT`

Leave the browser open with this URL.

- \_\_\_ d. Start a **new browser window** with the following URL

`http://localhost:9080/ivt/ivtejb`

With this request you have activated the ivtEJBObject of the ivt application.  
Also leave this browser window (with blank screen) open.

**Information:** When running the ivt application below, you will see a **only blank screen** as response in the browser. This is normal and expected.

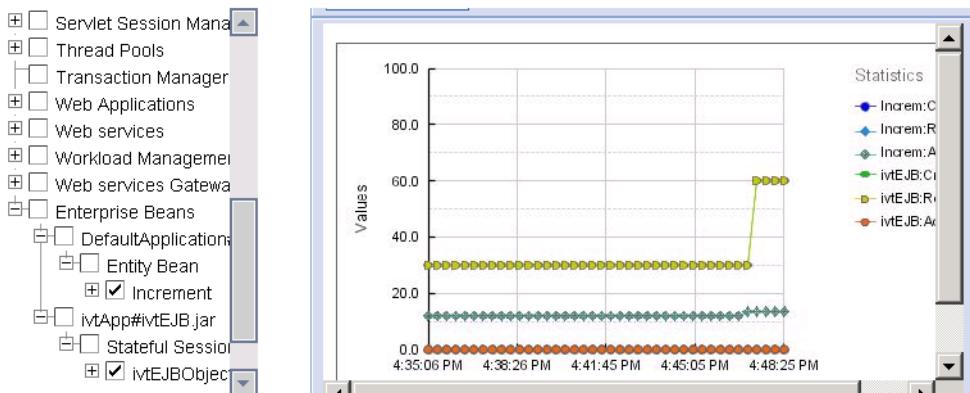
- \_\_\_ e. Back on the WebSphere administrative console, in the Tivoli Performance Viewer page (**Monitoring and Tuning** —> **Performance Viewer** —> **Current**

**Activity —> server1), click Refresh.** Expand **Performance Modules**. Expand **Enterprise Beans --> Default Application --> Entity Beans**. (if none are present click **Refresh**, you may need to click more than once). Do the same for ivtApp.

- \_\_\_ f. Enable for Monitoring both EJBs: **Increment** and **ivtEJBObject**.

Enterprise Beans->DefaultApplication->Entity Bean >Increment.

Enterprise Beans->ivtApp->Stateful Session Bean->ivtEJBObject).



- \_\_\_ g. You can use both browser windows to reload the servlets again:

/ivt/ivtejb

/hitcount?selection=EJB&lookup=LCL&trans=CMT

But you will want to create some “more” stress in the next step. Still leave both browser windows open.

- \_\_\_ 3. To create a simple stress test use ApacheBench (ab[.exe]) which is installed as part of IBM Http Server.

**Information:** ApacheBench is part of Apache Server and a very simple tool for quickly generation HTTP requests. You will use AB to stress the application server and you will want to monitor the EJB creation and activation process. The syntax is simple:

ab -options "http://hostname/requestURL"

Option -n: number of requests to perform

**UNIX:** Available also on LINUX/UNIX.

- \_\_\_ a. Open a command prompt window, change directory to the IBM Http Server directory (C:\Program Files\IBM HTTP Server\bin).

**Information:** You can copy and paste the information from the open browser windows in the next two steps.

**UNIX:** Navigate to the appropriate directory using Terminal Window.

Enter the following command to request the hitcount application 300 times: (all in one command line)

```
ab -n300 "http://localhost:9080/hitcount
?selection=EJB&lookup=LCL&trans=CMT"
```

- View the monitor's output using graphical view or table view. You may have to click **View Module** again.
  - How many Increment EJBs are in this application? Are there 300 created Beans?\_\_\_\_\_ (Hint: look at CreateCount!)  
Are there existing Beans activated?\_\_\_\_\_ (Hint: look at the ActivateCount)
- \_\_\_ b. Start ApacheBench with the ivtApp using the following command line (still in the IBM HTTP Server\bin directory):
- ```
ab -n300 "http://localhost:9080/ivt/ivtejb"
```
- View the monitor again. What about CreateCount and ActivateCount of ivtEJBObject? Why are the results not the same?_____
- ___ c. Go to the **Summery Reports -> EJBs** of the TPV. Expand ivtEJBObject and visit the number of Method Calls of the *.create* and *.remove* method. Compare the numbers with the calls of the Increment EJB.

Information: The hitcount application is calling a CMP entity bean (Increment), which is used for all of the 300 and more requests. You should see more than 300 ActivateCounts of the **one** Increment EJB. The ivtejb application simple try to test a Stateless Session EJB, which is created during each request, instead of ActivateCount you should see an number 300 or higher of CreateCount (or *.create*).

The number of EJBs may be an hint for a performance issue.

Step 4 Using Performance Advisor in TPV

WebSphere Application Server V6 introduces a new advisor, the TPV Advisor, which runs in the Tivoli Performance Viewer and provides more extensive advice, for example, dynachache size, JVM heap size, and so forth. The TPV Advisor provides recommendations on inefficient settings at specific point in time.

- ___ 1. In WebSphere Application Server V6 you can directly access to Advisory Messages using the WebSphere administrative console.

- ___ a. In the Tivoli Performance Viewer of WebSphere administrative console (**Monitoring and Tuning —> Performance Viewer —> Current Activity —> server1**). Click **Advisor** under **server1**.

The screenshot shows the Tivoli Performance Viewer interface. On the left is a navigation tree:

- Welcome
- Servers
- Applications
- Resources
- Security
- Environment
- System administration
- Monitoring and Tuning** (selected)
 - Performance Monitoring Infrastructure (PMI)
 - Request Metrics
 - Performance Viewer**
 - Current Activity**
 - View Logs**
- Troubleshooting
- Service integration
- UDDI

The main panel title is "Tivoli Performance Viewer". Below it, the path is "Tivoli Performance Viewer > server1". A message says "The performance data for this server." At the top right are "Refresh" and "View Module(s)" buttons. Below them is a tree view of the server's modules:

- server1
 - Advisor** (highlighted with a red oval)
 - + Settings
 - + Summary Reports
 - + Performance Modules

At the bottom right is a "Deselect all items" button.

- ___ b. On the area to the right, read the Warnings.
Are there some Alert messages?_____

Is there a Config Advice message? _____

If more than one page is available don't forget to look there too.

		<input type="button" value="Refresh All Advice"/>	<input type="button" value="Remove Selected Advice"/>
Select	Severity	Message	Status
<input type="checkbox"/>	Warning	TUNE0303W: Number of threads workin...	Read
<input type="checkbox"/>	Warning	TUNE0303W: Number of threads workin...	Unread
<input type="checkbox"/>	Warning	TUNE0318I: There is no data availab...	Read
<input type="checkbox"/>	Warning	TUNE0318I: There is no data availab...	Unread
<input type="checkbox"/>	Alert	TUNE0224W: Consider lowering the mo...	Read

Page: 1 of 2 Total 6

Information: In your lab environment the CPU utilization is usually low. You should see the Config Advice to enable the servlet caching with descriptions as to how to use the console to enable this feature.

You should run the performance advisor with a representative workload. ApacheBench is only a simple tool, not designed for distributed software testing. A more robust tool is OpenSTA (<http://opensta.org/>).

Hint: The TPV advisor requires that you enable performance modules, counters, or both.

More details in: IBM Redbook, SG24-6392, *IBM WebSphere Application Server V6 Performance and Scalability*.

Step 5 Enabling Request Metrics

Request Metrics log the time spent at major components of the application server, such as the Web server plug-in, Web container, EJB container, and so forth.

Request Metrics is different from Performance Monitoring Infrastructure (PMI), because PMI provides information about average system resource usage statistics, with no correlation between the data across different WebSphere components.

Request Metrics tracks each individual transaction within WebSphere Application Server, recording the response time of the major components such as time in the Web server or in the Enterprise JavaBean (EJB) container. The information can be saved to log files.

- ___ 1. Using the WebSphere administrative console, enable Request Metrics select **Monitoring and Tuning —> Request Metrics**.
 - ___ a. Under **General Properties**, select **Enable Request metrics**.

- b. Under **Components to be instrumented**, select **ALL**.
 - c. Set **Trace level to Debug**.
 - d. Under **Request Metrics Destination** select **Standard Logs**.
 - e. Click **OK**.
 - f. **Save** to the master configuration.
2. On a Web browser, run the TradeApplication. Use the following URL:
- `http://localhost:9080/Trade/web`
- 3. Log in to the Trade application, trade some stock to exercise the application.
 - 4. On the `<profile_root>/profile1/logs/server1` directory, open the **SystemOut.log** file. Find the entries beginning with **PmiRmArmWrapp**:

UNIX: Navigate to the `<profile_root>/profile1/logs/server1` directory.

```

PmiRmArmWrapp I PMRM0003I:
parent:ver=1,time=1102439810783,pid=3232,reqid=9,event=1 -
current:ver=1,ip=192.168.100.50,time=1102439810783,pid=3232,reqid=1
0,event=1 type=Web Services Requestor
detail=wsrequestor:StockQuoteSoap.getQuote?transport=http&parameters
s=symbol elapsed=541
PmiRmArmWrapp I PMRM0003I:
parent:ver=1,time=1102439810783,pid=3232,reqid=8,event=1 -
current:ver=1,ip=192.168.100.50,time=1102439810783,pid=3232,reqid=9
, event=1 type=EJB detail=com.ibm.trade.TradeFacadeBean.getQuote
elapsed=551
PmiRmArmWrapp I PMRM0003I:
parent:ver=1,time=1102439810783,pid=3232,reqid=8263,event=1 -
current:ver=1,ip=192.168.100.50,time=1102439810783,pid=3232,reqid=8
271,event=1 type=EJB
detail=com.ibm.trade.TradeFacadeBean.getAccountInfo elapsed=651
PmiRmArmWrapp I PMRM0003I:
parent:ver=1,time=1102439810783,pid=3232,reqid=8309,event=1 -
current:ver=1,ip=192.168.100.50,time=1102439810783,pid=3232,reqid=8
310,event=1 type=EJB
detail=com.ibm.trade.HoldingsBean.findAllHoldingsByAccountId
elapsed=0

```

Information: You see a lot of *PmiRmArmWrapp* entries. The Fields *time* and *pid* are start time and ID of application server process. The entries are *type* and *detail* are description of code type and name of the component. Most important is the measured *elapsed* time in milliseconds, which includes all sub operations called by this operation.

The *reqid* is an unique ID assigned by Request Metrics.

Note: If you have time to do the optional part of this exercise, skip the next two steps, which reset the server to a non-monitoring state. **Perform these steps after completing the optional part of the exercise, or if you will not do the optional part of the exercise.**

- ___ 5. Disable **Request Metrics**.
 - ___ a. Select **Monitoring and Tuning** —> **Request Metrics**.
 - ___ b. Deselect **Enable Request Metrics**.
 - ___ c. Click **OK**.
 - ___ d. **Save** the master configuration.
- ___ 6. Change the Level of PMI to the default value, which is **Basic**.
 - ___ a. Select **Monitoring and Tuning** —> **Performance Monitoring Infrastructure (PMI)**.
 - ___ b. Click **server1**.
 - ___ c. On both the **Runtime** and **Configuration** tabs, under **Currently monitored statistic set**, select **Basic**.
 - ___ d. Leave the **Enable Performance Monitoring Infrastructure (PMI)** selected in the **Configuration** tab.
 - ___ e. Click **OK**.
 - ___ f. **Save** to the Master Configuration
 - ___ g. Restart the server.

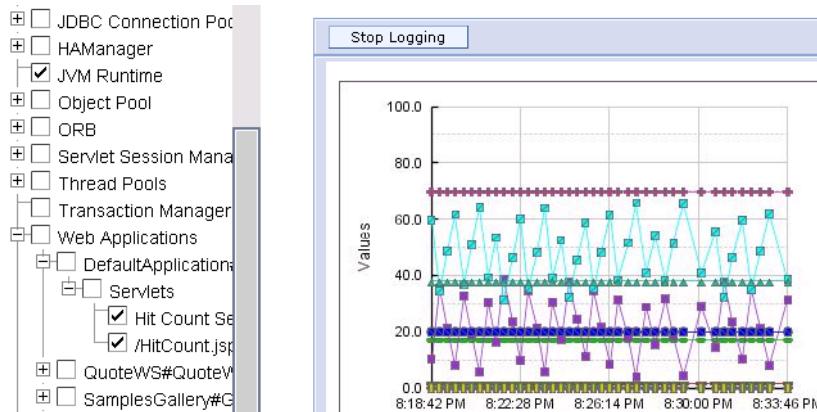
Information: Working with real-time data logging and replaying will be done in the optional part of this exercise. Go to the next page for the description.

End of lab

Optional exercises

Step 1 Using a log file in Tivoli Performance Viewer

- ___ 1. You can use TPV to create a log file which is stored as zip file in the log directory: <profile_root>\profile1\logs\tpv.
- ___ a. Select **Monitoring and Tuning** → **Performance Monitoring Infrastructure (PMI)** → **server1** and set the **Currently monitored statistic** set to **ALL** in the **Runtime** tab.
- ___ b. Click **OK**.
- ___ c. Select **Monitoring and Tuning** → **Performance Viewer** → **Current Activity** → **server1**. Expand **Performance Modules** and select the following:
 - **JVM Runtime**
 - **Web Applications** → **DefaultApplication** → **Servlets** → **HitCountServlet** and **/HitCount.jsp**



Information: If there are no servlets available start the application pointing the Web browser to: <http://localhost:9080/hitcount>

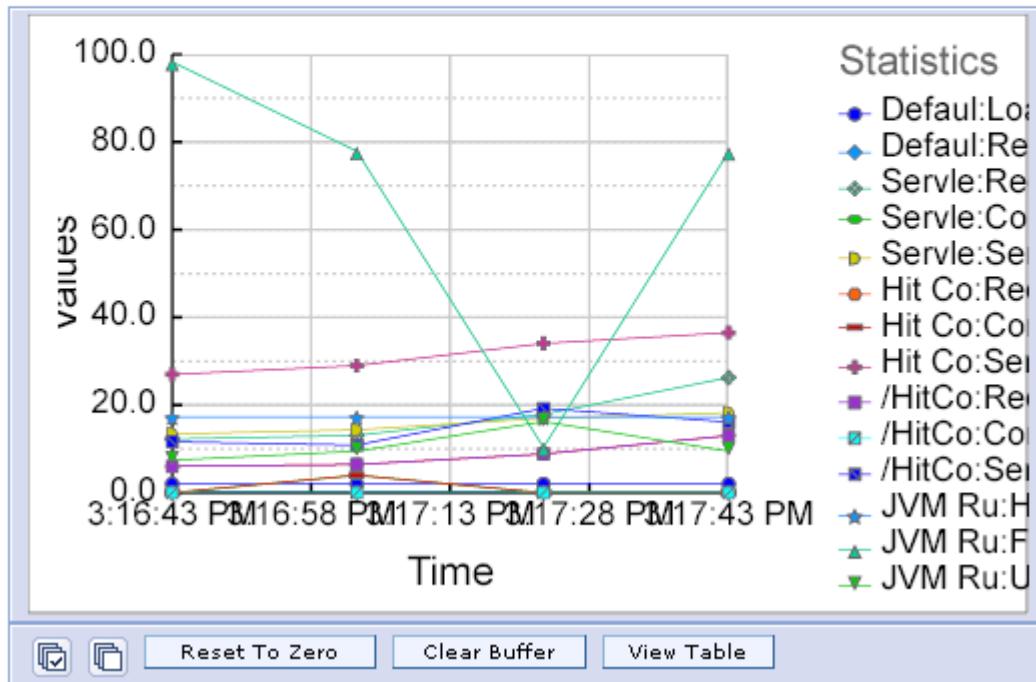
- ___ 2. Click **Start Logging**. The button's text should change to **Stop Logging**
- ___ 3. Run ApacheBench (with 30 concurrent users - only using servlets):


```
ab -n700 -c30 "http://localhost:9080/hitcount?selection=SS2"
```
- ___ 4. After one or two minutes run the same command again. When the command completes. Click **Stop Logging**.
- ___ 5. Select and view the Logs
 - ___ a. Select **Monitoring and Tuning** → **Performance Viewer** → **View Logs**.
 - ___ b. Under **View Logged Data**, select **Explicit Path to Log File**.

- ___ c. Click **Browse**. Navigate to the <profile_root>/profile1/logs/tpv directory. There should be file named tpv_server1<A_Date>.zip. Select this file with **Open** and click **View Log**.
- ___ d. On the tree control on the left, under the file name you selected you can navigate through the data stored in the log. This is the performance data stored for the JVM Runtime and both HitCountServlet /HitCount.jsp in the DefaultWebApplication.
- ___ e. Expand Performance Modules, select JVM Runtime.
- ___ f. Expand Web Applications, expand DefaultApplication and click servlets. Expand servlet and select Hit Count Servlet and HitCount.jsp.
- ___ g. If you are not able to switch to the Graph view, expand **Summary Reports** and click **Thread Pool**. Then click the **View Module** button.
- ___ h. Before you are able to review the real-time log data you may have to click << **Rewind** on the top of the display area.

« ■ ▶ ▷ Playing

[More information about this page](#)



- ___ i. Click the >> Fast Forward button. Look to the time stamp on the button of the Graphical Area. Try to Rewind, and so forth.

Information: The Tivoli Performance Viewer (TPV) provides an easy way to store real-time data for system resources, WebSphere pools and queues, and applications in log files for later retrieval. This logging feature is a special viewer that allows you to view the data without knowledge of the syntax and format of the log files.

By default, the data replays at the Refresh Rate specified in the user settings. You can choose the Fast Forward modes to play data at a faster rate. It will load the next data point every three seconds. Fast Forward 2 loads ten data points every three seconds.

- ___ 6. You may wish to return to Step 1 of this exercise and stop monitoring **server1**. Follow steps in Step 1 except press the **Stop Monitoring** button.

End of lab

Exercise review and wrap-up

In this lab, you used PMI architecture to monitor application server resources and applications. You have enabled the Request Metrics Data function in WebSphere Application Server V6. You worked with the integrated Tivoli Performance Viewer to view the data and to get Performance Advisor information. You are able to collect monitoring session data as log file.

Appendix A. General UNIX commands and information

This Appendix contains general UNIX commands and information as well as vi commands to enable using that editor.

General UNIX commands

Connecting and disconnecting to machines

ftp machine_id - Start an ftp session used to transfer files between machines

login - login to the OS

logout - Log out

su userid - Switch users for this shell

talk userid - Talk to another user

telnet machine_id - Start a terminal session to another machine

File management

chmod [-fR] <mode> file - Change access control on files

cd directory - Change to a new directory

cp file1 file2 - Copy a file

ls <arguments> <directory> - List files in a directory

mkdir new_directory - Create a directory

mv old_name new_name|directory - Change a file's name, move a file to a new directory

pwd - Displays current path

rm file - Delete (remove) files

tac file - Displays last additions to a file even as the file is being modified

whereis file - Reports location of a file

Information

finger userid - Find information on a logged in user

man <command> - Brings up manual pages on a topic

more file - Displays a file and allows for paging forward in that file. Other commands that do listings can be piped to more as well using the `... | more` syntax

passwd - Changing your password

quota -v - Available disk space quota

who - Finding out who's logged on

whoami - Who am I logged in as

Process management

Cntrl-z - Suspend a process

exit - Exit from a shell

fg - Resume a suspended process

kill processid - Terminate (kill) a process

nohup <command> - Causes a process to ignore hangup signals. Used to release ties from a process to its shell

ps <arguments> - List Processes

Modifiers

grep id - Piped after a command to identify search criteria for that command

more - Piped to more will allow paging through results

Using vi commands

vi is a text editor that operates in two modes, Command mode and Insert mode. Command mode indicates that keystrokes will be interpreted as commands to the editor where Insert mode indicates keystrokes will produce characters inserted into the text of the file. Use the **Esc** key to exit Insert mode.

Starting and stopping vi

vi filename - Open a file for editing using vi

:wq - Exit and save your work

:q! - Exit without saving

:q - Exit. vi Will warn if you have unsaved changes.

Placing the cursor

These commands are for placing the cursor while not in Insert Mode.

0 (zero) - Move cursor to start of line

\$ - Move cursor to end of line

h or **<Backspace>** - Moves cursor left one character

j,**<Return>** - Moves cursor down one line

k - Moves cursor up one line

I or **<Space>** - move cursor right one character

b - Move cursor to beginning of preceding word

w - Move cursor to beginning of next word

0 + <Return> or **1G** - Move cursor to first line in file

n + <Return> or **nG** - Move cursor to line n

\$ + <Return> or **G** - Move cursor to last line in file

Working with text and entering insert mode

Note: These commands will put vi into insert mode. Remember that insert mode is ended by using the Escape key.

a- Appends text after cursor

A - Appends text at the end of the line

i - Inserts text before cursor

I - Inserts text at the beginning of the line

o - Opens new line below the current line for text insertion

O - Opens new line above the current line for text insertion

DEL - Overwrites last character during text insertion

ESC - Stops text insertion

Appendix B. General Windows commands and information

This Appendix contains specific information that pertains to these labs in particular and WebSphere Application Server in general as it is used on the Windows platforms.

General activities

- Open a command prompt: **Start —> Programs —> Accessories —> Command Prompt.**

Path structures

Table 1:

<variable>	Value
<software_dir>	C:\Software
<software_cds>	C:\Software_CDs
<was_root>	C:\Program Files\IBM\WebSphere\AppServer
<profile_root>	C:\Program Files\IBM\WebSphere\AppServer\profiles
<ihs_root>	C:\Program Files\IBM HTTP Server
<plugin_root>	C:\Program Files\IBM\WebSphere\Plugins
<db2>	C:\Program Files\IBM\SQLLIB

Appendix C. General AIX commands and information

This Appendix contains specific information that pertains to these labs in particular and WebSphere Application Server in general as it is used on the AIX platform.

Path structures

Table 2:

<variable>	Value
<software_dir>	/usr/software
<software_cds>	/usr/software_CDs
<was_root>	/usr/IBM/WebSphere/AppServer
<profile_root>	/usr/IBM/WebSphere/AppServer/profiles
<ihs_root>	/usr/IBMIHS
<plugin_root>	/usr/IBM/WebSphere/Plugins
<db2>	/usr/opt/db2_08_01

Defining WebSphere administrative console users

Use the following commands to create the wasadmin, wasconfig, wasmonitor, and wasoperator users:

- mkgroup <username>
- mkuser pgrp=<username> home=/home/<username> <username>
- passwd <username>

Verify that passwords work properly before going on

Create the wasadmins group

Use the following command to create the wasadmins group and assign its users:

- mkgroup users=wasadmin,userid wasadmins

Appendix D. General Linux commands and information

This Appendix contains specific information that pertains to these labs in particular and WebSphere Application Server in general as it is used on the Linux platform.

Path structures

Table 3:

<variable>	Value
<software_dir>	/usr/software
<software_cds>	/usr/software_CDs
<was_root>	/opt/IBM/WebSphere/AppServer
<profile_root>	/opt/IBM/WebSphere/AppServer/profiles
<ihs_root>	/opt/IBMIHS
<plugin_root>	/opt/IBM/WebSphere/Plugins
<db2>	/opt/IBM/db2/V8.1

Defining WebSphere administrative console users

Use the following commands to create the wasadmin, wasconfig, wasmonitor, and wasoperator users:

- groupadd <username>
- useradd -g <username> -p <pwd> <username>

Verify that passwords work properly before going on

Create the wasadmins group

Use the following commands to create the wasadmins group and assign its users:

- groupadd wasadmins
- usermod -G wasadmins userid
- usermod -G wasadmins wasadmin

IBM[®]